

# Package: ladder.api (via r-universe)

September 9, 2024

**Type** Package

**Title** Google Slides API client and tools

**Version** 0.0.1

**Description** Create, read and modify Slides presentations with full REST API functionality.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**Imports** gargle, httpuv, httr, rlang

**Roxygen** list(markdown = TRUE)

**URL** <https://igrave.github.io/ladder.api/>

**Repository** <https://igrave.r-universe.dev>

**RemoteUrl** <https://github.com/igrave/ladder.api>

**RemoteRef** HEAD

**RemoteSha** 7ea530a3077ce69aa49bbb6f9a14316eae72056b

## Contents

AffineTransform . . . . .	5
Autofit . . . . .	6
AutoText . . . . .	7
BatchUpdatePresentationRequest . . . . .	7
BatchUpdatePresentationResponse . . . . .	8
Bullet . . . . .	9
choose_slides . . . . .	9
ColorScheme . . . . .	10
ColorStop . . . . .	10
CreateImageRequest . . . . .	11
CreateImageResponse . . . . .	12
CreateLineRequest . . . . .	12

CreateLineResponse . . . . .	13
CreateParagraphBulletsRequest . . . . .	14
CreateShapeRequest . . . . .	16
CreateShapeResponse . . . . .	22
CreateSheetsChartRequest . . . . .	23
CreateSheetsChartResponse . . . . .	24
CreateSlideRequest . . . . .	24
CreateSlideResponse . . . . .	25
CreateTableRequest . . . . .	26
CreateTableResponse . . . . .	27
CreateVideoRequest . . . . .	27
CreateVideoResponse . . . . .	28
CropProperties . . . . .	29
DeleteObjectRequest . . . . .	30
DeleteParagraphBulletsRequest . . . . .	31
DeleteTableColumnRequest . . . . .	31
DeleteTableRowRequest . . . . .	32
DeleteTextRequest . . . . .	33
Dimension . . . . .	33
DuplicateObjectRequest . . . . .	34
DuplicateObjectResponse . . . . .	35
Group . . . . .	35
GroupObjectsRequest . . . . .	36
GroupObjectsResponse . . . . .	37
Image . . . . .	37
ImageProperties . . . . .	38
InsertTableColumnsRequest . . . . .	39
InsertTableRowsRequest . . . . .	40
InsertTextRequest . . . . .	41
ladder_auth . . . . .	42
ladder_auth_configure . . . . .	44
ladder_deauth . . . . .	46
ladder_has_token . . . . .	47
ladder_token . . . . .	47
ladder_user . . . . .	48
LayoutPlaceholderIdMapping . . . . .	48
LayoutProperties . . . . .	49
LayoutReference . . . . .	50
Line . . . . .	51
LineConnection . . . . .	52
LineFill . . . . .	53
LineProperties . . . . .	53
Link . . . . .	55
List . . . . .	56
MasterProperties . . . . .	57
MergeTableCellsRequest . . . . .	57
NestingLevel . . . . .	58
NotesProperties . . . . .	58

OpaqueColor . . . . .	59
OptionalColor . . . . .	60
Outline . . . . .	61
OutlineFill . . . . .	62
Page . . . . .	63
PageBackgroundFill . . . . .	64
PageElement . . . . .	65
PageElementProperties . . . . .	66
PageProperties . . . . .	67
ParagraphMarker . . . . .	68
ParagraphStyle . . . . .	68
Placeholder . . . . .	70
Presentation . . . . .	71
presentations.batchUpdate . . . . .	72
presentations.create . . . . .	73
presentations.get . . . . .	73
presentations.pages.get . . . . .	74
presentations.pages.getThumbnail . . . . .	74
Range . . . . .	75
Recolor . . . . .	76
RefreshSheetsChartRequest . . . . .	77
ReplaceAllShapesWithImageRequest . . . . .	78
ReplaceAllShapesWithImageResponse . . . . .	79
ReplaceAllShapesWithSheetsChartRequest . . . . .	80
ReplaceAllShapesWithSheetsChartResponse . . . . .	81
ReplaceAllTextRequest . . . . .	82
ReplaceAllTextResponse . . . . .	82
ReplaceImageRequest . . . . .	83
Request . . . . .	84
RerouteLineRequest . . . . .	86
Response . . . . .	87
RgbColor . . . . .	88
Shadow . . . . .	88
Shape . . . . .	90
ShapeBackgroundFill . . . . .	97
ShapeProperties . . . . .	98
SheetsChart . . . . .	99
SheetsChartProperties . . . . .	100
Size . . . . .	100
SlideProperties . . . . .	101
SolidFill . . . . .	102
SpeakerSpotlight . . . . .	102
SpeakerSpotlightProperties . . . . .	103
StretchedPictureFill . . . . .	103
SubstringMatchCriteria . . . . .	104
Table . . . . .	105
TableBorderCell . . . . .	106
TableBorderFill . . . . .	106

TableBorderProperties . . . . .	107
TableBorderRow . . . . .	108
TableCell . . . . .	108
TableCellBackgroundFill . . . . .	109
TableCellLocation . . . . .	110
TableCellProperties . . . . .	111
TableColumnProperties . . . . .	112
TableRange . . . . .	112
TableRow . . . . .	113
TableRowProperties . . . . .	113
TextContent . . . . .	114
TextElement . . . . .	115
TextRun . . . . .	116
TextStyle . . . . .	116
ThemeColorPair . . . . .	118
Thumbnail . . . . .	119
UngroupObjectsRequest . . . . .	120
UnmergeTableCellsRequest . . . . .	121
UpdateImagePropertiesRequest . . . . .	121
UpdateLineCategoryRequest . . . . .	122
UpdateLinePropertiesRequest . . . . .	123
UpdatePageElementAltTextRequest . . . . .	124
UpdatePageElementsZOrderRequest . . . . .	124
UpdatePageElementTransformRequest . . . . .	125
UpdatePagePropertiesRequest . . . . .	126
UpdateParagraphStyleRequest . . . . .	127
UpdateShapePropertiesRequest . . . . .	128
UpdateSlidePropertiesRequest . . . . .	128
UpdateSlidesPositionRequest . . . . .	129
UpdateTableBorderPropertiesRequest . . . . .	130
UpdateTableCellPropertiesRequest . . . . .	131
UpdateTableColumnPropertiesRequest . . . . .	132
UpdateTableRowPropertiesRequest . . . . .	133
UpdateTextStyleRequest . . . . .	134
UpdateVideoPropertiesRequest . . . . .	135
Video . . . . .	135
VideoProperties . . . . .	136
WeightedFontFamily . . . . .	137
WordArt . . . . .	138
WriteControl . . . . .	138

---

AffineTransform	<i>AffineTransform Object</i>
-----------------	-------------------------------

---

**Description**

AffineTransform Object

**Usage**

```
AffineTransform(
    scaleX = NULL,
    scaleY = NULL,
    shearX = NULL,
    shearY = NULL,
    translateX = NULL,
    translateY = NULL,
    unit = NULL
)
```

**Arguments**

scaleX	(number) The X coordinate scaling element.
scaleY	(number) The Y coordinate scaling element.
shearX	(number) The X coordinate shearing element.
shearY	(number) The Y coordinate shearing element.
translateX	(number) The X coordinate translation element.
translateY	(number) The Y coordinate translation element.
unit	(string) The units for translate elements. <ul style="list-style-type: none"> <li>• UNIT_UNSPECIFIED: The units are unknown.</li> <li>• EMU: An English Metric Unit (EMU) is defined as 1/360,000 of a centimeter and thus there are 914,400 EMUs per inch, and 12,700 EMUs per point.</li> <li>• PT: A point, 1/72 of an inch.</li> </ul>

**Details**

AffineTransform uses a 3x3 matrix with an implied last row of [ 0 0 1 ] to transform source coordinates (x,y) into destination coordinates (x', y') according to:  $x' = \text{shear}_y \text{scale}_y \text{translate}_y + x$  [ 1 ] After transformation,  $x' = \text{scale}_x * x + \text{shear}_x * y + \text{translate}_x$ ;  $y' = \text{scale}_y * y + \text{shear}_y * x + \text{translate}_y$ ; This message is therefore composed of these six matrix elements.

**Value**

AffineTransform object

Autofit

*Autofit Object***Description**

Autofit Object

**Usage**

Autofit(autofitType = NULL, fontScale = NULL, lineSpacingReduction = NULL)

**Arguments**

- |                      |   |
|----------------------|---|
| autofitType          | (string) The autofit type of the shape. If the autofit type is AUTOFIT_TYPE_UNSPECIFIED, the autofit type is inherited from a parent placeholder if it exists. The field is automatically set to NONE if a request is made that might affect text fitting within its bounding text box. In this case the font_scale is applied to the font_size and the line_spacing_reduction is applied to the line_spacing. Both properties are also reset to default values. <ul style="list-style-type: none"> <li>• AUTOFIT_TYPE_UNSPECIFIED: The autofit type is unspecified.</li> <li>• NONE: Do not autofit.</li> <li>• TEXT_AUTOFIT: Shrink text on overflow to fit the shape.</li> <li>• SHAPE_AUTOFIT: Resize the shape to fit the text.</li> </ul> |
| fontScale            | (number) The font scale applied to the shape. For shapes with autofit_type NONE or SHAPE_AUTOFIT, this value is the default value of 1. For TEXT_AUTOFIT, this value multiplied by the font_size gives the font size that is rendered in the editor. This property is read-only.  |
| lineSpacingReduction | (number) The line spacing reduction applied to the shape. For shapes with autofit_type NONE or SHAPE_AUTOFIT, this value is the default value of 0. For TEXT_AUTOFIT, this value subtracted from the line_spacing gives the line spacing that is rendered in the editor. This property is read-only.  |

**Details**

The autofit properties of a Shape.

**Value**

Autofit object

---

 AutoText

*AutoText Object*


---

**Description**

AutoText Object

**Usage**

AutoText(type = NULL, content = NULL, style = NULL)

**Arguments**

type	(string) The type of this auto text. <ul style="list-style-type: none"> <li>• TYPE_UNSPECIFIED: An unspecified autotext type.</li> <li>• SLIDE_NUMBER: Type for autotext that represents the current slide number.</li> </ul>
content	(string) The rendered content of this auto text, if available.
style	( <a href="#">TextStyle</a> ) The styling applied to this auto text.

**Details**

A TextElement kind that represents auto text.

**Value**

AutoText object

---

 BatchUpdatePresentationRequest

*BatchUpdatePresentationRequest Object*


---

**Description**

BatchUpdatePresentationRequest Object

**Usage**

BatchUpdatePresentationRequest(requests = NULL, writeControl = NULL)

**Arguments**

requests	(list of <a href="#">Request</a> objects) A list of updates to apply to the presentation.
writeControl	( <a href="#">WriteControl</a> ) Provides control over how write requests are executed.

**Details**

Request message for PresentationsService.BatchUpdatePresentation.

**Value**

BatchUpdatePresentationRequest object

---

BatchUpdatePresentationResponse

*BatchUpdatePresentationResponse Object*

---

**Description**

BatchUpdatePresentationResponse Object

**Usage**

```
BatchUpdatePresentationResponse(  
    presentationId = NULL,  
    replies = NULL,  
    writeControl = NULL  
)
```

**Arguments**

presentationId (string) The presentation the updates were applied to.

replies (list of [Response](#) objects) The reply of the updates. This maps 1:1 with the updates, although replies to some requests may be empty.

writeControl ([WriteControl](#)) The updated write control after applying the request.

**Details**

Response message from a batch update.

**Value**

BatchUpdatePresentationResponse object



---

Bullet	<i>Bullet Object</i>
--------	----------------------

---

**Description**

Bullet Object

**Usage**

```
Bullet(listId = NULL, nestingLevel = NULL, glyph = NULL, bulletStyle = NULL)
```

**Arguments**

listId	(string) The ID of the list this paragraph belongs to.
nestingLevel	(integer) The nesting level of this paragraph in the list.
glyph	(string) The rendered bullet glyph for this paragraph.
bulletStyle	( <a href="#">TextStyle</a> ) The paragraph specific text style applied to this bullet.

**Details**

Describes the bullet of a paragraph.

**Value**

Bullet object

---

choose_slides	<i>Choose Slides presentation</i>
---------------	-----------------------------------

---

**Description**

Opens a webpage for a user to authenticate with Google and select a presentation. This presentation is then authorised for use with ladder.

**Usage**

```
choose_slides()
```

**Value**

A presentation id

**Examples**

```
id <- choose_slides()
```

---

ColorScheme	<i>ColorScheme Object</i>
-------------	---------------------------

---

**Description**

ColorScheme Object

**Usage**

```
ColorScheme(colors = NULL)
```

**Arguments**

colors	(list of <a href="#">ThemeColorPair</a> objects) The ThemeColorType and corresponding concrete color pairs.
--------	---

**Details**

The palette of predefined colors for a page.

**Value**

ColorScheme object

---

ColorStop	<i>ColorStop Object</i>
-----------	-------------------------

---

**Description**

ColorStop Object

**Usage**

```
ColorStop(color = NULL, alpha = NULL, position = NULL)
```

**Arguments**

color	( <a href="#">OpaqueColor</a> ) The color of the gradient stop.
alpha	(number) The alpha value of this color in the gradient band. Defaults to 1.0, fully opaque.
position	(number) The relative position of the color stop in the gradient band measured in percentage. The value should be in the interval [0.0, 1.0].

**Details**

A color and position in a gradient band.

**Value**

ColorStop object

---

CreateImageRequest	<i>CreateImageRequest Object</i>
--------------------	----------------------------------

---

**Description**

CreateImageRequest Object

**Usage**

```
CreateImageRequest(objectId = NULL, elementProperties = NULL, url = NULL)
```

**Arguments**

objectId	(string) A user-supplied object ID. If you specify an ID, it must be unique among all pages and page elements in the presentation. The ID must start with an alphanumeric character or an underscore (matches regex <code>[a-zA-Z0-9_]</code> ); remaining characters may include those as well as a hyphen or colon (matches regex <code>[a-zA-Z0-9_-: ]</code> ). The length of the ID must not be less than 5 or greater than 50. If you don't specify an ID, a unique one is generated.
elementProperties	( <a href="#">PageElementProperties</a> ) The element properties for the image. When the aspect ratio of the provided size does not match the image aspect ratio, the image is scaled and centered with respect to the size in order to maintain the aspect ratio. The provided transform is applied after this operation. The <code>PageElementProperties.size</code> property is optional. If you don't specify the size, the default size of the image is used. The <code>PageElementProperties.transform</code> property is optional. If you don't specify a transform, the image will be placed at the top-left corner of the page.
url	(string) The image URL. The image is fetched once at insertion time and a copy is stored for display inside the presentation. Images must be less than 50 MB in size, can't exceed 25 megapixels, and must be in one of PNG, JPEG, or GIF formats. The provided URL must be publicly accessible and up to 2 KB in length. The URL is saved with the image, and exposed through the <code>Image.source_url</code> field.

**Details**

Creates an image.

**Value**

CreateImageRequest object

---

CreateImageResponse    *CreateImageResponse Object*

---

**Description**

CreateImageResponse Object

**Usage**

```
CreateImageResponse(objectId = NULL)
```

**Arguments**

objectId            (string) The object ID of the created image.

**Details**

The result of creating an image.

**Value**

CreateImageResponse object

---

CreateLineRequest    *CreateLineRequest Object*

---

**Description**

CreateLineRequest Object

**Usage**

```
CreateLineRequest(  
  objectId = NULL,  
  elementProperties = NULL,  
  lineCategory = NULL,  
  category = NULL  
)
```

**Arguments**

objectId	(string) A user-supplied object ID. If you specify an ID, it must be unique among all pages and page elements in the presentation. The ID must start with an alphanumeric character or an underscore (matches regex <code>[a-zA-Z0-9_]</code> ); remaining characters may include those as well as a hyphen or colon (matches regex <code>[a-zA-Z0-9_-: ]</code> ). The length of the ID must not be less than 5 or greater than 50. If you don't specify an ID, a unique one is generated.
elementProperties	( <a href="#">PageElementProperties</a> ) The element properties for the line.
lineCategory	(string) The category of the line to be created. <i>Deprecated</i> : use <code>category</code> instead. The exact line type created is determined based on the category and how it's routed to connect to other page elements. If you specify both a category and a <code>line_category</code> , the category takes precedence. <ul style="list-style-type: none"> <li>• STRAIGHT: Straight connectors, including straight connector 1. This is the default category when one is not specified.</li> <li>• BENT: Bent connectors, including bent connector 2 to 5.</li> <li>• CURVED: Curved connectors, including curved connector 2 to 5.</li> </ul>
category	(string) The category of the line to be created. The exact line type created is determined based on the category and how it's routed to connect to other page elements. If you specify both a category and a <code>line_category</code> , the category takes precedence. If you do not specify a value for <code>category</code> , but specify a value for <code>line_category</code> , then the specified <code>line_category</code> value is used. If you do not specify either, then STRAIGHT is used. <ul style="list-style-type: none"> <li>• LINE_CATEGORY_UNSPECIFIED: Unspecified line category.</li> <li>• STRAIGHT: Straight connectors, including straight connector 1.</li> <li>• BENT: Bent connectors, including bent connector 2 to 5.</li> <li>• CURVED: Curved connectors, including curved connector 2 to 5.</li> </ul>

**Details**

Creates a line.

**Value**

CreateLineRequest object

---

CreateLineResponse      *CreateLineResponse Object*

---

**Description**

CreateLineResponse Object

**Usage**

```
CreateLineResponse(objectId = NULL)
```

**Arguments**

objectId (string) The object ID of the created line.

**Details**

The result of creating a line.

**Value**

CreateLineResponse object

---

CreateParagraphBulletsRequest

*CreateParagraphBulletsRequest Object*

---

**Description**

CreateParagraphBulletsRequest Object

**Usage**

```
CreateParagraphBulletsRequest(
  objectId = NULL,
  cellLocation = NULL,
  textRange = NULL,
  bulletPreset = NULL
)
```

**Arguments**

objectId	(string) The object ID of the shape or table containing the text to add bullets to.
cellLocation	( <a href="#">TableCellLocation</a> ) The optional table cell location if the text to be modified is in a table cell. If present, the object_id must refer to a table.
textRange	( <a href="#">Range</a> ) The range of text to apply the bullet presets to, based on TextElement indexes.
bulletPreset	(string) The kinds of bullet glyphs to be used. Defaults to the BULLET_DISC_CIRCLE_SQUARE preset. <ul style="list-style-type: none"> <li>BULLET_DISC_CIRCLE_SQUARE: A bulleted list with a DISC, CIRCLE and SQUARE bullet glyph for the first 3 list nesting levels.</li> <li>BULLET_DIAMONDX_ARROW3D_SQUARE: A bulleted list with a DIAMONDX, ARROW3D and SQUARE bullet glyph for the first 3 list nesting levels.</li> </ul>

- BULLET\_CHECKBOX: A bulleted list with CHECKBOX bullet glyphs for all list nesting levels.
- BULLET\_ARROW\_DIAMOND\_DISC: A bulleted list with a ARROW, DIAMOND and DISC bullet glyph for the first 3 list nesting levels.
- BULLET\_STAR\_CIRCLE\_SQUARE: A bulleted list with a STAR, CIRCLE and SQUARE bullet glyph for the first 3 list nesting levels.
- BULLET\_ARROW3D\_CIRCLE\_SQUARE: A bulleted list with a ARROW3D, CIRCLE and SQUARE bullet glyph for the first 3 list nesting levels.
- BULLET\_LEFTTRIANGLE\_DIAMOND\_DISC: A bulleted list with a LEFTTRIANGLE, DIAMOND and DISC bullet glyph for the first 3 list nesting levels.
- BULLET\_DIAMOND\_X\_HOLLOWDIAMOND\_SQUARE: A bulleted list with a DIAMOND\_X, HOLLOWDIAMOND and SQUARE bullet glyph for the first 3 list nesting levels.
- BULLET\_DIAMOND\_CIRCLE\_SQUARE: A bulleted list with a DIAMOND, CIRCLE and SQUARE bullet glyph for the first 3 list nesting levels.
- NUMBERED\_DIGIT\_ALPHA\_ROMAN: A numbered list with DIGIT, ALPHA and ROMAN numeric glyphs for the first 3 list nesting levels, followed by periods.
- NUMBERED\_DIGIT\_ALPHA\_ROMAN\_PARENS: A numbered list with DIGIT, ALPHA and ROMAN numeric glyphs for the first 3 list nesting levels, followed by parenthesis.
- NUMBERED\_DIGIT\_NESTED: A numbered list with DIGIT numeric glyphs separated by periods, where each nesting level uses the previous nesting level's glyph as a prefix. For example: '1.', '1.1.', '2.', '2.2.'.
- NUMBERED\_UPPERALPHA\_ALPHA\_ROMAN: A numbered list with UPPERALPHA, ALPHA and ROMAN numeric glyphs for the first 3 list nesting levels, followed by periods.
- NUMBERED\_UPPERROMAN\_UPPERALPHA\_DIGIT: A numbered list with UPPERROMAN, UPPERALPHA and DIGIT numeric glyphs for the first 3 list nesting levels, followed by periods.
- NUMBERED\_ZERODIGIT\_ALPHA\_ROMAN: A numbered list with ZERODIGIT, ALPHA and ROMAN numeric glyphs for the first 3 list nesting levels, followed by periods.

### Details

Creates bullets for all of the paragraphs that overlap with the given text index range. The nesting level of each paragraph will be determined by counting leading tabs in front of each paragraph. To avoid excess space between the bullet and the corresponding paragraph, these leading tabs are removed by this request. This may change the indices of parts of the text. If the paragraph immediately before paragraphs being updated is in a list with a matching preset, the paragraphs being updated are added to that preceding list.

### Value

CreateParagraphBulletsRequest object

---

CreateShapeRequest      *CreateShapeRequest Object*

---

## Description

CreateShapeRequest Object

## Usage

CreateShapeRequest(objectId = NULL, elementProperties = NULL, shapeType = NULL)

## Arguments

- |                   |  |
|-------------------|--|
| objectId          | (string) A user-supplied object ID. If you specify an ID, it must be unique among all pages and page elements in the presentation. The ID must start with an alphanumeric character or an underscore (matches regex <code>[a-zA-Z0-9_]</code> ); remaining characters may include those as well as a hyphen or colon (matches regex <code>[a-zA-Z0-9_-:]</code> ). The length of the ID must not be less than 5 or greater than 50. If empty, a unique identifier will be generated.   |
| elementProperties | ( <a href="#">PageElementProperties</a> ) The element properties for the shape.  |
| shapeType         | (string) The shape type. <ul style="list-style-type: none"> <li>• TYPE_UNSPECIFIED: The shape type that is not predefined.</li> <li>• TEXT_BOX: Text box shape.</li> <li>• RECTANGLE: Rectangle shape. Corresponds to ECMA-376 ST_ShapeType 'rect'.</li> <li>• ROUND_RECTANGLE: Round corner rectangle shape. Corresponds to ECMA-376 ST_ShapeType 'roundRect'.</li> <li>• ELLIPSE: Ellipse shape. Corresponds to ECMA-376 ST_ShapeType 'ellipse'.</li> <li>• ARC: Curved arc shape. Corresponds to ECMA-376 ST_ShapeType 'arc'.</li> <li>• BENT_ARROW: Bent arrow shape. Corresponds to ECMA-376 ST_ShapeType 'bentArrow'.</li> <li>• BENT_UP_ARROW: Bent up arrow shape. Corresponds to ECMA-376 ST_ShapeType 'bentUpArrow'.</li> <li>• BEVEL: Bevel shape. Corresponds to ECMA-376 ST_ShapeType 'bevel'.</li> <li>• BLOCK_ARC: Block arc shape. Corresponds to ECMA-376 ST_ShapeType 'blockArc'.</li> <li>• BRACE_PAIR: Brace pair shape. Corresponds to ECMA-376 ST_ShapeType 'bracePair'.</li> <li>• BRACKET_PAIR: Bracket pair shape. Corresponds to ECMA-376 ST_ShapeType 'bracketPair'.</li> <li>• CAN: Can shape. Corresponds to ECMA-376 ST_ShapeType 'can'.</li> <li>• CHEVRON: Chevron shape. Corresponds to ECMA-376 ST_ShapeType 'chevron'.</li> </ul> |



- CHORD: Chord shape. Corresponds to ECMA-376 ST\_ShapeType 'chord'
- CLOUD: Cloud shape. Corresponds to ECMA-376 ST\_ShapeType 'cloud'
- CORNER: Corner shape. Corresponds to ECMA-376 ST\_ShapeType 'corner'
- CUBE: Cube shape. Corresponds to ECMA-376 ST\_ShapeType 'cube'
- CURVED\_DOWN\_ARROW: Curved down arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'curvedDownArrow'
- CURVED\_LEFT\_ARROW: Curved left arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'curvedLeftArrow'
- CURVED\_RIGHT\_ARROW: Curved right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'curvedRightArrow'
- CURVED\_UP\_ARROW: Curved up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'curvedUpArrow'
- DECAGON: Decagon shape. Corresponds to ECMA-376 ST\_ShapeType 'decagon'
- DIAGONAL\_STRIPE: Diagonal stripe shape. Corresponds to ECMA-376 ST\_ShapeType 'diagStripe'
- DIAMOND: Diamond shape. Corresponds to ECMA-376 ST\_ShapeType 'diamond'
- DODECAGON: Dodecagon shape. Corresponds to ECMA-376 ST\_ShapeType 'dodecagon'
- DONUT: Donut shape. Corresponds to ECMA-376 ST\_ShapeType 'donut'
- DOUBLE\_WAVE: Double wave shape. Corresponds to ECMA-376 ST\_ShapeType 'doubleWave'
- DOWN\_ARROW: Down arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'downArrow'
- DOWN\_ARROW\_CALLOUT: Callout down arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'downArrowCallout'
- FOLDED\_CORNER: Folded corner shape. Corresponds to ECMA-376 ST\_ShapeType 'foldedCorner'
- FRAME: Frame shape. Corresponds to ECMA-376 ST\_ShapeType 'frame'
- HALF\_FRAME: Half frame shape. Corresponds to ECMA-376 ST\_ShapeType 'halfFrame'
- HEART: Heart shape. Corresponds to ECMA-376 ST\_ShapeType 'heart'
- HEPTAGON: Heptagon shape. Corresponds to ECMA-376 ST\_ShapeType 'heptagon'
- HEXAGON: Hexagon shape. Corresponds to ECMA-376 ST\_ShapeType 'hexagon'
- HOME\_PLATE: Home plate shape. Corresponds to ECMA-376 ST\_ShapeType 'homePlate'
- HORIZONTAL\_SCROLL: Horizontal scroll shape. Corresponds to ECMA-376 ST\_ShapeType 'horizontalScroll'
- IRREGULAR\_SEAL\_1: Irregular seal 1 shape. Corresponds to ECMA-376 ST\_ShapeType 'irregularSeal1'
- IRREGULAR\_SEAL\_2: Irregular seal 2 shape. Corresponds to ECMA-376 ST\_ShapeType 'irregularSeal2'
- LEFT\_ARROW: Left arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftArrow'

- LEFT\_ARROW\_CALLOUT: Callout left arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftArrowCallout'
- LEFT\_BRACE: Left brace shape. Corresponds to ECMA-376 ST\_ShapeType 'leftBrace'
- LEFT\_BRACKET: Left bracket shape. Corresponds to ECMA-376 ST\_ShapeType 'leftBracket'
- LEFT\_RIGHT\_ARROW: Left right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftRightArrow'
- LEFT\_RIGHT\_ARROW\_CALLOUT: Callout left right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftRightArrowCallout'
- LEFT\_RIGHT\_UP\_ARROW: Left right up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftRightUpArrow'
- LEFT\_UP\_ARROW: Left up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftUpArrow'
- LIGHTNING\_BOLT: Lightning bolt shape. Corresponds to ECMA-376 ST\_ShapeType 'lightningBolt'
- MATH\_DIVIDE: Divide math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathDivide'
- MATH\_EQUAL: Equal math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathEqual'
- MATH\_MINUS: Minus math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathMinus'
- MATH\_MULTIPLY: Multiply math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathMultiply'
- MATH\_NOT\_EQUAL: Not equal math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathNotEqual'
- MATH\_PLUS: Plus math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathPlus'
- MOON: Moon shape. Corresponds to ECMA-376 ST\_ShapeType 'moon'
- NO\_SMOKING: No smoking shape. Corresponds to ECMA-376 ST\_ShapeType 'noSmoking'
- NOTCHED\_RIGHT\_ARROW: Notched right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'notchedRightArrow'
- OCTAGON: Octagon shape. Corresponds to ECMA-376 ST\_ShapeType 'octagon'
- PARALLELOGRAM: Parallelogram shape. Corresponds to ECMA-376 ST\_ShapeType 'parallelogram'
- PENTAGON: Pentagon shape. Corresponds to ECMA-376 ST\_ShapeType 'pentagon'
- PIE: Pie shape. Corresponds to ECMA-376 ST\_ShapeType 'pie'
- PLAQUE: Plaque shape. Corresponds to ECMA-376 ST\_ShapeType 'plaque'
- PLUS: Plus shape. Corresponds to ECMA-376 ST\_ShapeType 'plus'
- QUAD\_ARROW: Quad-arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'quadArrow'

- QUAD\_ARROW\_CALLOUT: Callout quad-arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'quadArrowCallout'
- RIBBON: Ribbon shape. Corresponds to ECMA-376 ST\_ShapeType 'ribbon'
- RIBBON\_2: Ribbon 2 shape. Corresponds to ECMA-376 ST\_ShapeType 'ribbon2'
- RIGHT\_ARROW: Right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'rightArrow'
- RIGHT\_ARROW\_CALLOUT: Callout right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'rightArrowCallout'
- RIGHT\_BRACE: Right brace shape. Corresponds to ECMA-376 ST\_ShapeType 'rightBrace'
- RIGHT\_BRACKET: Right bracket shape. Corresponds to ECMA-376 ST\_ShapeType 'rightBracket'
- ROUND\_1\_RECTANGLE: One round corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'round1Rect'
- ROUND\_2\_DIAGONAL\_RECTANGLE: Two diagonal round corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'round2DiagRect'
- ROUND\_2\_SAME\_RECTANGLE: Two same-side round corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'round2SameRect'
- RIGHT\_TRIANGLE: Right triangle shape. Corresponds to ECMA-376 ST\_ShapeType 'rtTriangle'
- SMILEY\_FACE: Smiley face shape. Corresponds to ECMA-376 ST\_ShapeType 'smileyFace'
- SNIP\_1\_RECTANGLE: One snip corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'snip1Rect'
- SNIP\_2\_DIAGONAL\_RECTANGLE: Two diagonal snip corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'snip2DiagRect'
- SNIP\_2\_SAME\_RECTANGLE: Two same-side snip corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'snip2SameRect'
- SNIP\_ROUND\_RECTANGLE: One snip one round corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'snipRoundRect'
- STAR\_10: Ten pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star10'
- STAR\_12: Twelve pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star12'
- STAR\_16: Sixteen pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star16'
- STAR\_24: Twenty four pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star24'
- STAR\_32: Thirty two pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star32'
- STAR\_4: Four pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star4'
- STAR\_5: Five pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star5'

- STAR\_6: Six pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star6'
- STAR\_7: Seven pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star7'
- STAR\_8: Eight pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star8'
- STRIPED\_RIGHT\_ARROW: Striped right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'stripedRightArrow'
- SUN: Sun shape. Corresponds to ECMA-376 ST\_ShapeType 'sun'
- TRAPEZOID: Trapezoid shape. Corresponds to ECMA-376 ST\_ShapeType 'trapezoid'
- TRIANGLE: Triangle shape. Corresponds to ECMA-376 ST\_ShapeType 'triangle'
- UP\_ARROW: Up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'upArrow'
- UP\_ARROW\_CALLOUT: Callout up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'upArrowCallout'
- UP\_DOWN\_ARROW: Up down arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'upDownArrow'
- UTURN\_ARROW: U-turn arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'uturnArrow'
- VERTICAL\_SCROLL: Vertical scroll shape. Corresponds to ECMA-376 ST\_ShapeType 'verticalScroll'
- WAVE: Wave shape. Corresponds to ECMA-376 ST\_ShapeType 'wave'
- WEDGE\_ELLIPSE\_CALLOUT: Callout wedge ellipse shape. Corresponds to ECMA-376 ST\_ShapeType 'wedgeEllipseCallout'
- WEDGE\_RECTANGLE\_CALLOUT: Callout wedge rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'wedgeRectCallout'
- WEDGE\_ROUND\_RECTANGLE\_CALLOUT: Callout wedge round rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'wedgeRoundRectCallout'
- FLOW\_CHART\_ALTERNATE\_PROCESS: Alternate process flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartAlternateProcess'
- FLOW\_CHART\_COLLATE: Collate flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartCollate'
- FLOW\_CHART\_CONNECTOR: Connector flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartConnector'
- FLOW\_CHART\_DECISION: Decision flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartDecision'
- FLOW\_CHART\_DELAY: Delay flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartDelay'
- FLOW\_CHART\_DISPLAY: Display flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartDisplay'
- FLOW\_CHART\_DOCUMENT: Document flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartDocument'
- FLOW\_CHART\_EXTRACT: Extract flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartExtract'

- FLOW\_CHART\_INPUT\_OUTPUT: Input output flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartInputOutput'
- FLOW\_CHART\_INTERNAL\_STORAGE: Internal storage flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartInternalStorage'
- FLOW\_CHART\_MAGNETIC\_DISK: Magnetic disk flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartMagneticDisk'
- FLOW\_CHART\_MAGNETIC\_DRUM: Magnetic drum flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartMagneticDrum'
- FLOW\_CHART\_MAGNETIC\_TAPE: Magnetic tape flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartMagneticTape'
- FLOW\_CHART\_MANUAL\_INPUT: Manual input flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartManualInput'
- FLOW\_CHART\_MANUAL\_OPERATION: Manual operation flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartManualOperation'
- FLOW\_CHART\_MERGE: Merge flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartMerge'
- FLOW\_CHART\_MULTIDOCUMENT: Multi-document flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartMultidocument'
- FLOW\_CHART\_OFFLINE\_STORAGE: Offline storage flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartOfflineStorage'
- FLOW\_CHART\_OFFPAGE\_CONNECTOR: Off-page connector flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartOffpageConnector'
- FLOW\_CHART\_ONLINE\_STORAGE: Online storage flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartOnlineStorage'
- FLOW\_CHART\_OR: Or flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartOr'
- FLOW\_CHART\_PREDEFINED\_PROCESS: Predefined process flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartPredefinedProcess'
- FLOW\_CHART\_PREPARATION: Preparation flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartPreparation'
- FLOW\_CHART\_PROCESS: Process flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartProcess'
- FLOW\_CHART\_PUNCHED\_CARD: Punched card flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartPunchedCard'
- FLOW\_CHART\_PUNCHED\_TAPE: Punched tape flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartPunchedTape'
- FLOW\_CHART\_SORT: Sort flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartSort'
- FLOW\_CHART\_SUMMING\_JUNCTION: Summing junction flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartSummingJunction'
- FLOW\_CHART\_TERMINATOR: Terminator flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartTerminator'
- ARROW\_EAST: East arrow shape.
- ARROW\_NORTH\_EAST: Northeast arrow shape.
- ARROW\_NORTH: North arrow shape.

- SPEECH: Speech shape.
- STARBURST: Star burst shape.
- TEARDROP: Teardrop shape. Corresponds to ECMA-376 ST\_ShapeType 'teardrop'
- ELLIPSE\_RIBBON: Ellipse ribbon shape. Corresponds to ECMA-376 ST\_ShapeType 'ellipseRibbon'
- ELLIPSE\_RIBBON\_2: Ellipse ribbon 2 shape. Corresponds to ECMA-376 ST\_ShapeType 'ellipseRibbon2'
- CLOUD\_CALLOUT: Callout cloud shape. Corresponds to ECMA-376 ST\_ShapeType 'cloudCallout'
- CUSTOM: Custom shape.

**Details**

Creates a new shape.

**Value**

CreateShapeRequest object

---

CreateShapeResponse    *CreateShapeResponse Object*

---

**Description**

CreateShapeResponse Object

**Usage**

CreateShapeResponse(objectId = NULL)

**Arguments**

objectId            (string) The object ID of the created shape.

**Details**

The result of creating a shape.

**Value**

CreateShapeResponse object

---

CreateSheetsChartRequest

*CreateSheetsChartRequest Object*

---

## Description

CreateSheetsChartRequest Object

## Usage

```
CreateSheetsChartRequest(  
  objectId = NULL,  
  elementProperties = NULL,  
  spreadsheetId = NULL,  
  chartId = NULL,  
  linkingMode = NULL  
)
```

## Arguments

objectId	(string) A user-supplied object ID. If specified, the ID must be unique among all pages and page elements in the presentation. The ID should start with a word character [a-zA-Z0-9_] and then followed by any number of the following characters [a-zA-Z0-9_-:]. The length of the ID should not be less than 5 or greater than 50. If empty, a unique identifier will be generated.
elementProperties	( <a href="#">PageElementProperties</a> ) The element properties for the chart. When the aspect ratio of the provided size does not match the chart aspect ratio, the chart is scaled and centered with respect to the size in order to maintain aspect ratio. The provided transform is applied after this operation.
spreadsheetId	(string) The ID of the Google Sheets spreadsheet that contains the chart. You might need to add a resource key to the HTTP header for a subset of old files. For more information, see <a href="#">Access link-shared files using resource keys</a> .
chartId	(integer) The ID of the specific chart in the Google Sheets spreadsheet.
linkingMode	(string) The mode with which the chart is linked to the source spreadsheet. When not specified, the chart will be an image that is not linked. <ul style="list-style-type: none"><li>• NOT_LINKED_IMAGE: The chart is not associated with the source spreadsheet and cannot be updated. A chart that is not linked will be inserted as an image.</li><li>• LINKED: Linking the chart allows it to be updated, and other collaborators will see a link to the spreadsheet.</li></ul>

## Details

Creates an embedded Google Sheets chart. NOTE: Chart creation requires at least one of the spreadsheets.readonly, spreadsheets, drive.readonly, drive.file, or drive OAuth scopes.

**Value**

CreateSheetsChartRequest object

---

CreateSheetsChartResponse

*CreateSheetsChartResponse Object*

---

**Description**

CreateSheetsChartResponse Object

**Usage**

```
CreateSheetsChartResponse(objectId = NULL)
```

**Arguments**

objectId (string) The object ID of the created chart.

**Details**

The result of creating an embedded Google Sheets chart.

**Value**

CreateSheetsChartResponse object

---

CreateSlideRequest

*CreateSlideRequest Object*

---

**Description**

CreateSlideRequest Object

**Usage**

```
CreateSlideRequest(  
  objectId = NULL,  
  insertionIndex = NULL,  
  slideLayoutReference = NULL,  
  placeholderIdMappings = NULL  
)
```



**Arguments**

- objectId** (string) A user-supplied object ID. If you specify an ID, it must be unique among all pages and page elements in the presentation. The ID must start with an alphanumeric character or an underscore (matches regex `[a-zA-Z0-9_]`); remaining characters may include those as well as a hyphen or colon (matches regex `[a-zA-Z0-9_-: ]`). The ID length must be between 5 and 50 characters, inclusive. If you don't specify an ID, a unique one is generated.
- insertionIndex** (integer) The optional zero-based index indicating where to insert the slides. If you don't specify an index, the slide is created at the end.
- slideLayoutReference** ([LayoutReference](#)) Layout reference of the slide to be inserted, based on the *current master*, which is one of the following: - The master of the previous slide index. - The master of the first slide, if the `insertion_index` is zero. - The first master in the presentation, if there are no slides. If the `LayoutReference` is not found in the current master, a 400 bad request error is returned. If you don't specify a layout reference, the slide uses the predefined BLANK layout.
- placeholderIdMappings** (list of [LayoutPlaceholderIdMapping](#) objects) An optional list of object ID mappings from the placeholder(s) on the layout to the placeholders that are created on the slide from the specified layout. Can only be used when `slide_layout_reference` is specified.

**Details**

Creates a slide.

**Value**

CreateSlideRequest object

---

CreateSlideResponse    *CreateSlideResponse Object*

---

**Description**

CreateSlideResponse Object

**Usage**

CreateSlideResponse(objectId = NULL)

**Arguments**

- objectId** (string) The object ID of the created slide.

**Details**

The result of creating a slide.

**Value**

CreateSlideResponse object

---

CreateTableRequest      *CreateTableRequest Object*

---

**Description**

CreateTableRequest Object

**Usage**

```
CreateTableRequest(
  objectId = NULL,
  elementProperties = NULL,
  rows = NULL,
  columns = NULL
)
```

**Arguments**

objectId	(string) A user-supplied object ID. If you specify an ID, it must be unique among all pages and page elements in the presentation. The ID must start with an alphanumeric character or an underscore (matches regex <code>[a-zA-Z0-9_]</code> ); remaining characters may include those as well as a hyphen or colon (matches regex <code>[a-zA-Z0-9_-: ]</code> ). The length of the ID must not be less than 5 or greater than 50. If you don't specify an ID, a unique one is generated.
elementProperties	( <a href="#">PageElementProperties</a> ) The element properties for the table. The table will be created at the provided size, subject to a minimum size. If no size is provided, the table will be automatically sized. Table transforms must have a scale of 1 and no shear components. If no transform is provided, the table will be centered on the page.
rows	(integer) Number of rows in the table.
columns	(integer) Number of columns in the table.

**Details**

Creates a new table.

**Value**

CreateTableRequest object

---

CreateTableResponse    *CreateTableResponse Object*

---

**Description**

CreateTableResponse Object

**Usage**

```
CreateTableResponse(objectId = NULL)
```

**Arguments**

objectId            (string) The object ID of the created table.

**Details**

The result of creating a table.

**Value**

CreateTableResponse object

---

CreateVideoRequest    *CreateVideoRequest Object*

---

**Description**

CreateVideoRequest Object

**Usage**

```
CreateVideoRequest(  
  objectId = NULL,  
  elementProperties = NULL,  
  source = NULL,  
  id = NULL  
)
```

**Arguments**

objectId	(string) A user-supplied object ID. If you specify an ID, it must be unique among all pages and page elements in the presentation. The ID must start with an alphanumeric character or an underscore (matches regex <code>[a-zA-Z0-9_]</code> ); remaining characters may include those as well as a hyphen or colon (matches regex <code>[a-zA-Z0-9_-: ]</code> ). The length of the ID must not be less than 5 or greater than 50. If you don't specify an ID, a unique one is generated.
elementProperties	( <a href="#">PageElementProperties</a> ) The element properties for the video. The <code>PageElementProperties.size</code> property is optional. If you don't specify a size, a default size is chosen by the server. The <code>PageElementProperties.transform</code> property is optional. The transform must not have shear components. If you don't specify a transform, the video will be placed at the top left corner of the page.
source	(string) The video source. <ul style="list-style-type: none"> <li>• SOURCE_UNSPECIFIED: The video source is unspecified.</li> <li>• YOUTUBE: The video source is YouTube.</li> <li>• DRIVE: The video source is Google Drive.</li> </ul>
id	(string) The video source's unique identifier for this video. e.g. For YouTube video <code>https://www.youtube.com/watch?v=7U3axjORYZ0</code> , the ID is <code>7U3axjORYZ0</code> . For a Google Drive video <code>https://drive.google.com/file/d/1xCgQLFTJi5_XI8DgW_lcUYq5e-q6Hi5Q</code> the ID is <code>1xCgQLFTJi5_XI8DgW_lcUYq5e-q6Hi5Q</code> . To access a Google Drive video file, you might need to add a resource key to the HTTP header for a subset of old files. For more information, see <a href="#">Access link-shared files using resource keys</a> .

**Details**

Creates a video. NOTE: Creating a video from Google Drive requires that the requesting app have at least one of the `drive`, `drive.readonly`, or `drive.file` OAuth scopes.

**Value**

CreateVideoRequest object

---

CreateVideoResponse    *CreateVideoResponse Object*

---

**Description**

CreateVideoResponse Object

**Usage**

CreateVideoResponse(objectId = NULL)

**Arguments**

objectId (string) The object ID of the created video.

**Details**

The result of creating a video.

**Value**

CreateVideoResponse object

---

CropProperties	<i>CropProperties Object</i>
----------------	------------------------------

---

**Description**

CropProperties Object

**Usage**

```
CropProperties(
  leftOffset = NULL,
  rightOffset = NULL,
  topOffset = NULL,
  bottomOffset = NULL,
  angle = NULL
)
```

**Arguments**

leftOffset	(number) The offset specifies the left edge of the crop rectangle that is located to the right of the original bounding rectangle left edge, relative to the object's original width.
rightOffset	(number) The offset specifies the right edge of the crop rectangle that is located to the left of the original bounding rectangle right edge, relative to the object's original width.
topOffset	(number) The offset specifies the top edge of the crop rectangle that is located below the original bounding rectangle top edge, relative to the object's original height.
bottomOffset	(number) The offset specifies the bottom edge of the crop rectangle that is located above the original bounding rectangle bottom edge, relative to the object's original height.
angle	(number) The rotation angle of the crop window around its center, in radians. Rotation angle is applied after the offset.

**Details**

The crop properties of an object enclosed in a container. For example, an Image. The crop properties is represented by the offsets of four edges which define a crop rectangle. The offsets are measured in percentage from the corresponding edges of the object's original bounding rectangle towards inside, relative to the object's original dimensions.

- If the offset is in the interval (0, 1), the corresponding edge of crop rectangle is positioned inside of the object's original bounding rectangle.
- If the offset is negative or greater than 1, the corresponding edge of crop rectangle is positioned outside of the object's original bounding rectangle.
- If the left edge of the crop rectangle is on the right side of its right edge, the object will be flipped horizontally.
- If the top edge of the crop rectangle is below its bottom edge, the object will be flipped vertically.
- If all offsets and rotation angle is 0, the object is not cropped. After cropping, the content in the crop rectangle will be stretched to fit its container.

**Value**

CropProperties object

---

DeleteObjectRequest    *DeleteObjectRequest Object*

---

**Description**

DeleteObjectRequest Object

**Usage**

DeleteObjectRequest(objectId = NULL)

**Arguments**

objectId	(string) The object ID of the page or page element to delete. If after a delete operation a group contains only 1 or no page elements, the group is also deleted. If a placeholder is deleted on a layout, any empty inheriting placeholders are also deleted.
----------	--

**Details**

Deletes an object, either pages or page elements, from the presentation.

**Value**

DeleteObjectRequest object

---

DeleteParagraphBulletsRequest

*DeleteParagraphBulletsRequest Object*

---

### Description

DeleteParagraphBulletsRequest Object

### Usage

```
DeleteParagraphBulletsRequest(  
    objectId = NULL,  
    cellLocation = NULL,  
    textRange = NULL  
)
```

### Arguments

objectId	(string) The object ID of the shape or table containing the text to delete bullets from.
cellLocation	( <a href="#">TableCellLocation</a> ) The optional table cell location if the text to be modified is in a table cell. If present, the object_id must refer to a table.
textRange	( <a href="#">Range</a> ) The range of text to delete bullets from, based on TextElement indexes.

### Details

Deletes bullets from all of the paragraphs that overlap with the given text index range. The nesting level of each paragraph will be visually preserved by adding indent to the start of the corresponding paragraph.

### Value

DeleteParagraphBulletsRequest object

---

DeleteTableColumnRequest

*DeleteTableColumnRequest Object*

---

### Description

DeleteTableColumnRequest Object

### Usage

```
DeleteTableColumnRequest(tableObjectId = NULL, cellLocation = NULL)
```

**Arguments**

- tableObjectId (string) The table to delete columns from.
- cellLocation ([TableCellLocation](#)) The reference table cell location from which a column will be deleted. The column this cell spans will be deleted. If this is a merged cell, multiple columns will be deleted. If no columns remain in the table after this deletion, the whole table is deleted.

**Details**

Deletes a column from a table.

**Value**

DeleteTableColumnRequest object

---

DeleteTableRowRequest *DeleteTableRowRequest Object*

---

**Description**

DeleteTableRowRequest Object

**Usage**

```
DeleteTableRowRequest(tableObjectId = NULL, cellLocation = NULL)
```

**Arguments**

- tableObjectId (string) The table to delete rows from.
- cellLocation ([TableCellLocation](#)) The reference table cell location from which a row will be deleted. The row this cell spans will be deleted. If this is a merged cell, multiple rows will be deleted. If no rows remain in the table after this deletion, the whole table is deleted.

**Details**

Deletes a row from a table.

**Value**

DeleteTableRowRequest object



---

DeleteTextRequest      *DeleteTextRequest Object*


---

**Description**

DeleteTextRequest Object

**Usage**

DeleteTextRequest(objectId = NULL, cellLocation = NULL, textRange = NULL)

**Arguments**

objectId	(string) The object ID of the shape or table from which the text will be deleted.
cellLocation	( <a href="#">TableCellLocation</a> ) The optional table cell location if the text is to be deleted from a table cell. If present, the object_id must refer to a table.
textRange	( <a href="#">Range</a> ) The range of text to delete, based on TextElement indexes. There is always an implicit newline character at the end of a shape's or table cell's text that cannot be deleted. Range.Type.ALL will use the correct bounds, but care must be taken when specifying explicit bounds for range types FROM_START_INDEX and FIXED_RANGE. For example, if the text is "ABC", followed by an implicit newline, then the maximum value is 2 for text_range.start_index and 3 for text_range.end_index. Deleting text that crosses a paragraph boundary may result in changes to paragraph styles and lists as the two paragraphs are merged. Ranges that include only one code unit of a surrogate pair are expanded to include both code units.

**Details**

Deletes text from a shape or a table cell.

**Value**

DeleteTextRequest object

---

Dimension      *Dimension Object*


---

**Description**

Dimension Object

**Usage**

Dimension(magnitude = NULL, unit = NULL)

**Arguments**

magnitude	(number) The magnitude.
unit	(string) The units for magnitude. <ul style="list-style-type: none"> <li>• UNIT_UNSPECIFIED: The units are unknown.</li> <li>• EMU: An English Metric Unit (EMU) is defined as 1/360,000 of a centimeter and thus there are 914,400 EMUs per inch, and 12,700 EMUs per point.</li> <li>• PT: A point, 1/72 of an inch.</li> </ul>

**Details**

A magnitude in a single direction in the specified units.

**Value**

Dimension object

---

DuplicateObjectRequest

*DuplicateObjectRequest Object*

---

**Description**

DuplicateObjectRequest Object

**Usage**

DuplicateObjectRequest(objectId = NULL, objectIds = NULL)

**Arguments**

objectId	(string) The ID of the object to duplicate.
objectIds	(named list of <a href="#">string</a> objects) The object being duplicated may contain other objects, for example when duplicating a slide or a group page element. This map defines how the IDs of duplicated objects are generated: the keys are the IDs of the original objects and its values are the IDs that will be assigned to the corresponding duplicate object. The ID of the source object's duplicate may be specified in this map as well, using the same value of the <code>object_id</code> field as a key and the newly desired ID as the value. All keys must correspond to existing IDs in the presentation. All values must be unique in the presentation and must start with an alphanumeric character or an underscore (matches regex <code>[a-zA-Z0-9_]</code> ); remaining characters may include those as well as a hyphen or colon (matches regex <code>[a-zA-Z0-9_-:]</code> ). The length of the new ID must not be less than 5 or greater than 50. If any IDs of source objects are omitted from the map, a new random ID will be assigned. If the map is empty or unset, all

**Details**

Duplicates a slide or page element. When duplicating a slide, the duplicate slide will be created immediately following the specified slide. When duplicating a page element, the duplicate will be placed on the same page at the same position as the original.

**Value**

DuplicateObjectRequest object

---

DuplicateObjectResponse  
*DuplicateObjectResponse Object*

---

**Description**

DuplicateObjectResponse Object

**Usage**

DuplicateObjectResponse(objectId = NULL)

**Arguments**

objectId (string) The ID of the new duplicate object.

**Details**

The response of duplicating an object.

**Value**

DuplicateObjectResponse object

---

Group *Group Object*

---

**Description**

Group Object

**Usage**

Group(children = NULL)

**Arguments**

children (list of [PageElement](#) objects) The collection of elements in the group. The minimum size of a group is 2.

**Details**

A PageElement kind representing a joined collection of PageElements.

**Value**

Group object

---

GroupObjectsRequest *GroupObjectsRequest Object*

---

**Description**

GroupObjectsRequest Object

**Usage**

```
GroupObjectsRequest(groupObjectId = NULL, childrenObjectIds = NULL)
```

**Arguments**

groupObjectId (string) A user-supplied object ID for the group to be created. If you specify an ID, it must be unique among all pages and page elements in the presentation. The ID must start with an alphanumeric character or an underscore (matches regex `[a-zA-Z0-9_]`); remaining characters may include those as well as a hyphen or colon (matches regex `[a-zA-Z0-9_-:]`). The length of the ID must not be less than 5 or greater than 50. If you don't specify an ID, a unique one is generated.

childrenObjectIds (list of [string](#) objects) The object IDs of the objects to group. Only page elements can be grouped. There should be at least two page elements on the same page that are not already in another group. Some page elements, such as videos, tables and placeholders cannot be grouped.

**Details**

Groups objects to create an object group. For example, groups PageElements to create a Group on the same page as all the children.

**Value**

GroupObjectsRequest object

---

GroupObjectsResponse    *GroupObjectsResponse Object*

---

**Description**

GroupObjectsResponse Object

**Usage**

```
GroupObjectsResponse(objectId = NULL)
```

**Arguments**

objectId            (string) The object ID of the created group.

**Details**

The result of grouping objects.

**Value**

GroupObjectsResponse object

---

Image                    *Image Object*

---

**Description**

Image Object

**Usage**

```
Image(  
   contentType = NULL,  
   imageProperties = NULL,  
   sourceUrl = NULL,  
  placeholder = NULL  
)
```

**Arguments**

contentUrl	(string) An URL to an image with a default lifetime of 30 minutes. This URL is tagged with the account of the requester. Anyone with the URL effectively accesses the image as the original requester. Access to the image may be lost if the presentation's sharing settings change.
imageProperties	( <a href="#">ImageProperties</a> ) The properties of the image.
sourceUrl	(string) The source URL is the URL used to insert the image. The source URL can be empty.
placeholder	( <a href="#">Placeholder</a> ) Placeholders are page elements that inherit from corresponding placeholders on layouts and masters. If set, the image is a placeholder image and any inherited properties can be resolved by looking at the parent placeholder identified by the Placeholder.parent_object_id field.

**Details**

A PageElement kind representing an image.

**Value**

Image object

---

ImageProperties	<i>ImageProperties Object</i>
-----------------	-------------------------------

---

**Description**

ImageProperties Object

**Usage**

```
ImageProperties(
  cropProperties = NULL,
  transparency = NULL,
  brightness = NULL,
  contrast = NULL,
  recolor = NULL,
  outline = NULL,
  shadow = NULL,
  link = NULL
)
```

**Arguments**

cropProperties	( <a href="#">CropProperties</a> ) The crop properties of the image. If not set, the image is not cropped. This property is read-only.
transparency	(number) The transparency effect of the image. The value should be in the interval [0.0, 1.0], where 0 means no effect and 1 means completely transparent. This property is read-only.
brightness	(number) The brightness effect of the image. The value should be in the interval [-1.0, 1.0], where 0 means no effect. This property is read-only.
contrast	(number) The contrast effect of the image. The value should be in the interval [-1.0, 1.0], where 0 means no effect. This property is read-only.
recolor	( <a href="#">Recolor</a> ) The recolor effect of the image. If not set, the image is not recolored. This property is read-only.
outline	( <a href="#">Outline</a> ) The outline of the image. If not set, the image has no outline.
shadow	( <a href="#">Shadow</a> ) The shadow of the image. If not set, the image has no shadow. This property is read-only.
link	( <a href="#">Link</a> ) The hyperlink destination of the image. If unset, there is no link.

**Details**

The properties of the Image.

**Value**

ImageProperties object

---

InsertTableColumnsRequest

*InsertTableColumnsRequest Object*

---

**Description**

InsertTableColumnsRequest Object

**Usage**

```
InsertTableColumnsRequest(  
    tableObjectId = NULL,  
    cellLocation = NULL,  
    insertRight = NULL,  
    number = NULL  
)
```

**Arguments**

tableObjectId	(string) The table to insert columns into.
cellLocation	( <a href="#">TableCellLocation</a> ) The reference table cell location from which columns will be inserted. A new column will be inserted to the left (or right) of the column where the reference cell is. If the reference cell is a merged cell, a new column will be inserted to the left (or right) of the merged cell.
insertRight	(boolean) Whether to insert new columns to the right of the reference cell location. - True: insert to the right. - False: insert to the left.
number	(integer) The number of columns to be inserted. Maximum 20 per request.

**Details**

Inserts columns into a table. Other columns in the table will be resized to fit the new column.

**Value**

InsertTableColumnsRequest object

---

InsertTableRowsRequest

*InsertTableRowsRequest Object*

---

**Description**

InsertTableRowsRequest Object

**Usage**

```

InsertTableRowsRequest(
    tableObjectId = NULL,
    cellLocation = NULL,
    insertBelow = NULL,
    number = NULL
)

```

**Arguments**

tableObjectId	(string) The table to insert rows into.
cellLocation	( <a href="#">TableCellLocation</a> ) The reference table cell location from which rows will be inserted. A new row will be inserted above (or below) the row where the reference cell is. If the reference cell is a merged cell, a new row will be inserted above (or below) the merged cell.
insertBelow	(boolean) Whether to insert new rows below the reference cell location. - True: insert below the cell. - False: insert above the cell.
number	(integer) The number of rows to be inserted. Maximum 20 per request.



**Details**

Inserts rows into a table.

**Value**

InsertTableRowsRequest object

---

InsertTextRequest	<i>InsertTextRequest Object</i>
-------------------	---------------------------------

---

**Description**

InsertTextRequest Object

**Usage**

```
InsertTextRequest(
  objectId = NULL,
  cellLocation = NULL,
  text = NULL,
  insertionIndex = NULL
)
```

**Arguments**

objectId	(string) The object ID of the shape or table where the text will be inserted.
cellLocation	( <a href="#">TableCellLocation</a> ) The optional table cell location if the text is to be inserted into a table cell. If present, the object_id must refer to a table.
text	(string) The text to be inserted. Inserting a newline character will implicitly create a new ParagraphMarker at that index. The paragraph style of the new paragraph will be copied from the paragraph at the current insertion index, including lists and bullets. Text styles for inserted text will be determined automatically, generally preserving the styling of neighboring text. In most cases, the text will be added to the TextRun that exists at the insertion index. Some control characters (U+0000-U+0008, U+000C-U+001F) and characters from the Unicode Basic Multilingual Plane Private Use Area (U+E000-U+F8FF) will be stripped out of the inserted text.
insertionIndex	(integer) The index where the text will be inserted, in Unicode code units, based on TextElement indexes. The index is zero-based and is computed from the start of the string. The index may be adjusted to prevent insertions inside Unicode grapheme clusters. In these cases, the text will be inserted immediately after the grapheme cluster.

**Details**

Inserts text into a shape or a table cell.

**Value**

InsertTextRequest object

---

ladder_auth	<i>Authorize ladder</i>
-------------	-------------------------

---

**Description**

Authorize ladder.api to view and manage your presentations. This function is a wrapper around [gargle::token\\_fetch\(\)](#).

By default, you are directed to a web browser, asked to sign in to your Google account, and to grant ladder.api permission to operate on your behalf with Google Slides. By default, with your permission, these user credentials are cached in a folder below your home directory, from where they can be automatically refreshed, as necessary. Storage at the user level means the same token can be used across multiple projects and tokens are less likely to be synced to the cloud by accident.

**Usage**

```
ladder_auth(
  email = gargle::gargle_oauth_email(),
  path = NULL,
  subject = NULL,
  scopes = c("https://www.googleapis.com/auth/drive.file",
            "https://www.googleapis.com/auth/presentations.currentonly"),
  cache = gargle::gargle_oauth_cache(),
  use_oob = gargle::gargle_oob_default(),
  token = NULL
)
```

**Arguments**

email                   Optional. If specified, email can take several different forms:

- "jane@gmail.com", i.e. an actual email address. This allows the user to target a specific Google identity. If specified, this is used for token lookup, i.e. to determine if a suitable token is already available in the cache. If no such token is found, email is used to pre-select the targeted Google identity in the OAuth chooser. (Note, however, that the email associated with a token when it's cached is always determined from the token itself, never from this argument).
- "\*@example.com", i.e. a domain-only glob pattern. This can be helpful if you need code that "just works" for both alice@example.com and bob@example.com.
- TRUE means that you are approving email auto-discovery. If exactly one matching token is found in the cache, it will be used.
- FALSE or NA mean that you want to ignore the token cache and force a new OAuth dance in the browser.

	Defaults to the option named "gargle_oauth_email", retrieved by <code>gargle_oauth_email()</code> (unless a wrapper package implements different default behavior).
path	JSON identifying the service account, in one of the forms supported for the <code>txt</code> argument of <code>jsonlite::fromJSON()</code> (typically, a file path or JSON string).
subject	An optional subject claim. Specify this if you wish to use the service account represented by <code>path</code> to impersonate the subject, who is a normal user. Before this can work, an administrator must grant the service account domain-wide authority. Identify the user to impersonate via their email, e.g. <code>subject = "user@example.com"</code> . Note that gargle automatically adds the non-sensitive "https://www.googleapis.com/auth/userinfo.email" scope, so this scope must be enabled for the service account, along with any other scopes being requested.
scopes	A character vector of scopes to request. Pick from those listed at <a href="https://developers.google.com/identity/protocols/oauth2/scopes">https://developers.google.com/identity/protocols/oauth2/scopes</a> .
cache	Specifies the OAuth token cache. Defaults to the option named "gargle_oauth_cache", retrieved via <code>gargle_oauth_cache()</code> .
use_oob	Whether to use out-of-band authentication (or, perhaps, a variant implemented by gargle and known as "pseudo-OOB") when first acquiring the token. Defaults to the value returned by <code>gargle_oob_default()</code> . Note that (pseudo-)OOB auth only affects the initial OAuth dance. If we retrieve (and possibly refresh) a cached token, <code>use_oob</code> has no effect.  If the OAuth client is provided implicitly by a wrapper package, its type probably defaults to the value returned by <code>gargle_oauth_client_type()</code> . You can take control of the client type by setting <code>options(gargle_oauth_client_type = "web")</code> or <code>options(gargle_oauth_client_type = "installed")</code> .
token	A token with class <code>Token2.0</code> or an object of <code>httr</code> 's class <code>request</code> , i.e. a token that has been prepared with <code>httr::config()</code> and has a <code>Token2.0</code> in the <code>auth_token</code> component.

## Details

Most users, most of the time, do not need to call `ladder_auth()` explicitly – it is triggered by the first action that requires authorization. Even when called, the default arguments often suffice.

However, when necessary, `ladder_auth()` allows the user to explicitly:

- Declare which Google identity to use, via an email specification.
- Use a service account token or workload identity federation via `path`.
- Bring your own token.
- Customize scopes.
- Use a non-default cache folder or turn caching off.
- Explicitly request out-of-band (OOB) auth via `use_oob`.

If you are interacting with R within a browser (applies to RStudio Server, Posit Workbench, Posit Cloud, and Google Colaboratory), you need OOB auth or the pseudo-OOB variant. If this does not happen automatically, you can request it explicitly with `use_oob = TRUE` or, more persistently, by setting an option via `options(gargle_oob_default = TRUE)`.

The choice between conventional OOB or pseudo-OOB auth is determined by the type of OAuth client. If the client is of the "installed" type, `use_oob = TRUE` results in conventional OOB auth. If the client is of the "web" type, `use_oob = TRUE` results in pseudo-OOB auth. Packages that provide a built-in OAuth client can usually detect which type of client to use. But if you need to set this explicitly, use the "gargle\_oauth\_client\_type" option:

```
options(gargle_oauth_client_type = "web")      # pseudo-OOB
# or, alternatively
options(gargle_oauth_client_type = "installed") # conventional OOB
```

For details on the many ways to find a token, see [gargle::token\\_fetch\(\)](#). For deeper control over auth, use [ladder\\_auth\\_configure\(\)](#) to bring your own OAuth client or API key. To learn more about gargle options, see [gargle::gargle\\_options](#).

### See Also

Other auth functions: [ladder\\_auth\\_configure\(\)](#), [ladder\\_deauth\(\)](#)

### Examples

```
# load/refresh existing credentials, if available
# otherwise, go to browser for authentication and authorization
ladder_auth()

# see user associated with current token
ladder_user()

# force use of a token associated with a specific email
ladder_auth(email = "jenny@example.com")
ladder_user()

# force the OAuth web dance
ladder_auth(email = NA)

# use a 'read only' scope, so it's impossible to edit or delete files
ladder_auth(scopes = "slides.readonly")

# use a service account token
ladder_auth(path = "foofy-83ee9e7c9c48.json")
```

## Description

These functions give more control over and visibility into the auth configuration than `ladder_auth()` does. `ladder_auth_configure()` lets the user specify their own:

- OAuth client, which is used when obtaining a user token.
- API key. If `ladder.api` is de-authorized via `ladder_deauth()`, all requests are sent with an API key in lieu of a token.

See the vignette("get-api-credentials", package = "gargle") for more. If the user does not configure these settings, internal defaults are used.

`ladder_oauth_client()` and `ladder_api_key()` retrieve the currently configured OAuth client and API key, respectively.

## Usage

```
ladder_auth_configure(client, path, api_key, app)
```

```
ladder_api_key()
```

```
ladder_oauth_client()
```

## Arguments

<code>client</code>	A Google OAuth client, presumably constructed via <code>gargle::gargle_oauth_client_from_json()</code> . Note, however, that it is preferred to specify the client with JSON, using the <code>path</code> argument.
<code>path</code>	JSON downloaded from <a href="#">Google Cloud Console</a> , containing a client id and secret, in one of the forms supported for the <code>txt</code> argument of <code>jsonlite::fromJSON()</code> (typically, a file path or JSON string).
<code>api_key</code>	API key.
<code>app</code>	<b>[Deprecated]</b> Replaced by the <code>client</code> argument.

## Value

- `ladder_auth_configure()`: An object of R6 class `gargle::AuthState`, invisibly.
- `ladder_oauth_client()`: the current user-configured OAuth client.
- `ladder_api_key()`: the current user-configured API key.

## See Also

Other auth functions: `ladder_auth()`, `ladder_deauth()`

## Examples

```
# see and store the current user-configured OAuth client (probably `NULL`)  
(original_client <- ladder_oauth_client())  
  
# see and store the current user-configured API key (probably `NULL`)
```

```
(original_api_key <- ladder_api_key())

# the preferred way to configure your own client is via a JSON file
# downloaded from Google Developers Console
# this example JSON is indicative, but fake
path_to_json <- system.file(
  "extdata", "client_secret_installed.googleusercontent.com.json",
  package = "gargle"
)
ladder_auth_configure(path = path_to_json)

# this is also obviously a fake API key
ladder_auth_configure(api_key = "the_key_I_got_for_a_google_API")

# confirm the changes
ladder_oauth_client()
ladder_api_key()

# restore original auth config
ladder_auth_configure(client = original_client, api_key = original_api_key)
```

---

ladder\_deauth

*Suspend authorization*

---

## Description

Put ladder.api into a de-authorized state. Instead of sending a token, ladder.api will send an API key. This can be used to access public resources for which no Google sign-in is required. This is handy for using ladder.api in a non-interactive setting to make requests that do not require a token. It will prevent the attempt to obtain a token interactively in the browser. The user can configure their own API key via `ladder_auth_configure()` and retrieve that key via `ladder_api_key()`. In the absence of a user-configured key, a built-in default key is used.

## Usage

```
ladder_deauth()
```

## See Also

Other auth functions: `ladder_auth()`, `ladder_auth_configure()`

## Examples

```
## Not run:
ladder_deauth()
ladder_user()

## End(Not run)
```

---

ladder_has_token	<i>Is there a token on hand?</i>
------------------	----------------------------------

---

**Description**

Reports whether ladder.api has stored a token, ready for use in downstream requests.

**Usage**

```
ladder_has_token()
```

**Value**

Logical.

**See Also**

Other low-level API functions: [ladder\\_token\(\)](#)

**Examples**

```
ladder_has_token()
```

---

ladder_token	<i>Produce configured token</i>
--------------	---------------------------------

---

**Description**

For internal use or for those programming around the Slides API. Returns a token pre-processed with [httr::config\(\)](#). Most users do not need to handle tokens "by hand" or, even if they need some control, [ladder\\_auth\(\)](#) is what they need. If there is no current token, [ladder\\_auth\(\)](#) is called to either load from cache or initiate OAuth2.0 flow. If auth has been deactivated via [ladder\\_deauth\(\)](#), [ladder\\_token\(\)](#) returns NULL.

**Usage**

```
ladder_token()
```

**Value**

A request object (an S3 class provided by [httr](#)).

**See Also**

Other low-level API functions: [ladder\\_has\\_token\(\)](#)

**Examples**

```
## Not run:
ladder_token()

## End(Not run)
```

---

ladder_user	<i>Get info on current user</i>
-------------	---------------------------------

---

**Description**

Reveals the email address of the user associated with the current token. If no token has been loaded yet, this function does not initiate auth.

**Usage**

```
ladder_user()
```

**Value**

An email address or, if no token has been loaded, NULL.

**See Also**

[gargle::token\\_userinfo\(\)](#), [gargle::token\\_email\(\)](#), [gargle::token\\_tokeninfo\(\)](#)

**Examples**

```
## Not run:
ladder_user()

## End(Not run)
```

---

LayoutPlaceholderIdMapping	<i>LayoutPlaceholderIdMapping Object</i>
----------------------------	--

---

**Description**

LayoutPlaceholderIdMapping Object

**Usage**

```
LayoutPlaceholderIdMapping(
  layoutPlaceholder = NULL,
  layoutPlaceholderObjectId = NULL,
  objectId = NULL
)
```



**Arguments**

layoutPlaceholder	(Placeholder) The placeholder on a layout that will be applied to a slide. Only type and index are needed. For example, a predefined TITLE_AND_BODY layout may usually have a TITLE placeholder with index 0 and a BODY placeholder with index 0.
layoutPlaceholderObjectId	(string) The object ID of the placeholder on a layout that will be applied to a slide.
objectId	(string) A user-supplied object ID for the placeholder identified above that to be created onto a slide. If you specify an ID, it must be unique among all pages and page elements in the presentation. The ID must start with an alphanumeric character or an underscore (matches regex <code>[a-zA-Z0-9_]</code> ); remaining characters may include those as well as a hyphen or colon (matches regex <code>[a-zA-Z0-9_-: ]</code> ). The length of the ID must not be less than 5 or greater than 50. If you don't specify an ID, a unique one is generated.

**Details**

The user-specified ID mapping for a placeholder that will be created on a slide from a specified layout.

**Value**

LayoutPlaceholderIdMapping object

---

LayoutProperties	<i>LayoutProperties Object</i>
------------------	--------------------------------

---

**Description**

LayoutProperties Object

**Usage**

LayoutProperties(masterObjectId = NULL, name = NULL, displayName = NULL)

**Arguments**

masterObjectId	(string) The object ID of the master that this layout is based on.
name	(string) The name of the layout.
displayName	(string) The human-readable name of the layout.

**Details**

The properties of Page are only relevant for pages with page\_type LAYOUT.

**Value**

LayoutProperties object

---

LayoutReference	<i>LayoutReference Object</i>
-----------------	-------------------------------

---

**Description**

LayoutReference Object

**Usage**

LayoutReference(predefinedLayout = NULL, layoutId = NULL)

**Arguments**

predefinedLayout

(string) Predefined layout.

- PREDEFINED\_LAYOUT\_UNSPECIFIED: Unspecified layout.
- BLANK: Blank layout, with no placeholders.
- CAPTION\_ONLY: Layout with a caption at the bottom.
- TITLE: Layout with a title and a subtitle.
- TITLE\_AND\_BODY: Layout with a title and body.
- TITLE\_AND\_TWO\_COLUMNS: Layout with a title and two columns.
- TITLE\_ONLY: Layout with only a title.
- SECTION\_HEADER: Layout with a section title.
- SECTION\_TITLE\_AND\_DESCRIPTION: Layout with a title and subtitle on one side and description on the other.
- ONE\_COLUMN\_TEXT: Layout with one title and one body, arranged in a single column.
- MAIN\_POINT: Layout with a main point.
- BIG\_NUMBER: Layout with a big number heading.

layoutId

(string) Layout ID: the object ID of one of the layouts in the presentation.

**Details**

Slide layout reference. This may reference either:

- A predefined layout - One of the layouts in the presentation.

**Value**

LayoutReference object

---

 Line

*Line Object*


---

**Description**

Line Object

**Usage**

```
Line(lineProperties = NULL, lineType = NULL, lineCategory = NULL)
```

**Arguments**

- `lineProperties` ([LineProperties](#)) The properties of the line.
- `lineType` (string) The type of the line.
- `TYPE_UNSPECIFIED`: An unspecified line type.
  - `STRAIGHT_CONNECTOR_1`: Straight connector 1 form. Corresponds to ECMA-376 `ST_ShapeType 'straightConnector1'`.
  - `BENT_CONNECTOR_2`: Bent connector 2 form. Corresponds to ECMA-376 `ST_ShapeType 'bentConnector2'`.
  - `BENT_CONNECTOR_3`: Bent connector 3 form. Corresponds to ECMA-376 `ST_ShapeType 'bentConnector3'`.
  - `BENT_CONNECTOR_4`: Bent connector 4 form. Corresponds to ECMA-376 `ST_ShapeType 'bentConnector4'`.
  - `BENT_CONNECTOR_5`: Bent connector 5 form. Corresponds to ECMA-376 `ST_ShapeType 'bentConnector5'`.
  - `CURVED_CONNECTOR_2`: Curved connector 2 form. Corresponds to ECMA-376 `ST_ShapeType 'curvedConnector2'`.
  - `CURVED_CONNECTOR_3`: Curved connector 3 form. Corresponds to ECMA-376 `ST_ShapeType 'curvedConnector3'`.
  - `CURVED_CONNECTOR_4`: Curved connector 4 form. Corresponds to ECMA-376 `ST_ShapeType 'curvedConnector4'`.
  - `CURVED_CONNECTOR_5`: Curved connector 5 form. Corresponds to ECMA-376 `ST_ShapeType 'curvedConnector5'`.
  - `STRAIGHT_LINE`: Straight line. Corresponds to ECMA-376 `ST_ShapeType 'line'`. This line type is not a connector.
- `lineCategory` (string) The category of the line. It matches the category specified in `CreateLineRequest`, and can be updated with `UpdateLineCategoryRequest`.
- `LINE_CATEGORY_UNSPECIFIED`: Unspecified line category.
  - `STRAIGHT`: Straight connectors, including straight connector 1.
  - `BENT`: Bent connectors, including bent connector 2 to 5.
  - `CURVED`: Curved connectors, including curved connector 2 to 5.

**Details**

A PageElement kind representing a non-connector line, straight connector, curved connector, or bent connector.

**Value**

Line object

---

LineConnection	<i>LineConnection Object</i>
----------------	------------------------------

---

**Description**

LineConnection Object

**Usage**

```
LineConnection(connectedObjectId = NULL, connectionSiteIndex = NULL)
```

**Arguments**

connectedObjectId

(string) The object ID of the connected page element. Some page elements, such as groups, tables, and lines do not have connection sites and therefore cannot be connected to a connector line.

connectionSiteIndex

(integer) The index of the connection site on the connected page element. In most cases, it corresponds to the predefined connection site index from the ECMA-376 standard. More information on those connection sites can be found in the description of the "cnx" attribute in section 20.1.9.9 and Annex H. "Predefined DrawingML Shape and Text Geometries" of "Office Open XML File Formats-Fundamentals and Markup Language Reference", part 1 of [ECMA-376 5th edition](#). The position of each connection site can also be viewed from Slides editor.

**Details**

The properties for one end of a Line connection.

**Value**

LineConnection object

---

LineFill

*LineFill Object*

---

**Description**

LineFill Object

**Usage**

```
LineFill(solidFill = NULL)
```

**Arguments**

solidFill      ([SolidFill](#)) Solid color fill.

**Details**

The fill of the line.

**Value**

LineFill object

---

LineProperties

*LineProperties Object*

---

**Description**

LineProperties Object

**Usage**

```
LineProperties(  
  lineFill = NULL,  
  weight = NULL,  
  dashStyle = NULL,  
  startArrow = NULL,  
  endArrow = NULL,  
  link = NULL,  
  startConnection = NULL,  
  endConnection = NULL  
)
```

**Arguments**

lineFill	( <b>LineFill</b> ) The fill of the line. The default line fill matches the defaults for new lines created in the Slides editor.
weight	( <b>Dimension</b> ) The thickness of the line.
dashStyle	(string) The dash style of the line. <ul style="list-style-type: none"> <li>DASH_STYLE_UNSPECIFIED: Unspecified dash style.</li> <li>SOLID: Solid line. Corresponds to ECMA-376 ST_PresetLineDashVal value 'solid'. This is the default dash style.</li> <li>DOT: Dotted line. Corresponds to ECMA-376 ST_PresetLineDashVal value 'dot'.</li> <li>DASH: Dashed line. Corresponds to ECMA-376 ST_PresetLineDashVal value 'dash'.</li> <li>DASH_DOT: Alternating dashes and dots. Corresponds to ECMA-376 ST_PresetLineDashVal value 'dashDot'.</li> <li>LONG_DASH: Line with large dashes. Corresponds to ECMA-376 ST_PresetLineDashVal value 'lgDash'.</li> <li>LONG_DASH_DOT: Alternating large dashes and dots. Corresponds to ECMA-376 ST_PresetLineDashVal value 'lgDashDot'.</li> </ul>
startArrow	(string) The style of the arrow at the beginning of the line. <ul style="list-style-type: none"> <li>ARROW_STYLE_UNSPECIFIED: An unspecified arrow style.</li> <li>NONE: No arrow.</li> <li>STEALTH_ARROW: Arrow with notched back. Corresponds to ECMA-376 ST_LineEndType value 'stealth'.</li> <li>FILL_ARROW: Filled arrow. Corresponds to ECMA-376 ST_LineEndType value 'triangle'.</li> <li>FILL_CIRCLE: Filled circle. Corresponds to ECMA-376 ST_LineEndType value 'oval'.</li> <li>FILL_SQUARE: Filled square.</li> <li>FILL_DIAMOND: Filled diamond. Corresponds to ECMA-376 ST_LineEndType value 'diamond'.</li> <li>OPEN_ARROW: Hollow arrow.</li> <li>OPEN_CIRCLE: Hollow circle.</li> <li>OPEN_SQUARE: Hollow square.</li> <li>OPEN_DIAMOND: Hollow diamond.</li> </ul>
endArrow	(string) The style of the arrow at the end of the line. <ul style="list-style-type: none"> <li>ARROW_STYLE_UNSPECIFIED: An unspecified arrow style.</li> <li>NONE: No arrow.</li> <li>STEALTH_ARROW: Arrow with notched back. Corresponds to ECMA-376 ST_LineEndType value 'stealth'.</li> <li>FILL_ARROW: Filled arrow. Corresponds to ECMA-376 ST_LineEndType value 'triangle'.</li> <li>FILL_CIRCLE: Filled circle. Corresponds to ECMA-376 ST_LineEndType value 'oval'.</li> </ul>

	<ul style="list-style-type: none"> <li>• FILL_SQUARE: Filled square.</li> <li>• FILL_DIAMOND: Filled diamond. Corresponds to ECMA-376 ST_LineEndType value 'diamond'.</li> <li>• OPEN_ARROW: Hollow arrow.</li> <li>• OPEN_CIRCLE: Hollow circle.</li> <li>• OPEN_SQUARE: Hollow square.</li> <li>• OPEN_DIAMOND: Hollow diamond.</li> </ul>
link	( <a href="#">Link</a> ) The hyperlink destination of the line. If unset, there is no link.
startConnection	( <a href="#">LineConnection</a> ) The connection at the beginning of the line. If unset, there is no connection. Only lines with a Type indicating it is a "connector" can have a start_connection.
endConnection	( <a href="#">LineConnection</a> ) The connection at the end of the line. If unset, there is no connection. Only lines with a Type indicating it is a "connector" can have an end_connection.

### Details

The properties of the Line. When unset, these fields default to values that match the appearance of new lines created in the Slides editor.

### Value

LineProperties object

---

Link	<i>Link Object</i>
------	--------------------

---

### Description

Link Object

### Usage

Link(url = NULL, relativeLink = NULL, pageObjectId = NULL, slideIndex = NULL)

### Arguments

url	(string) If set, indicates this is a link to the external web page at this URL.
relativeLink	(string) If set, indicates this is a link to a slide in this presentation, addressed by its position. <ul style="list-style-type: none"> <li>• RELATIVE_SLIDE_LINK_UNSPECIFIED: An unspecified relative slide link.</li> <li>• NEXT_SLIDE: A link to the next slide.</li> <li>• PREVIOUS_SLIDE: A link to the previous slide.</li> <li>• FIRST_SLIDE: A link to the first slide in the presentation.</li> </ul>

- LAST\_SLIDE: A link to the last slide in the presentation.

pageObjectId	(string) If set, indicates this is a link to the specific page in this presentation with this ID. A page with this ID may not exist.
slideIndex	(integer) If set, indicates this is a link to the slide at this zero-based index in the presentation. There may not be a slide at this index.

**Details**

A hypertext link.

**Value**

Link object

---

List

*List Object*

---

**Description**

List Object

**Usage**

```
List(listId = NULL, nestingLevel = NULL)
```

**Arguments**

listId	(string) The ID of the list.
nestingLevel	(named list of <a href="#">NestingLevel</a> objects) A map of nesting levels to the properties of bullets at the associated level. A list has at most nine levels of nesting, so the possible values for the keys of this map are 0 through 8, inclusive.

**Details**

A List describes the look and feel of bullets belonging to paragraphs associated with a list. A paragraph that is part of a list has an implicit reference to that list's ID.

**Value**

List object



---

MasterProperties	<i>MasterProperties Object</i>
------------------	--------------------------------

---

**Description**

MasterProperties Object

**Usage**

```
MasterProperties(displayName = NULL)
```

**Arguments**

displayName (string) The human-readable name of the master.

**Details**

The properties of Page that are only relevant for pages with page\_type MASTER.

**Value**

MasterProperties object

---

MergeTableCellsRequest	<i>MergeTableCellsRequest Object</i>
------------------------	--------------------------------------

---

**Description**

MergeTableCellsRequest Object

**Usage**

```
MergeTableCellsRequest(objectId = NULL, tableRange = NULL)
```

**Arguments**

objectId (string) The object ID of the table.

tableRange ([TableRange](#)) The table range specifying which cells of the table to merge. Any text in the cells being merged will be concatenated and stored in the upper-left ("head") cell of the range. If the range is non-rectangular (which can occur in some cases where the range covers cells that are already merged), a 400 bad request error is returned.

**Details**

Merges cells in a Table.

**Value**

MergeTableCellsRequest object

---

NestingLevel	<i>NestingLevel Object</i>
--------------	----------------------------

---

**Description**

NestingLevel Object

**Usage**

NestingLevel(bulletStyle = NULL)

**Arguments**

bulletStyle     ([TextStyle](#)) The style of a bullet at this level of nesting.

**Details**

Contains properties describing the look and feel of a list bullet at a given level of nesting.

**Value**

NestingLevel object

---

NotesProperties	<i>NotesProperties Object</i>
-----------------	-------------------------------

---

**Description**

NotesProperties Object

**Usage**

NotesProperties(speakerNotesObjectId = NULL)

**Arguments**

speakerNotesObjectId

(string) The object ID of the shape on this notes page that contains the speaker notes for the corresponding slide. The actual shape may not always exist on the notes page. Inserting text using this object ID will automatically create the shape. In this case, the actual shape may have different object ID. The GetPresentation or GetPage action will always return the latest object ID.

**Details**

The properties of Page that are only relevant for pages with page\_type NOTES.

**Value**

NotesProperties object

---

OpaqueColor	<i>OpaqueColor Object</i>
-------------	---------------------------

---

**Description**

OpaqueColor Object

**Usage**

OpaqueColor(rgbaColor = NULL, themeColor = NULL)

**Arguments**

rgbaColor      ([RgbaColor](#)) An opaque RGBA color.

themeColor    (string) An opaque theme color.

- THEME\_COLOR\_TYPE\_UNSPECIFIED: Unspecified theme color. This value should not be used.
- DARK1: Represents the first dark color.
- LIGHT1: Represents the first light color.
- DARK2: Represents the second dark color.
- LIGHT2: Represents the second light color.
- ACCENT1: Represents the first accent color.
- ACCENT2: Represents the second accent color.
- ACCENT3: Represents the third accent color.
- ACCENT4: Represents the fourth accent color.
- ACCENT5: Represents the fifth accent color.
- ACCENT6: Represents the sixth accent color.
- HYPERLINK: Represents the color to use for hyperlinks.
- FOLLOWED\_HYPERLINK: Represents the color to use for visited hyperlinks.

- TEXT1: Represents the first text color.
- BACKGROUND1: Represents the first background color.
- TEXT2: Represents the second text color.
- BACKGROUND2: Represents the second background color.

**Details**

A themeable solid color value.

**Value**

OpaqueColor object

---

OptionalColor	<i>OptionalColor Object</i>
---------------	-----------------------------

---

**Description**

OptionalColor Object

**Usage**

```
OptionalColor(opaqueColor = NULL)
```

**Arguments**

opaqueColor     ([OpaqueColor](#)) If set, this will be used as an opaque color. If unset, this represents a transparent color.

**Details**

A color that can either be fully opaque or fully transparent.

**Value**

OptionalColor object

---

 Outline

 Outline Object
 

---

**Description**

Outline Object

**Usage**

```
Outline(
  outlineFill = NULL,
  weight = NULL,
  dashStyle = NULL,
  propertyState = NULL
)
```

**Arguments**

- |               |   |
|---------------|---|
| outlineFill   | ( <a href="#">OutlineFill</a> ) The fill of the outline.  |
| weight        | ( <a href="#">Dimension</a> ) The thickness of the outline.   |
| dashStyle     | (string) The dash style of the outline. <ul style="list-style-type: none"> <li>• DASH_STYLE_UNSPECIFIED: Unspecified dash style.</li> <li>• SOLID: Solid line. Corresponds to ECMA-376 ST_PresetLineDashVal value 'solid'. This is the default dash style.</li> <li>• DOT: Dotted line. Corresponds to ECMA-376 ST_PresetLineDashVal value 'dot'.</li> <li>• DASH: Dashed line. Corresponds to ECMA-376 ST_PresetLineDashVal value 'dash'.</li> <li>• DASH_DOT: Alternating dashes and dots. Corresponds to ECMA-376 ST_PresetLineDashVal value 'dashDot'.</li> <li>• LONG_DASH: Line with large dashes. Corresponds to ECMA-376 ST_PresetLineDashVal value 'lgDash'.</li> <li>• LONG_DASH_DOT: Alternating large dashes and dots. Corresponds to ECMA-376 ST_PresetLineDashVal value 'lgDashDot'.</li> </ul>   |
| propertyState | (string) The outline property state. Updating the outline on a page element will implicitly update this field to RENDERED, unless another value is specified in the same request. To have no outline on a page element, set this field to NOT_RENDERED. In this case, any other outline fields set in the same request will be ignored. <ul style="list-style-type: none"> <li>• RENDERED: If a property's state is RENDERED, then the element has the corresponding property when rendered on a page. If the element is a placeholder shape as determined by the placeholder field, and it inherits from a placeholder shape, the corresponding field may be unset, meaning that the property value is inherited from a parent placeholder. If the element does not inherit, then the field will contain the rendered value. This is the default value.</li> </ul> |

- NOT\_RENDERED: If a property's state is NOT\_RENDERED, then the element does not have the corresponding property when rendered on a page. However, the field may still be set so it can be inherited by child shapes. To remove a property from a rendered element, set its property\_state to NOT\_RENDERED.
- INHERIT: If a property's state is INHERIT, then the property state uses the value of corresponding property\_state field on the parent shape. Elements that do not inherit will never have an INHERIT property state.

### Details

The outline of a PageElement. If these fields are unset, they may be inherited from a parent placeholder if it exists. If there is no parent, the fields will default to the value used for new page elements created in the Slides editor, which may depend on the page element kind.

### Value

Outline object

---

OutlineFill

*OutlineFill Object*

---

### Description

OutlineFill Object

### Usage

```
OutlineFill(solidFill = NULL)
```

### Arguments

solidFill      ([SolidFill](#)) Solid color fill.

### Details

The fill of the outline.

### Value

OutlineFill object

---

 Page

*Page Object*


---

**Description**

Page Object

**Usage**

```

Page(
  objectId = NULL,
  pageType = NULL,
  pageElements = NULL,
  slideProperties = NULL,
  layoutProperties = NULL,
  notesProperties = NULL,
  masterProperties = NULL,
  revisionId = NULL,
  pageProperties = NULL
)

```

**Arguments**

objectId	(string) The object ID for this page. Object IDs used by Page and PageElement share the same namespace.
pageType	(string) The type of the page. <ul style="list-style-type: none"> <li>• SLIDE: A slide page.</li> <li>• MASTER: A master slide page.</li> <li>• LAYOUT: A layout page.</li> <li>• NOTES: A notes page.</li> <li>• NOTES_MASTER: A notes master page.</li> </ul>
pageElements	(list of <a href="#">PageElement</a> objects) The page elements rendered on the page.
slideProperties	( <a href="#">SlideProperties</a> ) Slide specific properties. Only set if page_type = SLIDE.
layoutProperties	( <a href="#">LayoutProperties</a> ) Layout specific properties. Only set if page_type = LAYOUT.
notesProperties	( <a href="#">NotesProperties</a> ) Notes specific properties. Only set if page_type = NOTES.
masterProperties	( <a href="#">MasterProperties</a> ) Master specific properties. Only set if page_type = MASTER.

revisionId	(string) Output only. The revision ID of the presentation. Can be used in update requests to assert the presentation revision hasn't changed since the last read operation. Only populated if the user has edit access to the presentation. The revision ID is not a sequential number but an opaque string. The format of the revision ID might change over time. A returned revision ID is only guaranteed to be valid for 24 hours after it has been returned and cannot be shared across users. If the revision ID is unchanged between calls, then the presentation has not changed. Conversely, a changed ID (for the same presentation and user) usually means the presentation has been updated. However, a changed ID can also be due to internal factors such as ID format changes.
pageProperties	( <a href="#">PageProperties</a> ) The properties of the page.

### Details

A page in a presentation.

### Value

Page object

---

PageBackgroundFill     *PageBackgroundFill Object*

---

### Description

PageBackgroundFill Object

### Usage

```
PageBackgroundFill(
  propertyState = NULL,
  solidFill = NULL,
  stretchedPictureFill = NULL
)
```

### Arguments

propertyState	(string) The background fill property state. Updating the fill on a page will implicitly update this field to RENDERED, unless another value is specified in the same request. To have no fill on a page, set this field to NOT_RENDERED. In this case, any other fill fields set in the same request will be ignored. <ul style="list-style-type: none"> <li>RENDERED: If a property's state is RENDERED, then the element has the corresponding property when rendered on a page. If the element is a placeholder shape as determined by the placeholder field, and it inherits from a placeholder shape, the corresponding field may be unset, meaning that the property value is inherited from a parent placeholder. If the element does not inherit, then the field will contain the rendered value. This is the default value.</li> </ul>
---------------	--



- NOT\_RENDERED: If a property's state is NOT\_RENDERED, then the element does not have the corresponding property when rendered on a page. However, the field may still be set so it can be inherited by child shapes. To remove a property from a rendered element, set its property\_state to NOT\_RENDERED.
- INHERIT: If a property's state is INHERIT, then the property state uses the value of corresponding property\_state field on the parent shape. Elements that do not inherit will never have an INHERIT property state.

solidFill (SolidFill) Solid color fill.

stretchedPictureFill

(StretchedPictureFill) Stretched picture fill.

### Details

The page background fill.

### Value

PageBackgroundFill object

---

PageElement

*PageElement Object*

---

### Description

PageElement Object

### Usage

```
PageElement(
  objectId = NULL,
  size = NULL,
  transform = NULL,
  title = NULL,
  description = NULL,
  elementGroup = NULL,
  shape = NULL,
  image = NULL,
  video = NULL,
  line = NULL,
  table = NULL,
  wordArt = NULL,
  sheetsChart = NULL,
  speakerSpotlight = NULL
)
```

**Arguments**

objectId	(string) The object ID for this page element. Object IDs used by google.apps.slides.v1.Page and google.apps.slides.v1.PageElement share the same namespace.
size	(Size) The size of the page element.
transform	(AffineTransform) The transform of the page element. The visual appearance of the page element is determined by its absolute transform. To compute the absolute transform, preconcatenate a page element's transform with the transforms of all of its parent groups. If the page element is not in a group, its absolute transform is the same as the value in this field. The initial transform for the newly created Group is always the identity transform.
title	(string) The title of the page element. Combined with description to display alt text. The field is not supported for Group elements.
description	(string) The description of the page element. Combined with title to display alt text. The field is not supported for Group elements.
elementGroup	(Group) A collection of page elements joined as a single unit.
shape	(Shape) A generic shape.
image	(Image) An image page element.
video	(Video) A video page element.
line	(Line) A line page element.
table	(Table) A table page element.
wordArt	(WordArt) A word art page element.
sheetsChart	(SheetsChart) A linked chart embedded from Google Sheets. Unlinked charts are represented as images.
speakerSpotlight	(SpeakerSpotlight) A Speaker Spotlight.

**Details**

A visual element rendered on a page.

**Value**

PageElement object

---

PageElementProperties *PageElementProperties Object*

---

**Description**

PageElementProperties Object

**Usage**

PageElementProperties(pageObjectId = NULL, size = NULL, transform = NULL)

**Arguments**

pageObjectId	(string) The object ID of the page where the element is located.
size	( <a href="#">Size</a> ) The size of the element.
transform	( <a href="#">AffineTransform</a> ) The transform for the element.

**Details**

Common properties for a page element. Note: When you initially create a PageElement, the API may modify the values of both size and transform, but the visual size will be unchanged.

**Value**

PageElementProperties object

---

PageProperties	<i>PageProperties Object</i>
----------------	------------------------------

---

**Description**

PageProperties Object

**Usage**

```
PageProperties(pageBackgroundFill = NULL, colorScheme = NULL)
```

**Arguments**

pageBackgroundFill	( <a href="#">PageBackgroundFill</a> ) The background fill of the page. If unset, the background fill is inherited from a parent page if it exists. If the page has no parent, then the background fill defaults to the corresponding fill in the Slides editor.
colorScheme	( <a href="#">ColorScheme</a> ) The color scheme of the page. If unset, the color scheme is inherited from a parent page. If the page has no parent, the color scheme uses a default Slides color scheme, matching the defaults in the Slides editor. Only the concrete colors of the first 12 ThemeColorTypes are editable. In addition, only the color scheme on Master pages can be updated. To update the field, a color scheme containing mappings from all the first 12 ThemeColorTypes to their concrete colors must be provided. Colors for the remaining ThemeColorTypes will be ignored.

**Details**

The properties of the Page. The page will inherit properties from the parent page. Depending on the page type the hierarchy is defined in either SlideProperties or LayoutProperties.

**Value**

PageProperties object

---

ParagraphMarker	<i>ParagraphMarker Object</i>
-----------------	-------------------------------

---

**Description**

ParagraphMarker Object

**Usage**

```
ParagraphMarker(style = NULL, bullet = NULL)
```

**Arguments**

style	( <a href="#">ParagraphStyle</a> ) The paragraph's style
bullet	( <a href="#">Bullet</a> ) The bullet for this paragraph. If not present, the paragraph does not belong to a list.

**Details**

A TextElement kind that represents the beginning of a new paragraph.

**Value**

ParagraphMarker object

---

ParagraphStyle	<i>ParagraphStyle Object</i>
----------------	------------------------------

---

**Description**

ParagraphStyle Object

**Usage**

```
ParagraphStyle(
  lineSpacing = NULL,
  alignment = NULL,
  indentStart = NULL,
  indentEnd = NULL,
  spaceAbove = NULL,
  spaceBelow = NULL,
  indentFirstLine = NULL,
  direction = NULL,
  spacingMode = NULL
)
```

**Arguments**

lineSpacing	(number) The amount of space between lines, as a percentage of normal, where normal is represented as 100.0. If unset, the value is inherited from the parent.
alignment	(string) The text alignment for this paragraph. <ul style="list-style-type: none"> <li>• ALIGNMENT_UNSPECIFIED: The paragraph alignment is inherited from the parent.</li> <li>• START: The paragraph is aligned to the start of the line. Left-aligned for LTR text, right-aligned otherwise.</li> <li>• CENTER: The paragraph is centered.</li> <li>• END: The paragraph is aligned to the end of the line. Right-aligned for LTR text, left-aligned otherwise.</li> <li>• JUSTIFIED: The paragraph is justified.</li> </ul>
indentStart	(Dimension) The amount indentation for the paragraph on the side that corresponds to the start of the text, based on the current text direction. If unset, the value is inherited from the parent.
indentEnd	(Dimension) The amount indentation for the paragraph on the side that corresponds to the end of the text, based on the current text direction. If unset, the value is inherited from the parent.
spaceAbove	(Dimension) The amount of extra space above the paragraph. If unset, the value is inherited from the parent.
spaceBelow	(Dimension) The amount of extra space below the paragraph. If unset, the value is inherited from the parent.
indentFirstLine	(Dimension) The amount of indentation for the start of the first line of the paragraph. If unset, the value is inherited from the parent.
direction	(string) The text direction of this paragraph. If unset, the value defaults to LEFT_TO_RIGHT since text direction is not inherited. <ul style="list-style-type: none"> <li>• TEXT_DIRECTION_UNSPECIFIED: The text direction is inherited from the parent.</li> <li>• LEFT_TO_RIGHT: The text goes from left to right.</li> <li>• RIGHT_TO_LEFT: The text goes from right to left.</li> </ul>
spacingMode	(string) The spacing mode for the paragraph. <ul style="list-style-type: none"> <li>• SPACING_MODE_UNSPECIFIED: The spacing mode is inherited from the parent.</li> <li>• NEVER_COLLAPSE: Paragraph spacing is always rendered.</li> <li>• COLLAPSE_LISTS: Paragraph spacing is skipped between list elements.</li> </ul>

**Details**

Styles that apply to a whole paragraph. If this text is contained in a shape with a parent placeholder, then these paragraph styles may be inherited from the parent. Which paragraph styles are inherited depend on the nesting level of lists:

- A paragraph not in a list will inherit its paragraph style from the paragraph at the 0 nesting level of the list inside the parent placeholder.

- A paragraph in a list will inherit its paragraph style from the paragraph at its corresponding nesting level of the list inside the parent placeholder. Inherited paragraph styles are represented as unset fields in this message.

### Value

ParagraphStyle object

---

Placeholder

*Placeholder Object*

---

### Description

Placeholder Object

### Usage

Placeholder(type = NULL, index = NULL, parentObjectId = NULL)

### Arguments

type	<p>(string) The type of the placeholder.</p> <ul style="list-style-type: none"> <li>• NONE: Default value, signifies it is not a placeholder.</li> <li>• BODY: Body text.</li> <li>• CHART: Chart or graph.</li> <li>• CLIP_ART: Clip art image.</li> <li>• CENTERED_TITLE: Title centered.</li> <li>• DIAGRAM: Diagram.</li> <li>• DATE_AND_TIME: Date and time.</li> <li>• FOOTER: Footer text.</li> <li>• HEADER: Header text.</li> <li>• MEDIA: Multimedia.</li> <li>• OBJECT: Any content type.</li> <li>• PICTURE: Picture.</li> <li>• SLIDE_NUMBER: Number of a slide.</li> <li>• SUBTITLE: Subtitle.</li> <li>• TABLE: Table.</li> <li>• TITLE: Slide title.</li> <li>• SLIDE_IMAGE: Slide image.</li> </ul>
index	<p>(integer) The index of the placeholder. If the same placeholder types are present in the same page, they would have different index values.</p>
parentObjectId	<p>(string) The object ID of this shape's parent placeholder. If unset, the parent placeholder shape does not exist, so the shape does not inherit properties from any other shape.</p>

**Details**

The placeholder information that uniquely identifies a placeholder shape.

**Value**

Placeholder object

---

Presentation

*Presentation Object*

---

**Description**

Presentation Object

**Usage**

```
Presentation(
  presentationId = NULL,
  pageSize = NULL,
  slides = NULL,
  title = NULL,
  masters = NULL,
  layouts = NULL,
  locale = NULL,
  revisionId = NULL,
  notesMaster = NULL
)
```

**Arguments**

presentationId	(string) The ID of the presentation.
pageSize	( <a href="#">Size</a> ) The size of pages in the presentation.
slides	(list of <a href="#">Page</a> objects) The slides in the presentation. A slide inherits properties from a slide layout.
title	(string) The title of the presentation.
masters	(list of <a href="#">Page</a> objects) The slide masters in the presentation. A slide master contains all common page elements and the common properties for a set of layouts. They serve three purposes: - Placeholder shapes on a master contain the default text styles and shape properties of all placeholder shapes on pages that use that master. - The master page properties define the common page properties inherited by its layouts. - Any other shapes on the master slide appear on all slides using that master, regardless of their layout.
layouts	(list of <a href="#">Page</a> objects) The layouts in the presentation. A layout is a template that determines how content is arranged and styled on the slides that inherit from that layout.

locale	(string) The locale of the presentation, as an IETF BCP 47 language tag.
revisionId	(string) Output only. The revision ID of the presentation. Can be used in update requests to assert the presentation revision hasn't changed since the last read operation. Only populated if the user has edit access to the presentation. The revision ID is not a sequential number but a nebulous string. The format of the revision ID may change over time, so it should be treated opaquely. A returned revision ID is only guaranteed to be valid for 24 hours after it has been returned and cannot be shared across users. If the revision ID is unchanged between calls, then the presentation has not changed. Conversely, a changed ID (for the same presentation and user) usually means the presentation has been updated. However, a changed ID can also be due to internal factors such as ID format changes.
notesMaster	(Page) The notes master in the presentation. It serves three purposes: - Placeholder shapes on a notes master contain the default text styles and shape properties of all placeholder shapes on notes pages. Specifically, a SLIDE_IMAGE placeholder shape contains the slide thumbnail, and a BODY placeholder shape contains the speaker notes. - The notes master page properties define the common page properties inherited by all notes pages. - Any other shapes on the notes master appear on all notes pages. The notes master is read-only.

### Details

A Google Slides presentation.

### Value

Presentation object

---

presentations.batchUpdate

*Applies one or more updates to the presentation.*

---

### Description

Each request is validated before being applied. If any request is not valid, then the entire request will fail and nothing will be applied. Some requests have replies to give you some information about how they are applied. Other requests do not need to return information; these each return an empty reply. The order of replies matches that of the requests. For example, suppose you call batchUpdate with four updates, and only the third one returns information. The response would have two empty replies: the reply to the third request, and another empty reply, in that order. Because other users may be editing the presentation, the presentation might not exactly reflect your changes: your changes may be altered with respect to collaborator changes. If there are no collaborators, the presentation should reflect your changes. In any case, the updates in your request are guaranteed to be applied together atomically.

### Usage

presentations.batchUpdate(BatchUpdatePresentationRequest, presentationId)



**Arguments**

BatchUpdatePresentationRequest

The [BatchUpdatePresentationRequest](#) object to pass to this method

presentationId The presentation to apply the updates to

**See Also**

[Google Documentation](#)

---

presentations.create *Creates a blank presentation using the title given in the request.*

---

**Description**

If a presentationId is provided, it is used as the ID of the new presentation. Otherwise, a new ID is generated. Other fields in the request, including any provided content, are ignored. Returns the created presentation.

**Usage**

```
presentations.create(Presentation)
```

**Arguments**

Presentation The [Presentation](#) object to pass to this method

**See Also**

[Google Documentation](#)

---

presentations.get *Gets the latest version of the specified presentation.*

---

**Description**

Gets the latest version of the specified presentation.

**Usage**

```
presentations.get(presentationId)
```

**Arguments**

presentationId The ID of the presentation to retrieve

**See Also**

[Google Documentation](#)

presentations.pages.get

*Gets the latest version of the specified page in the presentation.*

---

### Description

Gets the latest version of the specified page in the presentation.

### Usage

```
presentations.pages.get(presentationId, pageObjectId)
```

### Arguments

presentationId The ID of the presentation to retrieve  
pageObjectId The object ID of the page to retrieve

### See Also

[Google Documentation](#)

---

presentations.pages.getThumbnail

*Generates a thumbnail of the latest version of the specified page.*

---

### Description

Generates a thumbnail of the latest version of the specified page in the presentation and returns a URL to the thumbnail image. This request counts as an [expensive read request](#) for quota purposes.

### Usage

```
presentations.pages.getThumbnail(  
  presentationId,  
  pageObjectId,  
  thumbnailProperties.thumbnailSize = NULL,  
  thumbnailProperties.mimeType = NULL  
)
```

### Arguments

presentationId The ID of the presentation to retrieve  
pageObjectId The object ID of the page whose thumbnail to retrieve  
thumbnailProperties.thumbnailSize  
The optional thumbnail image size  
thumbnailProperties.mimeType  
The optional mime type of the thumbnail image

**See Also**

[Google Documentation](#)

---

Range

*Range Object*

---

**Description**

Range Object

**Usage**

```
Range(startIndex = NULL, endIndex = NULL, type = NULL)
```

**Arguments**

startIndex	(integer) The optional zero-based index of the beginning of the collection. Required for FIXED_RANGE and FROM_START_INDEX ranges.
endIndex	(integer) The optional zero-based index of the end of the collection. Required for FIXED_RANGE ranges.
type	(string) The type of range. <ul style="list-style-type: none"><li>• RANGE_TYPE_UNSPECIFIED: Unspecified range type. This value must not be used.</li><li>• FIXED_RANGE: A fixed range. Both the start_index and end_index must be specified.</li><li>• FROM_START_INDEX: Starts the range at start_index and continues until the end of the collection. The end_index must not be specified.</li><li>• ALL: Sets the range to be the whole length of the collection. Both the start_index and the end_index must not be specified.</li></ul>

**Details**

Specifies a contiguous range of an indexed collection, such as characters in text.

**Value**

Range object

Recolor

*Recolor Object***Description**

Recolor Object

**Usage**

Recolor(recolorStops = NULL, name = NULL)

**Arguments**

- |              |  |
|--------------|--|
| recolorStops | (list of <a href="#">ColorStop</a> objects) The recolor effect is represented by a gradient, which is a list of color stops. The colors in the gradient will replace the corresponding colors at the same position in the color palette and apply to the image. This property is read-only.  |
| name         | (string) The name of the recolor effect. The name is determined from the <code>recolor_stops</code> by matching the gradient against the colors in the page's current color scheme. This property is read-only. <ul style="list-style-type: none"> <li>• NONE: No recolor effect. The default value.</li> <li>• LIGHT1: A recolor effect that lightens the image using the page's first available color from its color scheme.</li> <li>• LIGHT2: A recolor effect that lightens the image using the page's second available color from its color scheme.</li> <li>• LIGHT3: A recolor effect that lightens the image using the page's third available color from its color scheme.</li> <li>• LIGHT4: A recolor effect that lightens the image using the page's fourth available color from its color scheme.</li> <li>• LIGHT5: A recolor effect that lightens the image using the page's fifth available color from its color scheme.</li> <li>• LIGHT6: A recolor effect that lightens the image using the page's sixth available color from its color scheme.</li> <li>• LIGHT7: A recolor effect that lightens the image using the page's seventh available color from its color scheme.</li> <li>• LIGHT8: A recolor effect that lightens the image using the page's eighth available color from its color scheme.</li> <li>• LIGHT9: A recolor effect that lightens the image using the page's ninth available color from its color scheme.</li> <li>• LIGHT10: A recolor effect that lightens the image using the page's tenth available color from its color scheme.</li> <li>• DARK1: A recolor effect that darkens the image using the page's first available color from its color scheme.</li> <li>• DARK2: A recolor effect that darkens the image using the page's second available color from its color scheme.</li> </ul> |

- DARK3: A recolor effect that darkens the image using the page's third available color from its color scheme.
- DARK4: A recolor effect that darkens the image using the page's fourth available color from its color scheme.
- DARK5: A recolor effect that darkens the image using the page's fifth available color from its color scheme.
- DARK6: A recolor effect that darkens the image using the page's sixth available color from its color scheme.
- DARK7: A recolor effect that darkens the image using the page's seventh available color from its color scheme.
- DARK8: A recolor effect that darkens the image using the page's eighth available color from its color scheme.
- DARK9: A recolor effect that darkens the image using the page's ninth available color from its color scheme.
- DARK10: A recolor effect that darkens the image using the page's tenth available color from its color scheme.
- GRAYSCALE: A recolor effect that recolors the image to grayscale.
- NEGATIVE: A recolor effect that recolors the image to negative grayscale.
- SEPIA: A recolor effect that recolors the image using the sepia color.
- CUSTOM: Custom recolor effect. Refer to `recolor_stops` for the concrete gradient.

**Details**

A recolor effect applied on an image.

**Value**

Recolor object

---

RefreshSheetsChartRequest

*RefreshSheetsChartRequest Object*

---

**Description**

RefreshSheetsChartRequest Object

**Usage**

RefreshSheetsChartRequest(objectId = NULL)

**Arguments**

objectId (string) The object ID of the chart to refresh.

**Details**

Refreshes an embedded Google Sheets chart by replacing it with the latest version of the chart from Google Sheets. NOTE: Refreshing charts requires at least one of the spreadsheets.readonly, spreadsheets, drive.readonly, or drive OAuth scopes.

**Value**

RefreshSheetsChartRequest object

---

ReplaceAllShapesWithImageRequest

*ReplaceAllShapesWithImageRequest Object*

---

**Description**

ReplaceAllShapesWithImageRequest Object

**Usage**

```
ReplaceAllShapesWithImageRequest(
  containsText = NULL,
  imageUrl = NULL,
  replaceMethod = NULL,
  imageReplaceMethod = NULL,
  pageObjectIds = NULL
)
```

**Arguments**

- |               |   |
|---------------|---|
| containsText  | ( <a href="#">SubstringMatchCriteria</a> ) If set, this request will replace all of the shapes that contain the given text.   |
| imageUrl      | (string) The image URL. The image is fetched once at insertion time and a copy is stored for display inside the presentation. Images must be less than 50MB in size, cannot exceed 25 megapixels, and must be in one of PNG, JPEG, or GIF format. The provided URL can be at most 2 kB in length. The URL itself is saved with the image, and exposed via the Image.source_url field.   |
| replaceMethod | (string) The replace method. <i>Deprecated</i> : use image_replace_method instead. If you specify both a replace_method and an image_replace_method, the image_replace_method takes precedence. <ul style="list-style-type: none"> <li>• CENTER_INSIDE: Scales and centers the image to fit within the bounds of the original shape and maintains the image's aspect ratio. The rendered size of the image may be smaller than the size of the shape. This is the default method when one is not specified.</li> <li>• CENTER_CROP: Scales and centers the image to fill the bounds of the original shape. The image may be cropped in order to fill the shape. The rendered size of the image will be the same as that of the original shape.</li> </ul> |

**imageReplaceMethod**

(string) The image replace method. If you specify both a `replace_method` and an `image_replace_method`, the `image_replace_method` takes precedence. If you do not specify a value for `image_replace_method`, but specify a value for `replace_method`, then the specified `replace_method` value is used. If you do not specify either, then `CENTER_INSIDE` is used.

- `IMAGE_REPLACE_METHOD_UNSPECIFIED`: Unspecified image replace method. This value must not be used.
- `CENTER_INSIDE`: Scales and centers the image to fit within the bounds of the original shape and maintains the image's aspect ratio. The rendered size of the image may be smaller than the size of the shape. This is the default method when one is not specified.
- `CENTER_CROP`: Scales and centers the image to fill the bounds of the original shape. The image may be cropped in order to fill the shape. The rendered size of the image will be the same as the original shape.

**pageObjectIds** (list of `string` objects) If non-empty, limits the matches to page elements only on the given pages. Returns a 400 bad request error if given the page object ID of a notes page or a notes master, or if a page with that object ID doesn't exist in the presentation.

**Details**

Replaces all shapes that match the given criteria with the provided image. The images replacing the shapes are rectangular after being inserted into the presentation and do not take on the forms of the shapes.

**Value**

ReplaceAllShapesWithImageRequest object

---

ReplaceAllShapesWithImageResponse

*ReplaceAllShapesWithImageResponse Object*

---

**Description**

ReplaceAllShapesWithImageResponse Object

**Usage**

ReplaceAllShapesWithImageResponse(occurrencesChanged = NULL)

**Arguments**

`occurrencesChanged`

(integer) The number of shapes replaced with images.

**Details**

The result of replacing shapes with an image.

**Value**

ReplaceAllShapesWithImageResponse object

---

ReplaceAllShapesWithSheetsChartRequest

*ReplaceAllShapesWithSheetsChartRequest Object*

---

**Description**

ReplaceAllShapesWithSheetsChartRequest Object

**Usage**

```
ReplaceAllShapesWithSheetsChartRequest(
  containsText = NULL,
  spreadsheetId = NULL,
  chartId = NULL,
  linkingMode = NULL,
  pageObjectIds = NULL
)
```

**Arguments**

containsText	( <a href="#">SubstringMatchCriteria</a> ) The criteria that the shapes must match in order to be replaced. The request will replace all of the shapes that contain the given text.
spreadsheetId	(string) The ID of the Google Sheets spreadsheet that contains the chart.
chartId	(integer) The ID of the specific chart in the Google Sheets spreadsheet.
linkingMode	(string) The mode with which the chart is linked to the source spreadsheet. When not specified, the chart will be an image that is not linked. <ul style="list-style-type: none"> <li>NOT_LINKED_IMAGE: The chart is not associated with the source spreadsheet and cannot be updated. A chart that is not linked will be inserted as an image.</li> <li>LINKED: Linking the chart allows it to be updated, and other collaborators will see a link to the spreadsheet.</li> </ul>
pageObjectIds	(list of <a href="#">string</a> objects) If non-empty, limits the matches to page elements only on the given pages. Returns a 400 bad request error if given the page object ID of a notes page or a notes master, or if a page with that object ID doesn't exist in the presentation.



**Details**

Replaces all shapes that match the given criteria with the provided Google Sheets chart. The chart will be scaled and centered to fit within the bounds of the original shape. NOTE: Replacing shapes with a chart requires at least one of the spreadsheets.readonly, spreadsheets, drive.readonly, or drive OAuth scopes.

**Value**

ReplaceAllShapesWithSheetsChartRequest object

---

ReplaceAllShapesWithSheetsChartResponse

*ReplaceAllShapesWithSheetsChartResponse Object*

---

**Description**

ReplaceAllShapesWithSheetsChartResponse Object

**Usage**

ReplaceAllShapesWithSheetsChartResponse(occurrencesChanged = NULL)

**Arguments**

occurrencesChanged  
(integer) The number of shapes replaced with charts.

**Details**

The result of replacing shapes with a Google Sheets chart.

**Value**

ReplaceAllShapesWithSheetsChartResponse object

---

ReplaceAllTextRequest *ReplaceAllTextRequest Object*

---

### Description

ReplaceAllTextRequest Object

### Usage

```
ReplaceAllTextRequest(
    replaceText = NULL,
    pageObjectIds = NULL,
    containsText = NULL
)
```

### Arguments

`replaceText` (string) The text that will replace the matched text.

`pageObjectIds` (list of [string](#) objects) If non-empty, limits the matches to page elements only on the given pages. Returns a 400 bad request error if given the page object ID of a notes master, or if a page with that object ID doesn't exist in the presentation.

`containsText` ([SubstringMatchCriteria](#)) Finds text in a shape matching this substring.

### Details

Replaces all instances of text matching a criteria with replace text.

### Value

ReplaceAllTextRequest object

---

ReplaceAllTextResponse  
*ReplaceAllTextResponse Object*

---

### Description

ReplaceAllTextResponse Object

### Usage

```
ReplaceAllTextResponse(occurrencesChanged = NULL)
```

**Arguments**

occurrencesChanged  
 (integer) The number of occurrences changed by replacing all text.

**Details**

The result of replacing text.

**Value**

ReplaceAllTextResponse object

---

ReplaceImageRequest    *ReplaceImageRequest Object*

---

**Description**

ReplaceImageRequest Object

**Usage**

```
ReplaceImageRequest(
  imageObjectId = NULL,
  url = NULL,
  imageReplaceMethod = NULL
)
```

**Arguments**

imageObjectId (string) The ID of the existing image that will be replaced. The ID can be retrieved from the response of a get request.

url (string) The image URL. The image is fetched once at insertion time and a copy is stored for display inside the presentation. Images must be less than 50MB, cannot exceed 25 megapixels, and must be in PNG, JPEG, or GIF format. The provided URL can't surpass 2 KB in length. The URL is saved with the image, and exposed through the Image.source\_url field.

imageReplaceMethod (string) The replacement method.

- IMAGE\_REPLACE\_METHOD\_UNSPECIFIED: Unspecified image replace method. This value must not be used.
- CENTER\_INSIDE: Scales and centers the image to fit within the bounds of the original shape and maintains the image's aspect ratio. The rendered size of the image may be smaller than the size of the shape. This is the default method when one is not specified.
- CENTER\_CROP: Scales and centers the image to fill the bounds of the original shape. The image may be cropped in order to fill the shape. The rendered size of the image will be the same as the original shape.

**Details**

Replaces an existing image with a new image. Replacing an image removes some image effects from the existing image.

**Value**

ReplaceImageRequest object

---

Request

*Request Object*

---

**Description**

Request Object

**Usage**

Request(...)

**Arguments**

createSlide	( <a href="#">CreateSlideRequest</a> ) Creates a new slide.
createShape	( <a href="#">CreateShapeRequest</a> ) Creates a new shape.
createTable	( <a href="#">CreateTableRequest</a> ) Creates a new table.
insertText	( <a href="#">InsertTextRequest</a> ) Inserts text into a shape or table cell.
insertTableRows	( <a href="#">InsertTableRowsRequest</a> ) Inserts rows into a table.
insertTableColumns	( <a href="#">InsertTableColumnsRequest</a> ) Inserts columns into a table.
deleteTableRow	( <a href="#">DeleteTableRowRequest</a> ) Deletes a row from a table.
deleteTableColumn	( <a href="#">DeleteTableColumnRequest</a> ) Deletes a column from a table.
replaceAllText	( <a href="#">ReplaceAllTextRequest</a> ) Replaces all instances of specified text.
deleteObject	( <a href="#">DeleteObjectRequest</a> ) Deletes a page or page element from the presentation.
updatePageElementTransform	( <a href="#">UpdatePageElementTransformRequest</a> ) Updates the transform of a page element.
updateSlidesPosition	( <a href="#">UpdateSlidesPositionRequest</a> ) Updates the position of a set of slides in the presentation.
deleteText	( <a href="#">DeleteTextRequest</a> ) Deletes text from a shape or a table cell.
createImage	( <a href="#">CreateImageRequest</a> ) Creates an image.
createVideo	( <a href="#">CreateVideoRequest</a> ) Creates a video.

createSheetsChart	( <a href="#">CreateSheetsChartRequest</a> ) Creates an embedded Google Sheets chart.
createLine	( <a href="#">CreateLineRequest</a> ) Creates a line.
refreshSheetsChart	( <a href="#">RefreshSheetsChartRequest</a> ) Refreshes a Google Sheets chart.
updateShapeProperties	( <a href="#">UpdateShapePropertiesRequest</a> ) Updates the properties of a Shape.
updateImageProperties	( <a href="#">UpdateImagePropertiesRequest</a> ) Updates the properties of an Image.
updateVideoProperties	( <a href="#">UpdateVideoPropertiesRequest</a> ) Updates the properties of a Video.
updatePageProperties	( <a href="#">UpdatePagePropertiesRequest</a> ) Updates the properties of a Page.
updateTableCellProperties	( <a href="#">UpdateTableCellPropertiesRequest</a> ) Updates the properties of a TableCell.
updateLineProperties	( <a href="#">UpdateLinePropertiesRequest</a> ) Updates the properties of a Line.
createParagraphBullets	( <a href="#">CreateParagraphBulletsRequest</a> ) Creates bullets for paragraphs.
replaceAllShapesWithImage	( <a href="#">ReplaceAllShapesWithImageRequest</a> ) Replaces all shapes matching some criteria with an image.
duplicateObject	( <a href="#">DuplicateObjectRequest</a> ) Duplicates a slide or page element.
updateTextStyle	( <a href="#">UpdateTextStyleRequest</a> ) Updates the styling of text within a Shape or Table.
replaceAllShapesWithSheetsChart	( <a href="#">ReplaceAllShapesWithSheetsChartRequest</a> ) Replaces all shapes matching some criteria with a Google Sheets chart.
deleteParagraphBullets	( <a href="#">DeleteParagraphBulletsRequest</a> ) Deletes bullets from paragraphs.
updateParagraphStyle	( <a href="#">UpdateParagraphStyleRequest</a> ) Updates the styling of paragraphs within a Shape or Table.
updateTableBorderProperties	( <a href="#">UpdateTableBorderPropertiesRequest</a> ) Updates the properties of the table borders in a Table.
updateTableColumnProperties	( <a href="#">UpdateTableColumnPropertiesRequest</a> ) Updates the properties of a Table column.
updateTableRowProperties	( <a href="#">UpdateTableRowPropertiesRequest</a> ) Updates the properties of a Table row.
mergeTableCells	( <a href="#">MergeTableCellsRequest</a> ) Merges cells in a Table.
unmergeTableCells	( <a href="#">UnmergeTableCellsRequest</a> ) Unmerges cells in a Table.

groupObjects	(GroupObjectsRequest)	Groups objects, such as page elements.
ungroupObjects	(UngroupObjectsRequest)	Ungroups objects, such as groups.
updatePageElementAltText	(UpdatePageElementAltTextRequest)	Updates the alt text title and/or description of a page element.
replaceImage	(ReplaceImageRequest)	Replaces an existing image with a new image.
updateSlideProperties	(UpdateSlidePropertiesRequest)	Updates the properties of a Slide
updatePageElementsZOrder	(UpdatePageElementsZOrderRequest)	Updates the Z-order of page elements.
updateLineCategory	(UpdateLineCategoryRequest)	Updates the category of a line.
rerouteLine	(RerouteLineRequest)	Reroutes a line such that it's connected at the two closest connection sites on the connected page elements.

**Details**

A single kind of update to apply to a presentation.

**Value**

Request object

---

RerouteLineRequest      *RerouteLineRequest Object*

---

**Description**

RerouteLineRequest Object

**Usage**

RerouteLineRequest(objectId = NULL)

**Arguments**

objectId	(string) The object ID of the line to reroute. Only a line with a category indicating it is a "connector" can be rerouted. The start and end connections of the line must be on different page elements.
----------	--

**Details**

Reroutes a line such that it's connected at the two closest connection sites on the connected page elements.

**Value**

RerouteLineRequest object

---

Response	<i>Response Object</i>
----------	------------------------

---

**Description**

Response Object

**Usage**

```
Response(
  createSlide = NULL,
  createShape = NULL,
  createTable = NULL,
  replaceAllText = NULL,
  createImage = NULL,
  createVideo = NULL,
  createSheetsChart = NULL,
  createLine = NULL,
  replaceAllShapesWithImage = NULL,
  duplicateObject = NULL,
  replaceAllShapesWithSheetsChart = NULL,
  groupObjects = NULL
)
```

**Arguments**

createSlide	( <a href="#">CreateSlideResponse</a> ) The result of creating a slide.
createShape	( <a href="#">CreateShapeResponse</a> ) The result of creating a shape.
createTable	( <a href="#">CreateTableResponse</a> ) The result of creating a table.
replaceAllText	( <a href="#">ReplaceAllTextResponse</a> ) The result of replacing text.
createImage	( <a href="#">CreateImageResponse</a> ) The result of creating an image.
createVideo	( <a href="#">CreateVideoResponse</a> ) The result of creating a video.
createSheetsChart	( <a href="#">CreateSheetsChartResponse</a> ) The result of creating a Google Sheets chart.
createLine	( <a href="#">CreateLineResponse</a> ) The result of creating a line.
replaceAllShapesWithImage	( <a href="#">ReplaceAllShapesWithImageResponse</a> ) The result of replacing all shapes matching some criteria with an image.
duplicateObject	( <a href="#">DuplicateObjectResponse</a> ) The result of duplicating an object.
replaceAllShapesWithSheetsChart	( <a href="#">ReplaceAllShapesWithSheetsChartResponse</a> ) The result of replacing all shapes matching some criteria with a Google Sheets chart.
groupObjects	( <a href="#">GroupObjectsResponse</a> ) The result of grouping objects.

**Details**

A single response from an update.

**Value**

Response object

---

RgbColor

*RgbColor Object*

---

**Description**

RgbColor Object

**Usage**

`RgbColor(red = NULL, green = NULL, blue = NULL)`

**Arguments**

red	(number) The red component of the color, from 0.0 to 1.0.
green	(number) The green component of the color, from 0.0 to 1.0.
blue	(number) The blue component of the color, from 0.0 to 1.0.

**Details**

An RGB color.

**Value**

RgbColor object

---

Shadow

*Shadow Object*

---

**Description**

Shadow Object



**Usage**

```
Shadow(
  type = NULL,
  transform = NULL,
  alignment = NULL,
  blurRadius = NULL,
  color = NULL,
  alpha = NULL,
  rotateWithShape = NULL,
  propertyState = NULL
)
```

**Arguments**

type	(string) The type of the shadow. This property is read-only. <ul style="list-style-type: none"> <li>SHADOW_TYPE_UNSPECIFIED: Unspecified shadow type.</li> <li>OUTER: Outer shadow.</li> </ul>
transform	( <a href="#">AffineTransform</a> ) Transform that encodes the translate, scale, and skew of the shadow, relative to the alignment position.
alignment	(string) The alignment point of the shadow, that sets the origin for translate, scale and skew of the shadow. This property is read-only. <ul style="list-style-type: none"> <li>RECTANGLE_POSITION_UNSPECIFIED: Unspecified.</li> <li>TOP_LEFT: Top left.</li> <li>TOP_CENTER: Top center.</li> <li>TOP_RIGHT: Top right.</li> <li>LEFT_CENTER: Left center.</li> <li>CENTER: Center.</li> <li>RIGHT_CENTER: Right center.</li> <li>BOTTOM_LEFT: Bottom left.</li> <li>BOTTOM_CENTER: Bottom center.</li> <li>BOTTOM_RIGHT: Bottom right.</li> </ul>
blurRadius	( <a href="#">Dimension</a> ) The radius of the shadow blur. The larger the radius, the more diffuse the shadow becomes.
color	( <a href="#">OpaqueColor</a> ) The shadow color value.
alpha	(number) The alpha of the shadow's color, from 0.0 to 1.0.
rotateWithShape	(boolean) Whether the shadow should rotate with the shape. This property is read-only.
propertyState	(string) The shadow property state. Updating the shadow on a page element will implicitly update this field to RENDERED, unless another value is specified in the same request. To have no shadow on a page element, set this field to NOT_RENDERED. In this case, any other shadow fields set in the same request will be ignored.

- **RENDERED**: If a property's state is **RENDERED**, then the element has the corresponding property when rendered on a page. If the element is a placeholder shape as determined by the placeholder field, and it inherits from a placeholder shape, the corresponding field may be unset, meaning that the property value is inherited from a parent placeholder. If the element does not inherit, then the field will contain the rendered value. This is the default value.
- **NOT\_RENDERED**: If a property's state is **NOT\_RENDERED**, then the element does not have the corresponding property when rendered on a page. However, the field may still be set so it can be inherited by child shapes. To remove a property from a rendered element, set its `property_state` to **NOT\_RENDERED**.
- **INHERIT**: If a property's state is **INHERIT**, then the property state uses the value of corresponding `property_state` field on the parent shape. Elements that do not inherit will never have an **INHERIT** property state.

### Details

The shadow properties of a page element. If these fields are unset, they may be inherited from a parent placeholder if it exists. If there is no parent, the fields will default to the value used for new page elements created in the Slides editor, which may depend on the page element kind.

### Value

Shadow object

---

Shape	<i>Shape Object</i>
-------	---------------------

---

### Description

Shape Object

### Usage

```
Shape(
  shapeType = NULL,
  text = NULL,
  shapeProperties = NULL,
  placeholder = NULL
)
```

### Arguments

- |                        |   |
|------------------------|---|
| <code>shapeType</code> | (string) The type of the shape. <ul style="list-style-type: none"> <li>• <b>TYPE_UNSPECIFIED</b>: The shape type that is not predefined.</li> <li>• <b>TEXT_BOX</b>: Text box shape.</li> </ul> |
|------------------------|---|

- RECTANGLE: Rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'rect'.
- ROUND\_RECTANGLE: Round corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'roundRect'
- ELLIPSE: Ellipse shape. Corresponds to ECMA-376 ST\_ShapeType 'ellipse'
- ARC: Curved arc shape. Corresponds to ECMA-376 ST\_ShapeType 'arc'
- BENT\_ARROW: Bent arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'bentArrow'
- BENT\_UP\_ARROW: Bent up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'bentUpArrow'
- BEVEL: Bevel shape. Corresponds to ECMA-376 ST\_ShapeType 'bevel'
- BLOCK\_ARC: Block arc shape. Corresponds to ECMA-376 ST\_ShapeType 'blockArc'
- BRACE\_PAIR: Brace pair shape. Corresponds to ECMA-376 ST\_ShapeType 'bracePair'
- BRACKET\_PAIR: Bracket pair shape. Corresponds to ECMA-376 ST\_ShapeType 'bracketPair'
- CAN: Can shape. Corresponds to ECMA-376 ST\_ShapeType 'can'
- CHEVRON: Chevron shape. Corresponds to ECMA-376 ST\_ShapeType 'chevron'
- CHORD: Chord shape. Corresponds to ECMA-376 ST\_ShapeType 'chord'
- CLOUD: Cloud shape. Corresponds to ECMA-376 ST\_ShapeType 'cloud'
- CORNER: Corner shape. Corresponds to ECMA-376 ST\_ShapeType 'corner'
- CUBE: Cube shape. Corresponds to ECMA-376 ST\_ShapeType 'cube'
- CURVED\_DOWN\_ARROW: Curved down arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'curvedDownArrow'
- CURVED\_LEFT\_ARROW: Curved left arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'curvedLeftArrow'
- CURVED\_RIGHT\_ARROW: Curved right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'curvedRightArrow'
- CURVED\_UP\_ARROW: Curved up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'curvedUpArrow'
- DECAGON: Decagon shape. Corresponds to ECMA-376 ST\_ShapeType 'decagon'
- DIAGONAL\_STRIPE: Diagonal stripe shape. Corresponds to ECMA-376 ST\_ShapeType 'diagStripe'
- DIAMOND: Diamond shape. Corresponds to ECMA-376 ST\_ShapeType 'diamond'
- DODECAGON: Dodecagon shape. Corresponds to ECMA-376 ST\_ShapeType 'dodecagon'
- DONUT: Donut shape. Corresponds to ECMA-376 ST\_ShapeType 'donut'
- DOUBLE\_WAVE: Double wave shape. Corresponds to ECMA-376 ST\_ShapeType 'doubleWave'
- DOWN\_ARROW: Down arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'downArrow'

- DOWN\_ARROW\_CALLOUT: Callout down arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'downArrowCallout'
- FOLDED\_CORNER: Folded corner shape. Corresponds to ECMA-376 ST\_ShapeType 'foldedCorner'
- FRAME: Frame shape. Corresponds to ECMA-376 ST\_ShapeType 'frame'
- HALF\_FRAME: Half frame shape. Corresponds to ECMA-376 ST\_ShapeType 'halfFrame'
- HEART: Heart shape. Corresponds to ECMA-376 ST\_ShapeType 'heart'
- HEPTAGON: Heptagon shape. Corresponds to ECMA-376 ST\_ShapeType 'heptagon'
- HEXAGON: Hexagon shape. Corresponds to ECMA-376 ST\_ShapeType 'hexagon'
- HOME\_PLATE: Home plate shape. Corresponds to ECMA-376 ST\_ShapeType 'homePlate'
- HORIZONTAL\_SCROLL: Horizontal scroll shape. Corresponds to ECMA-376 ST\_ShapeType 'horizontalScroll'
- IRREGULAR\_SEAL\_1: Irregular seal 1 shape. Corresponds to ECMA-376 ST\_ShapeType 'irregularSeal1'
- IRREGULAR\_SEAL\_2: Irregular seal 2 shape. Corresponds to ECMA-376 ST\_ShapeType 'irregularSeal2'
- LEFT\_ARROW: Left arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftArrow'
- LEFT\_ARROW\_CALLOUT: Callout left arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftArrowCallout'
- LEFT\_BRACE: Left brace shape. Corresponds to ECMA-376 ST\_ShapeType 'leftBrace'
- LEFT\_BRACKET: Left bracket shape. Corresponds to ECMA-376 ST\_ShapeType 'leftBracket'
- LEFT\_RIGHT\_ARROW: Left right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftRightArrow'
- LEFT\_RIGHT\_ARROW\_CALLOUT: Callout left right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftRightArrowCallout'
- LEFT\_RIGHT\_UP\_ARROW: Left right up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftRightUpArrow'
- LEFT\_UP\_ARROW: Left up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'leftUpArrow'
- LIGHTNING\_BOLT: Lightning bolt shape. Corresponds to ECMA-376 ST\_ShapeType 'lightningBolt'
- MATH\_DIVIDE: Divide math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathDivide'
- MATH\_EQUAL: Equal math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathEqual'
- MATH\_MINUS: Minus math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathMinus'
- MATH\_MULTIPLY: Multiply math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathMultiply'

- MATH\_NOT\_EQUAL: Not equal math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathNotEqual'
- MATH\_PLUS: Plus math shape. Corresponds to ECMA-376 ST\_ShapeType 'mathPlus'
- MOON: Moon shape. Corresponds to ECMA-376 ST\_ShapeType 'moon'
- NO\_SMOKING: No smoking shape. Corresponds to ECMA-376 ST\_ShapeType 'noSmoking'
- NOTCHED\_RIGHT\_ARROW: Notched right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'notchedRightArrow'
- OCTAGON: Octagon shape. Corresponds to ECMA-376 ST\_ShapeType 'octagon'
- PARALLELOGRAM: Parallelogram shape. Corresponds to ECMA-376 ST\_ShapeType 'parallelogram'
- PENTAGON: Pentagon shape. Corresponds to ECMA-376 ST\_ShapeType 'pentagon'
- PIE: Pie shape. Corresponds to ECMA-376 ST\_ShapeType 'pie'
- PLAQUE: Plaque shape. Corresponds to ECMA-376 ST\_ShapeType 'plaque'
- PLUS: Plus shape. Corresponds to ECMA-376 ST\_ShapeType 'plus'
- QUAD\_ARROW: Quad-arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'quadArrow'
- QUAD\_ARROW\_CALLOUT: Callout quad-arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'quadArrowCallout'
- RIBBON: Ribbon shape. Corresponds to ECMA-376 ST\_ShapeType 'ribbon'
- RIBBON\_2: Ribbon 2 shape. Corresponds to ECMA-376 ST\_ShapeType 'ribbon2'
- RIGHT\_ARROW: Right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'rightArrow'
- RIGHT\_ARROW\_CALLOUT: Callout right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'rightArrowCallout'
- RIGHT\_BRACE: Right brace shape. Corresponds to ECMA-376 ST\_ShapeType 'rightBrace'
- RIGHT\_BRACKET: Right bracket shape. Corresponds to ECMA-376 ST\_ShapeType 'rightBracket'
- ROUND\_1\_RECTANGLE: One round corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'round1Rect'
- ROUND\_2\_DIAGONAL\_RECTANGLE: Two diagonal round corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'round2DiagRect'
- ROUND\_2\_SAME\_RECTANGLE: Two same-side round corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'round2SameRect'
- RIGHT\_TRIANGLE: Right triangle shape. Corresponds to ECMA-376 ST\_ShapeType 'rtTriangle'
- SMILEY\_FACE: Smiley face shape. Corresponds to ECMA-376 ST\_ShapeType 'smileyFace'

- SNIP\_1\_RECTANGLE: One snip corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'snip1Rect'
- SNIP\_2\_DIAGONAL\_RECTANGLE: Two diagonal snip corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'snip2DiagRect'
- SNIP\_2\_SAME\_RECTANGLE: Two same-side snip corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'snip2SameRect'
- SNIP\_ROUND\_RECTANGLE: One snip one round corner rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'snipRoundRect'
- STAR\_10: Ten pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star10'
- STAR\_12: Twelve pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star12'
- STAR\_16: Sixteen pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star16'
- STAR\_24: Twenty four pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star24'
- STAR\_32: Thirty two pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star32'
- STAR\_4: Four pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star4'
- STAR\_5: Five pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star5'
- STAR\_6: Six pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star6'
- STAR\_7: Seven pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star7'
- STAR\_8: Eight pointed star shape. Corresponds to ECMA-376 ST\_ShapeType 'star8'
- STRIPED\_RIGHT\_ARROW: Striped right arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'stripedRightArrow'
- SUN: Sun shape. Corresponds to ECMA-376 ST\_ShapeType 'sun'
- TRAPEZOID: Trapezoid shape. Corresponds to ECMA-376 ST\_ShapeType 'trapezoid'
- TRIANGLE: Triangle shape. Corresponds to ECMA-376 ST\_ShapeType 'triangle'
- UP\_ARROW: Up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'upArrow'
- UP\_ARROW\_CALLOUT: Callout up arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'upArrowCallout'
- UP\_DOWN\_ARROW: Up down arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'upDownArrow'
- UTURN\_ARROW: U-turn arrow shape. Corresponds to ECMA-376 ST\_ShapeType 'uturnArrow'
- VERTICAL\_SCROLL: Vertical scroll shape. Corresponds to ECMA-376 ST\_ShapeType 'verticalScroll'

- WAVE: Wave shape. Corresponds to ECMA-376 ST\_ShapeType 'wave'
- WEDGE\_ELLIPSE\_CALLOUT: Callout wedge ellipse shape. Corresponds to ECMA-376 ST\_ShapeType 'wedgeEllipseCallout'
- WEDGE\_RECTANGLE\_CALLOUT: Callout wedge rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'wedgeRectCallout'
- WEDGE\_ROUND\_RECTANGLE\_CALLOUT: Callout wedge round rectangle shape. Corresponds to ECMA-376 ST\_ShapeType 'wedgeRoundRectCallout'
- FLOW\_CHART\_ALTERNATE\_PROCESS: Alternate process flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartAlternateProcess'
- FLOW\_CHART\_COLLATE: Collate flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartCollate'
- FLOW\_CHART\_CONNECTOR: Connector flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartConnector'
- FLOW\_CHART\_DECISION: Decision flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartDecision'
- FLOW\_CHART\_DELAY: Delay flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartDelay'
- FLOW\_CHART\_DISPLAY: Display flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartDisplay'
- FLOW\_CHART\_DOCUMENT: Document flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartDocument'
- FLOW\_CHART\_EXTRACT: Extract flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartExtract'
- FLOW\_CHART\_INPUT\_OUTPUT: Input output flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartInputOutput'
- FLOW\_CHART\_INTERNAL\_STORAGE: Internal storage flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartInternalStorage'
- FLOW\_CHART\_MAGNETIC\_DISK: Magnetic disk flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartMagneticDisk'
- FLOW\_CHART\_MAGNETIC\_DRUM: Magnetic drum flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartMagneticDrum'
- FLOW\_CHART\_MAGNETIC\_TAPE: Magnetic tape flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartMagneticTape'
- FLOW\_CHART\_MANUAL\_INPUT: Manual input flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartManualInput'
- FLOW\_CHART\_MANUAL\_OPERATION: Manual operation flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartManualOperation'
- FLOW\_CHART\_MERGE: Merge flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartMerge'
- FLOW\_CHART\_MULTIDOCUMENT: Multi-document flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartMultidocument'
- FLOW\_CHART\_OFFLINE\_STORAGE: Offline storage flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartOfflineStorage'
- FLOW\_CHART\_OFFPAGE\_CONNECTOR: Off-page connector flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartOffpageConnector'

- FLOW\_CHART\_ONLINE\_STORAGE: Online storage flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartOnlineStorage'
- FLOW\_CHART\_OR: Or flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartOr'
- FLOW\_CHART\_PREDEFINED\_PROCESS: Predefined process flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartPredefinedProcess'
- FLOW\_CHART\_PREPARATION: Preparation flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartPreparation'
- FLOW\_CHART\_PROCESS: Process flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartProcess'
- FLOW\_CHART\_PUNCHED\_CARD: Punched card flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartPunchedCard'
- FLOW\_CHART\_PUNCHED\_TAPE: Punched tape flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartPunchedTape'
- FLOW\_CHART\_SORT: Sort flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartSort'
- FLOW\_CHART\_SUMMING\_JUNCTION: Summing junction flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartSummingJunction'
- FLOW\_CHART\_TERMINATOR: Terminator flow shape. Corresponds to ECMA-376 ST\_ShapeType 'flowChartTerminator'
- ARROW\_EAST: East arrow shape.
- ARROW\_NORTH\_EAST: Northeast arrow shape.
- ARROW\_NORTH: North arrow shape.
- SPEECH: Speech shape.
- STARBURST: Star burst shape.
- TEARDROP: Teardrop shape. Corresponds to ECMA-376 ST\_ShapeType 'teardrop'
- ELLIPSE\_RIBBON: Ellipse ribbon shape. Corresponds to ECMA-376 ST\_ShapeType 'ellipseRibbon'
- ELLIPSE\_RIBBON\_2: Ellipse ribbon 2 shape. Corresponds to ECMA-376 ST\_ShapeType 'ellipseRibbon2'
- CLOUD\_CALLOUT: Callout cloud shape. Corresponds to ECMA-376 ST\_ShapeType 'cloudCallout'
- CUSTOM: Custom shape.

text (TextContent) The text content of the shape.

shapeProperties

(ShapeProperties) The properties of the shape.

placeholder (Placeholder) Placeholders are page elements that inherit from corresponding placeholders on layouts and masters. If set, the shape is a placeholder shape and any inherited properties can be resolved by looking at the parent placeholder identified by the Placeholder.parent\_object\_id field.

## Details

A PageElement kind representing a generic shape that does not have a more specific classification.



**Value**

Shape object

---

 ShapeBackgroundFill    *ShapeBackgroundFill Object*


---

**Description**

ShapeBackgroundFill Object

**Usage**

```
ShapeBackgroundFill(propertyState = NULL, solidFill = NULL)
```

**Arguments**

**propertyState** (string) The background fill property state. Updating the fill on a shape will implicitly update this field to RENDERED, unless another value is specified in the same request. To have no fill on a shape, set this field to NOT\_RENDERED. In this case, any other fill fields set in the same request will be ignored.

- **RENDERED**: If a property's state is RENDERED, then the element has the corresponding property when rendered on a page. If the element is a placeholder shape as determined by the placeholder field, and it inherits from a placeholder shape, the corresponding field may be unset, meaning that the property value is inherited from a parent placeholder. If the element does not inherit, then the field will contain the rendered value. This is the default value.
- **NOT\_RENDERED**: If a property's state is NOT\_RENDERED, then the element does not have the corresponding property when rendered on a page. However, the field may still be set so it can be inherited by child shapes. To remove a property from a rendered element, set its property\_state to NOT\_RENDERED.
- **INHERIT**: If a property's state is INHERIT, then the property state uses the value of corresponding property\_state field on the parent shape. Elements that do not inherit will never have an INHERIT property state.

**solidFill** ([SolidFill](#)) Solid color fill.

**Details**

The shape background fill.

**Value**

ShapeBackgroundFill object

---

ShapeProperties	<i>ShapeProperties Object</i>
-----------------	-------------------------------

---

## Description

ShapeProperties Object

## Usage

```
ShapeProperties(
  shapeBackgroundFill = NULL,
  outline = NULL,
  shadow = NULL,
  link = NULL,
  contentAlignment = NULL,
  autofit = NULL
)
```

## Arguments

- |                     |  |
|---------------------|--|
| shapeBackgroundFill | <b>(ShapeBackgroundFill)</b> The background fill of the shape. If unset, the background fill is inherited from a parent placeholder if it exists. If the shape has no parent, then the default background fill depends on the shape type, matching the defaults for new shapes created in the Slides editor.   |
| outline             | <b>(Outline)</b> The outline of the shape. If unset, the outline is inherited from a parent placeholder if it exists. If the shape has no parent, then the default outline depends on the shape type, matching the defaults for new shapes created in the Slides editor.   |
| shadow              | <b>(Shadow)</b> The shadow properties of the shape. If unset, the shadow is inherited from a parent placeholder if it exists. If the shape has no parent, then the default shadow matches the defaults for new shapes created in the Slides editor. This property is read-only.  |
| link                | <b>(Link)</b> The hyperlink destination of the shape. If unset, there is no link. Links are not inherited from parent placeholders.  |
| contentAlignment    | <p><b>(string)</b> The alignment of the content in the shape. If unspecified, the alignment is inherited from a parent placeholder if it exists. If the shape has no parent, the default alignment matches the alignment for new shapes created in the Slides editor.</p> <ul style="list-style-type: none"> <li>• CONTENT_ALIGNMENT_UNSPECIFIED: An unspecified content alignment. The content alignment is inherited from the parent if it exists.</li> <li>• CONTENT_ALIGNMENT_UNSUPPORTED: An unsupported content alignment.</li> <li>• TOP: An alignment that aligns the content to the top of the content holder. Corresponds to ECMA-376 ST_TextAnchoringType 't'.</li> </ul> |

- MIDDLE: An alignment that aligns the content to the middle of the content holder. Corresponds to ECMA-376 ST\_TextAnchoringType 'ctr'.
  - BOTTOM: An alignment that aligns the content to the bottom of the content holder. Corresponds to ECMA-376 ST\_TextAnchoringType 'b'.
- autofit      ([Autofit](#)) The autofit properties of the shape. This property is only set for shapes that allow text.

### Details

The properties of a Shape. If the shape is a placeholder shape as determined by the placeholder field, then these properties may be inherited from a parent placeholder shape. Determining the rendered value of the property depends on the corresponding property\_state field value. Any text autofit settings on the shape are automatically deactivated by requests that can impact how text fits in the shape.

### Value

ShapeProperties object

---

SheetsChart

*SheetsChart Object*

---

### Description

SheetsChart Object

### Usage

```

SheetsChart(
  spreadsheetId = NULL,
  chartId = NULL,
  contentUrl = NULL,
  sheetsChartProperties = NULL
)

```

### Arguments

- spreadsheetId    (string) The ID of the Google Sheets spreadsheet that contains the source chart.
- chartId            (integer) The ID of the specific chart in the Google Sheets spreadsheet that is embedded.
- contentUrl        (string) The URL of an image of the embedded chart, with a default lifetime of 30 minutes. This URL is tagged with the account of the requester. Anyone with the URL effectively accesses the image as the original requester. Access to the image may be lost if the presentation's sharing settings change.
- sheetsChartProperties    ([SheetsChartProperties](#)) The properties of the Sheets chart.

**Details**

A PageElement kind representing a linked chart embedded from Google Sheets.

**Value**

SheetsChart object

---

SheetsChartProperties *SheetsChartProperties Object*

---

**Description**

SheetsChartProperties Object

**Usage**

SheetsChartProperties(chartImageProperties = NULL)

**Arguments**

chartImageProperties

[\(ImageProperties\)](#) The properties of the embedded chart image.

**Details**

The properties of the SheetsChart.

**Value**

SheetsChartProperties object

---

Size *Size Object*

---

**Description**

Size Object

**Usage**

Size(width = NULL, height = NULL)

**Arguments**

width [\(Dimension\)](#) The width of the object.

height [\(Dimension\)](#) The height of the object.

**Details**

A width and height.

**Value**

Size object

---

SlideProperties	<i>SlideProperties Object</i>
-----------------	-------------------------------

---

**Description**

SlideProperties Object

**Usage**

```
SlideProperties(
    layoutObjectId = NULL,
    masterObjectId = NULL,
    notesPage = NULL,
    isSkipped = NULL
)
```

**Arguments**

layoutObjectId	(string) The object ID of the layout that this slide is based on. This property is read-only.
masterObjectId	(string) The object ID of the master that this slide is based on. This property is read-only.
notesPage	<a href="#">(Page)</a> The notes page that this slide is associated with. It defines the visual appearance of a notes page when printing or exporting slides with speaker notes. A notes page inherits properties from the notes master. The placeholder shape with type BODY on the notes page contains the speaker notes for this slide. The ID of this shape is identified by the speakerNotesObjectId field. The notes page is read-only except for the text content and styles of the speaker notes shape. This property is read-only.
isSkipped	(boolean) Whether the slide is skipped in the presentation mode. Defaults to false.

**Details**

The properties of Page that are only relevant for pages with page\_type SLIDE.

**Value**

SlideProperties object

---

SolidFill

*SolidFill Object*


---

**Description**

SolidFill Object

**Usage**

SolidFill(color = NULL, alpha = NULL)

**Arguments**

- |       |  |
|-------|--|
| color | ( <a href="#">OpaqueColor</a> ) The color value of the solid fill.   |
| alpha | (number) The fraction of this color that should be applied to the pixel. That is, the final pixel color is defined by the equation: pixel color = alpha <ul style="list-style-type: none"> <li>• (color) + (1.0 - alpha)</li> <li>• (background color) This means that a value of 1.0 corresponds to a solid color, whereas a value of 0.0 corresponds to a completely transparent color.</li> </ul> |

**Details**

A solid color fill. The page or page element is filled entirely with the specified color value. If any field is unset, its value may be inherited from a parent placeholder if it exists.

**Value**

SolidFill object

---

SpeakerSpotlight

*SpeakerSpotlight Object*


---

**Description**

SpeakerSpotlight Object

**Usage**

SpeakerSpotlight(speakerSpotlightProperties = NULL)

**Arguments**

- |                            |   |
|----------------------------|---|
| speakerSpotlightProperties | ( <a href="#">SpeakerSpotlightProperties</a> ) The properties of the Speaker Spotlight. |
|----------------------------|---|

**Details**

A PageElement kind representing a Speaker Spotlight.

**Value**

SpeakerSpotlight object

---

SpeakerSpotlightProperties  
*SpeakerSpotlightProperties Object*

---

**Description**

SpeakerSpotlightProperties Object

**Usage**

SpeakerSpotlightProperties(outline = NULL, shadow = NULL)

**Arguments**

outline            ([Outline](#)) The outline of the Speaker Spotlight. If not set, it has no outline.  
shadow            ([Shadow](#)) The shadow of the Speaker Spotlight. If not set, it has no shadow.

**Details**

The properties of the SpeakerSpotlight.

**Value**

SpeakerSpotlightProperties object

---

StretchedPictureFill    *StretchedPictureFill Object*

---

**Description**

StretchedPictureFill Object

**Usage**

StretchedPictureFill(contentUrl = NULL, size = NULL)

**Arguments**

contentUrl	(string) Reading the content_url: An URL to a picture with a default lifetime of 30 minutes. This URL is tagged with the account of the requester. Anyone with the URL effectively accesses the picture as the original requester. Access to the picture may be lost if the presentation's sharing settings change. Writing the content_url: The picture is fetched once at insertion time and a copy is stored for display inside the presentation. Pictures must be less than 50MB in size, cannot exceed 25 megapixels, and must be in one of PNG, JPEG, or GIF format. The provided URL can be at most 2 kB in length.
size	(Size) The original size of the picture fill. This field is read-only.

**Details**

The stretched picture fill. The page or page element is filled entirely with the specified picture. The picture is stretched to fit its container.

**Value**

StretchedPictureFill object

---

SubstringMatchCriteria

*SubstringMatchCriteria Object*

---

**Description**

SubstringMatchCriteria Object

**Usage**

SubstringMatchCriteria(text = NULL, matchCase = NULL)

**Arguments**

text	(string) The text to search for in the shape or table.
matchCase	(boolean) Indicates whether the search should respect case: - True: the search is case sensitive. - False: the search is case insensitive.

**Details**

A criteria that matches a specific string of text in a shape or table.

**Value**

SubstringMatchCriteria object



---

Table	<i>Table Object</i>
-------	---------------------

---

**Description**

Table Object

**Usage**

```
Table(
  rows = NULL,
  columns = NULL,
  tableRows = NULL,
  tableColumns = NULL,
  horizontalBorderRows = NULL,
  verticalBorderRows = NULL
)
```

**Arguments**

**rows** (integer) Number of rows in the table.

**columns** (integer) Number of columns in the table.

**tableRows** (list of [TableRow](#) objects) Properties and contents of each row. Cells that span multiple rows are contained in only one of these rows and have a `row_span` greater than 1.

**tableColumns** (list of [TableColumnProperties](#) objects) Properties of each column.

**horizontalBorderRows** (list of [TableBorderRow](#) objects) Properties of horizontal cell borders. A table's horizontal cell borders are represented as a grid. The grid has one more row than the number of rows in the table and the same number of columns as the table. For example, if the table is 3 x 3, its horizontal borders will be represented as a grid with 4 rows and 3 columns.

**verticalBorderRows** (list of [TableBorderRow](#) objects) Properties of vertical cell borders. A table's vertical cell borders are represented as a grid. The grid has the same number of rows as the table and one more column than the number of columns in the table. For example, if the table is 3 x 3, its vertical borders will be represented as a grid with 3 rows and 4 columns.

**Details**

A PageElement kind representing a table.

**Value**

Table object

---

TableBorderCell	<i>TableBorderCell Object</i>
-----------------	-------------------------------

---

**Description**

TableBorderCell Object

**Usage**

```
TableBorderCell(location = NULL, tableBorderProperties = NULL)
```

**Arguments**

location           ([TableCellLocation](#)) The location of the border within the border table.  
tableBorderProperties           ([TableBorderProperties](#)) The border properties.

**Details**

The properties of each border cell.

**Value**

TableBorderCell object

---

TableBorderFill	<i>TableBorderFill Object</i>
-----------------	-------------------------------

---

**Description**

TableBorderFill Object

**Usage**

```
TableBorderFill(solidFill = NULL)
```

**Arguments**

solidFill           ([SolidFill](#)) Solid fill.

**Details**

The fill of the border.

**Value**

TableBorderFill object

---

TableBorderProperties *TableBorderProperties Object*

---

### Description

TableBorderProperties Object

### Usage

```
TableBorderProperties(tableBorderFill = NULL, weight = NULL, dashStyle = NULL)
```

### Arguments

tableBorderFill

([TableBorderFill](#)) The fill of the table border.

weight

([Dimension](#)) The thickness of the border.

dashStyle

(string) The dash style of the border.

- DASH\_STYLE\_UNSPECIFIED: Unspecified dash style.
- SOLID: Solid line. Corresponds to ECMA-376 ST\_PresetLineDashVal value 'solid'. This is the default dash style.
- DOT: Dotted line. Corresponds to ECMA-376 ST\_PresetLineDashVal value 'dot'.
- DASH: Dashed line. Corresponds to ECMA-376 ST\_PresetLineDashVal value 'dash'.
- DASH\_DOT: Alternating dashes and dots. Corresponds to ECMA-376 ST\_PresetLineDashVal value 'dashDot'.
- LONG\_DASH: Line with large dashes. Corresponds to ECMA-376 ST\_PresetLineDashVal value 'lgDash'.
- LONG\_DASH\_DOT: Alternating large dashes and dots. Corresponds to ECMA-376 ST\_PresetLineDashVal value 'lgDashDot'.

### Details

The border styling properties of the TableBorderCell.

### Value

TableBorderProperties object

---

TableBorderRow	<i>TableBorderRow Object</i>
----------------	------------------------------

---

**Description**

TableBorderRow Object

**Usage**

```
TableBorderRow(tableBorderCells = NULL)
```

**Arguments**

tableBorderCells  
(list of [TableBorderCell](#) objects) Properties of each border cell. When a border's adjacent table cells are merged, it is not included in the response.

**Details**

Contents of each border row in a table.

**Value**

TableBorderRow object

---

TableCell	<i>TableCell Object</i>
-----------	-------------------------

---

**Description**

TableCell Object

**Usage**

```
TableCell(  
  location = NULL,  
  rowSpan = NULL,  
  columnSpan = NULL,  
  text = NULL,  
  tableCellProperties = NULL  
)
```

**Arguments**

location	( <a href="#">TableCellLocation</a> ) The location of the cell within the table.
rowSpan	(integer) Row span of the cell.
columnSpan	(integer) Column span of the cell.
text	( <a href="#">TextContent</a> ) The text content of the cell.
tableCellProperties	( <a href="#">TableCellProperties</a> ) The properties of the table cell.

**Details**

Properties and contents of each table cell.

**Value**

TableCell object

---

TableCellBackgroundFill

*TableCellBackgroundFill Object*

---

**Description**

TableCellBackgroundFill Object

**Usage**

```
TableCellBackgroundFill(propertyState = NULL, solidFill = NULL)
```

**Arguments**

propertyState	(string) The background fill property state. Updating the fill on a table cell will implicitly update this field to RENDERED, unless another value is specified in the same request. To have no fill on a table cell, set this field to NOT_RENDERED. In this case, any other fill fields set in the same request will be ignored. <ul style="list-style-type: none"> <li>RENDERED: If a property's state is RENDERED, then the element has the corresponding property when rendered on a page. If the element is a placeholder shape as determined by the placeholder field, and it inherits from a placeholder shape, the corresponding field may be unset, meaning that the property value is inherited from a parent placeholder. If the element does not inherit, then the field will contain the rendered value. This is the default value.</li> <li>NOT_RENDERED: If a property's state is NOT_RENDERED, then the element does not have the corresponding property when rendered on a page. However, the field may still be set so it can be inherited by child shapes. To remove a property from a rendered element, set its property_state to NOT_RENDERED.</li> </ul>
---------------	---

- INHERIT: If a property's state is INHERIT, then the property state uses the value of corresponding property\_state field on the parent shape. Elements that do not inherit will never have an INHERIT property state.

solidFill      (**SolidFill**) Solid color fill.

### Details

The table cell background fill.

### Value

TableCellBackgroundFill object

---

TableCellLocation	<i>TableCellLocation Object</i>
-------------------	---------------------------------

---

### Description

TableCellLocation Object

### Usage

TableCellLocation(rowIndex = NULL, columnIndex = NULL)

### Arguments

rowIndex      (integer) The 0-based row index.  
 columnIndex    (integer) The 0-based column index.

### Details

A location of a single table cell within a table.

### Value

TableCellLocation object

---

TableCellProperties    *TableCellProperties Object*

---

## Description

TableCellProperties Object

## Usage

```
TableCellProperties(tableCellBackgroundFill = NULL, contentAlignment = NULL)
```

## Arguments

tableCellBackgroundFill

([TableCellBackgroundFill](#)) The background fill of the table cell. The default fill matches the fill for newly created table cells in the Slides editor.

contentAlignment

(string) The alignment of the content in the table cell. The default alignment matches the alignment for newly created table cells in the Slides editor.

- CONTENT\_ALIGNMENT\_UNSPECIFIED: An unspecified content alignment. The content alignment is inherited from the parent if it exists.
- CONTENT\_ALIGNMENT\_UNSUPPORTED: An unsupported content alignment.
- TOP: An alignment that aligns the content to the top of the content holder. Corresponds to ECMA-376 ST\_TextAnchoringType 't'.
- MIDDLE: An alignment that aligns the content to the middle of the content holder. Corresponds to ECMA-376 ST\_TextAnchoringType 'ctr'.
- BOTTOM: An alignment that aligns the content to the bottom of the content holder. Corresponds to ECMA-376 ST\_TextAnchoringType 'b'.

## Details

The properties of the TableCell.

## Value

TableCellProperties object

---

TableColumnProperties *TableColumnProperties Object*

---

### Description

TableColumnProperties Object

### Usage

TableColumnProperties(columnWidth = NULL)

### Arguments

columnWidth      ([Dimension](#)) Width of a column.

### Details

Properties of each column in a table.

### Value

TableColumnProperties object

---

TableRange                      *TableRange Object*

---

### Description

TableRange Object

### Usage

TableRange(location = NULL, rowSpan = NULL, columnSpan = NULL)

### Arguments

location              ([TableCellLocation](#)) The starting location of the table range.

rowSpan              (integer) The row span of the table range.

columnSpan          (integer) The column span of the table range.

### Details

A table range represents a reference to a subset of a table. It's important to note that the cells specified by a table range do not necessarily form a rectangle. For example, let's say we have a 3 x 3 table where all the cells of the last row are merged together. The table looks like this: [ ] A table range with location = (0, 0), row span = 3 and column span = 2 specifies the following cells: x x [ x x x ]



**Value**

TableRange object

TableRow                      *TableRow Object*

**Description**

TableRow Object

**Usage**

TableRow(rowHeight = NULL, tableRowProperties = NULL, tableCells = NULL)

**Arguments**

- rowHeight            ([Dimension](#)) Height of a row.
- tableRowProperties            ([TableRowProperties](#)) Properties of the row.
- tableCells            (list of [TableCell](#) objects) Properties and contents of each cell. Cells that span multiple columns are represented only once with a column\_span greater than 1. As a result, the length of this collection does not always match the number of columns of the entire table.

**Details**

Properties and contents of each row in a table.

**Value**

TableRow object

TableRowProperties            *TableRowProperties Object*

**Description**

TableRowProperties Object

**Usage**

TableRowProperties(minRowHeight = NULL)

**Arguments**

`minRowHeight` ([Dimension](#)) Minimum height of the row. The row will be rendered in the Slides editor at a height equal to or greater than this value in order to show all the text in the row's cell(s).

**Details**

Properties of each row in a table.

**Value**

TableRowProperties object

---

TextContent	<i>TextContent Object</i>
-------------	---------------------------

---

**Description**

TextContent Object

**Usage**

```
TextContent(textElements = NULL, lists = NULL)
```

**Arguments**

`textElements` (list of [TextElement](#) objects) The text contents broken down into its component parts, including styling information. This property is read-only.

`lists` (named list of [List](#) objects) The bulleted lists contained in this text, keyed by list ID.

**Details**

The general text content. The text must reside in a compatible shape (e.g. text box or rectangle) or a table cell in a page.

**Value**

TextContent object

---

TextElement

*TextElement Object*


---

**Description**

TextElement Object

**Usage**

```
TextElement(
    startIndex = NULL,
    endIndex = NULL,
    paragraphMarker = NULL,
    textRun = NULL,
    autoText = NULL
)
```

**Arguments**

startIndex	(integer) The zero-based start index of this text element, in Unicode code units.
endIndex	(integer) The zero-based end index of this text element, exclusive, in Unicode code units.
paragraphMarker	<a href="#">(ParagraphMarker)</a> A marker representing the beginning of a new paragraph. The <code>start_index</code> and <code>end_index</code> of this <code>TextElement</code> represent the range of the paragraph. Other <code>TextElements</code> with an index range contained inside this paragraph's range are considered to be part of this paragraph. The range of indices of two separate paragraphs will never overlap.
textRun	<a href="#">(TextRun)</a> A <code>TextElement</code> representing a run of text where all of the characters in the run have the same <code>TextStyle</code> . The <code>start_index</code> and <code>end_index</code> of <code>TextRuns</code> will always be fully contained in the index range of a single <code>paragraph_marker</code> <code>TextElement</code> . In other words, a <code>TextRun</code> will never span multiple paragraphs.
autoText	<a href="#">(AutoText)</a> A <code>TextElement</code> representing a spot in the text that is dynamically replaced with content that can change over time.

**Details**

A `TextElement` describes the content of a range of indices in the text content of a `Shape` or `TableCell`.

**Value**

TextElement object

---

TextRun	<i>TextRun Object</i>
---------	-----------------------

---

**Description**

TextRun Object

**Usage**

```
TextRun(content = NULL, style = NULL)
```

**Arguments**

content	(string) The text of this run.
style	( <a href="#">TextStyle</a> ) The styling applied to this run.

**Details**

A TextElement kind that represents a run of text that all has the same styling.

**Value**

TextRun object

---

TextStyle	<i>TextStyle Object</i>
-----------	-------------------------

---

**Description**

TextStyle Object

**Usage**

```
TextStyle(  
  backgroundColor = NULL,  
  foregroundColor = NULL,  
  bold = NULL,  
  italic = NULL,  
  fontFamily = NULL,  
  fontSize = NULL,  
  link = NULL,  
  baselineOffset = NULL,  
  smallCaps = NULL,  
  strikethrough = NULL,  
  underline = NULL,  
  weightedFontFamily = NULL  
)
```

**Arguments**

backgroundColor	( <a href="#">OptionalColor</a> ) The background color of the text. If set, the color is either opaque or transparent, depending on if the opaque_color field in it is set.
foregroundColor	( <a href="#">OptionalColor</a> ) The color of the text itself. If set, the color is either opaque or transparent, depending on if the opaque_color field in it is set.
bold	(boolean) Whether or not the text is rendered as bold.
italic	(boolean) Whether or not the text is italicized.
fontFamily	(string) The font family of the text. The font family can be any font from the Font menu in Slides or from <a href="#">Google Fonts</a> . If the font name is unrecognized, the text is rendered in Arial. Some fonts can affect the weight of the text. If an update request specifies values for both font_family and bold, the explicitly-set bold value is used.
fontSize	( <a href="#">Dimension</a> ) The size of the text's font. When read, the font_size will specified in points.
link	( <a href="#">Link</a> ) The hyperlink destination of the text. If unset, there is no link. Links are not inherited from parent text. Changing the link in an update request causes some other changes to the text style of the range: <ul style="list-style-type: none"> <li>• When setting a link, the text foreground color will be set to ThemeColorType.HYPERLINK and the text will be underlined. If these fields are modified in the same request, those values will be used instead of the link defaults.</li> <li>• Setting a link on a text range that overlaps with an existing link will also update the existing link to point to the new URL.</li> <li>• Links are not settable on newline characters. As a result, setting a link on a text range that crosses a paragraph boundary, such as "ABC\n123", will separate the newline character(s) into their own text runs. The link will be applied separately to the runs before and after the newline.</li> <li>• Removing a link will update the text style of the range to match the style of the preceding text (or the default text styles if the preceding text is another link) unless different styles are being set in the same request.</li> </ul>
baselineOffset	(string) The text's vertical offset from its normal position. Text with SUPERScript or SUBSCRIPT baseline offsets is automatically rendered in a smaller font size, computed based on the font_size field. The font_size itself is not affected by changes in this field. <ul style="list-style-type: none"> <li>• BASELINE_OFFSET_UNSPECIFIED: The text's baseline offset is inherited from the parent.</li> <li>• NONE: The text is not vertically offset.</li> <li>• SUPERScript: The text is vertically offset upwards (superscript).</li> <li>• SUBSCRIPT: The text is vertically offset downwards (subscript).</li> </ul>
smallCaps	(boolean) Whether or not the text is in small capital letters.
strikethrough	(boolean) Whether or not the text is struck through.
underline	(boolean) Whether or not the text is underlined.

**weightedFontFamily**

(**WeightedFontFamily**) The font family and rendered weight of the text. This field is an extension of `font_family` meant to support explicit font weights without breaking backwards compatibility. As such, when reading the style of a range of text, the value of `weighted_font_family#font_family` will always be equal to that of `font_family`. However, when writing, if both fields are included in the field mask (either explicitly or through the wildcard "\*"), their values are reconciled as follows:

- If `font_family` is set and `weighted_font_family` is not, the value of `font_family` is applied with weight 400 ("normal").
- If both fields are set, the value of `font_family` must match that of `weighted_font_family#font_family`. If so, the font family and weight of `weighted_font_family` is applied. Otherwise, a 400 bad request error is returned.
- If `weighted_font_family` is set and `font_family` is not, the font family and weight of `weighted_font_family` is applied.
- If neither field is set, the font family and weight of the text inherit from the parent. Note that these properties cannot inherit separately from each other. If an update request specifies values for both `weighted_font_family` and `bold`, the `weighted_font_family` is applied first, then `bold`. If `weighted_font_family#weight` is not set, it defaults to 400. If `weighted_font_family` is set, then `weighted_font_family#font_family` must also be set with a non-empty value. Otherwise, a 400 bad request error is returned.

**Details**

Represents the styling that can be applied to a `TextRun`. If this text is contained in a shape with a parent placeholder, then these text styles may be inherited from the parent. Which text styles are inherited depend on the nesting level of lists:

- A text run in a paragraph that is not in a list will inherit its text style from the the newline character in the paragraph at the 0 nesting level of the list inside the parent placeholder.
- A text run in a paragraph that is in a list will inherit its text style from the newline character in the paragraph at its corresponding nesting level of the list inside the parent placeholder. Inherited text styles are represented as unset fields in this message. If text is contained in a shape without a parent placeholder, unsetting these fields will revert the style to a value matching the defaults in the Slides editor.

**Value**

TextStyle object

---

ThemeColorPair	<i>ThemeColorPair Object</i>
----------------	------------------------------

---

**Description**

ThemeColorPair Object

**Usage**

```
ThemeColorPair(type = NULL, color = NULL)
```

**Arguments**

type	<p>(string) The type of the theme color.</p> <ul style="list-style-type: none"> <li>• THEME_COLOR_TYPE_UNSPECIFIED: Unspecified theme color. This value should not be used.</li> <li>• DARK1: Represents the first dark color.</li> <li>• LIGHT1: Represents the first light color.</li> <li>• DARK2: Represents the second dark color.</li> <li>• LIGHT2: Represents the second light color.</li> <li>• ACCENT1: Represents the first accent color.</li> <li>• ACCENT2: Represents the second accent color.</li> <li>• ACCENT3: Represents the third accent color.</li> <li>• ACCENT4: Represents the fourth accent color.</li> <li>• ACCENT5: Represents the fifth accent color.</li> <li>• ACCENT6: Represents the sixth accent color.</li> <li>• HYPERLINK: Represents the color to use for hyperlinks.</li> <li>• FOLLOWED_HYPERLINK: Represents the color to use for visited hyperlinks.</li> <li>• TEXT1: Represents the first text color.</li> <li>• BACKGROUND1: Represents the first background color.</li> <li>• TEXT2: Represents the second text color.</li> <li>• BACKGROUND2: Represents the second background color.</li> </ul>
color	<p>(<a href="#">RgbColor</a>) The concrete color corresponding to the theme color type above.</p>

**Details**

A pair mapping a theme color type to the concrete color it represents.

**Value**

ThemeColorPair object

---

Thumbnail

*Thumbnail Object*


---

**Description**

Thumbnail Object

**Usage**

```
Thumbnail(width = NULL, height = NULL, contentUrl = NULL)
```

**Arguments**

width	(integer) The positive width in pixels of the thumbnail image.
height	(integer) The positive height in pixels of the thumbnail image.
contentUrl	(string) The content URL of the thumbnail image. The URL to the image has a default lifetime of 30 minutes. This URL is tagged with the account of the requester. Anyone with the URL effectively accesses the image as the original requester. Access to the image may be lost if the presentation's sharing settings change. The mime type of the thumbnail image is the same as specified in the GetPageThumbnailRequest.

**Details**

The thumbnail of a page.

**Value**

Thumbnail object

---

UngroupObjectsRequest *UngroupObjectsRequest Object*

---

**Description**

UngroupObjectsRequest Object

**Usage**

UngroupObjectsRequest(objectIds = NULL)

**Arguments**

objectIds	(list of <i>string</i> objects) The object IDs of the objects to ungroup. Only groups that are not inside other groups can be ungrouped. All the groups should be on the same page. The group itself is deleted. The visual sizes and positions of all the children are preserved.
-----------	--

**Details**

Ungroups objects, such as groups.

**Value**

UngroupObjectsRequest object



---

UnmergeTableCellsRequest

*UnmergeTableCellsRequest Object*

---

### Description

UnmergeTableCellsRequest Object

### Usage

```
UnmergeTableCellsRequest(objectId = NULL, tableRange = NULL)
```

### Arguments

objectId	(string) The object ID of the table.
tableRange	( <a href="#">TableRange</a> ) The table range specifying which cells of the table to unmerge. All merged cells in this range will be unmerged, and cells that are already unmerged will not be affected. If the range has no merged cells, the request will do nothing. If there is text in any of the merged cells, the text will remain in the upper-left ("head") cell of the resulting block of unmerged cells.

### Details

Unmerges cells in a Table.

### Value

UnmergeTableCellsRequest object

---

UpdateImagePropertiesRequest

*UpdateImagePropertiesRequest Object*

---

### Description

UpdateImagePropertiesRequest Object

### Usage

```
UpdateImagePropertiesRequest(  
  objectId = NULL,  
  imageProperties = NULL,  
  fields = NULL  
)
```

**Arguments**

objectId	(string) The object ID of the image the updates are applied to.
imageProperties	( <a href="#">ImageProperties</a> ) The image properties to update.
fields	(string) The fields that should be updated. At least one field must be specified. The root imageProperties is implied and should not be specified. A single "*" can be used as short-hand for listing every field. For example to update the image outline color, set fields to "outline.outlineFill.solidFill.color". To reset a property to its default value, include its field name in the field mask but leave the field itself unset.

**Details**

Update the properties of an Image.

**Value**

UpdateImagePropertiesRequest object

---

UpdateLineCategoryRequest

*UpdateLineCategoryRequest Object*

---

**Description**

UpdateLineCategoryRequest Object

**Usage**

UpdateLineCategoryRequest(objectId = NULL, lineCategory = NULL)

**Arguments**

objectId	(string) The object ID of the line the update is applied to. Only a line with a category indicating it is a "connector" can be updated. The line may be rerouted after updating its category.
lineCategory	(string) The line category to update to. The exact line type is determined based on the category to update to and how it's routed to connect to other page elements. <ul style="list-style-type: none"> <li>• LINE_CATEGORY_UNSPECIFIED: Unspecified line category.</li> <li>• STRAIGHT: Straight connectors, including straight connector 1.</li> <li>• BENT: Bent connectors, including bent connector 2 to 5.</li> <li>• CURVED: Curved connectors, including curved connector 2 to 5.</li> </ul>

**Details**

Updates the category of a line.

**Value**

UpdateLineCategoryRequest object

---

UpdateLinePropertiesRequest

*UpdateLinePropertiesRequest Object*

---

**Description**

UpdateLinePropertiesRequest Object

**Usage**

```
UpdateLinePropertiesRequest(  
  objectId = NULL,  
  lineProperties = NULL,  
  fields = NULL  
)
```

**Arguments**

**objectId** (string) The object ID of the line the update is applied to.

**lineProperties** ([LineProperties](#)) The line properties to update.

**fields** (string) The fields that should be updated. At least one field must be specified. The root `lineProperties` is implied and should not be specified. A single "\*" can be used as short-hand for listing every field. For example to update the line solid fill color, set `fields` to "lineFill.solidFill.color". To reset a property to its default value, include its field name in the field mask but leave the field itself unset.

**Details**

Updates the properties of a Line.

**Value**

UpdateLinePropertiesRequest object

---

 UpdatePageElementAltTextRequest

*UpdatePageElementAltTextRequest Object*


---

**Description**

UpdatePageElementAltTextRequest Object

**Usage**

```
UpdatePageElementAltTextRequest(
  objectId = NULL,
  title = NULL,
  description = NULL
)
```

**Arguments**

objectId	(string) The object ID of the page element the updates are applied to.
title	(string) The updated alt text title of the page element. If unset the existing value will be maintained. The title is exposed to screen readers and other accessibility interfaces. Only use human readable values related to the content of the page element.
description	(string) The updated alt text description of the page element. If unset the existing value will be maintained. The description is exposed to screen readers and other accessibility interfaces. Only use human readable values related to the content of the page element.

**Details**

Updates the alt text title and/or description of a page element.

**Value**

UpdatePageElementAltTextRequest object

---

 UpdatePageElementsZOrderRequest

*UpdatePageElementsZOrderRequest Object*


---

**Description**

UpdatePageElementsZOrderRequest Object

**Usage**

```
UpdatePageElementsZOrderRequest(pageElementObjectIds = NULL, operation = NULL)
```

**Arguments**

pageElementObjectIds

(list of [string](#) objects) The object IDs of the page elements to update. All the page elements must be on the same page and must not be grouped.

operation

(string) The Z-order operation to apply on the page elements. When applying the operation on multiple page elements, the relative Z-orders within these page elements before the operation is maintained.

- Z\_ORDER\_OPERATION\_UNSPECIFIED: Unspecified operation.
- BRING\_TO\_FRONT: Brings the page elements to the front of the page.
- BRING\_FORWARD: Brings the page elements forward on the page by one element relative to the forwardmost one in the specified page elements.
- SEND\_BACKWARD: Sends the page elements backward on the page by one element relative to the furthest behind one in the specified page elements.
- SEND\_TO\_BACK: Sends the page elements to the back of the page.

**Details**

Updates the Z-order of page elements. Z-order is an ordering of the elements on the page from back to front. The page element in the front may cover the elements that are behind it.

**Value**

UpdatePageElementsZOrderRequest object

---

UpdatePageElementTransformRequest

*UpdatePageElementTransformRequest Object*

---

**Description**

UpdatePageElementTransformRequest Object

**Usage**

```
UpdatePageElementTransformRequest(
  objectId = NULL,
  transform = NULL,
  applyMode = NULL
)
```

**Arguments**

objectId	(string) The object ID of the page element to update.
transform	( <a href="#">AffineTransform</a> ) The input transform matrix used to update the page element.
applyMode	(string) The apply mode of the transform update. <ul style="list-style-type: none"> <li>• APPLY_MODE_UNSPECIFIED: Unspecified mode.</li> <li>• RELATIVE: Applies the new AffineTransform matrix to the existing one, and replaces the existing one with the resulting concatenation.</li> <li>• ABSOLUTE: Replaces the existing AffineTransform matrix with the new one.</li> </ul>

**Details**

Updates the transform of a page element. Updating the transform of a group will change the absolute transform of the page elements in that group, which can change their visual appearance. See the documentation for `PageElement.transform` for more details.

**Value**

UpdatePageElementTransformRequest object

---

UpdatePagePropertiesRequest

*UpdatePagePropertiesRequest Object*

---

**Description**

UpdatePagePropertiesRequest Object

**Usage**

```
UpdatePagePropertiesRequest(
  objectId = NULL,
  pageProperties = NULL,
  fields = NULL
)
```

**Arguments**

objectId	(string) The object ID of the page the update is applied to.
pageProperties	( <a href="#">PageProperties</a> ) The page properties to update.
fields	(string) The fields that should be updated. At least one field must be specified. The root <code>pageProperties</code> is implied and should not be specified. A single "*" can be used as short-hand for listing every field. For example to update the page background solid fill color, set <code>fields</code> to <code>"pageBackgroundFill.solidFill.color"</code> . To reset a property to its default value, include its field name in the field mask but leave the field itself unset.

**Details**

Updates the properties of a Page.

**Value**

UpdatePagePropertiesRequest object

---

UpdateParagraphStyleRequest

*UpdateParagraphStyleRequest Object*

---

**Description**

UpdateParagraphStyleRequest Object

**Usage**

```
UpdateParagraphStyleRequest(
    objectId = NULL,
    cellLocation = NULL,
    style = NULL,
    textRange = NULL,
    fields = NULL
)
```

**Arguments**

objectId	(string) The object ID of the shape or table with the text to be styled.
cellLocation	( <a href="#">TableCellLocation</a> ) The location of the cell in the table containing the paragraph(s) to style. If object_id refers to a table, cell_location must have a value. Otherwise, it must not.
style	( <a href="#">ParagraphStyle</a> ) The paragraph's style.
textRange	( <a href="#">Range</a> ) The range of text containing the paragraph(s) to style.
fields	(string) The fields that should be updated. At least one field must be specified. The root style is implied and should not be specified. A single "*" can be used as short-hand for listing every field. For example, to update the paragraph alignment, set fields to "alignment". To reset a property to its default value, include its field name in the field mask but leave the field itself unset.

**Details**

Updates the styling for all of the paragraphs within a Shape or Table that overlap with the given text index range.

**Value**

UpdateParagraphStyleRequest object

---

 UpdateShapePropertiesRequest

*UpdateShapePropertiesRequest Object*


---

**Description**

UpdateShapePropertiesRequest Object

**Usage**

```
UpdateShapePropertiesRequest(
  objectId = NULL,
  shapeProperties = NULL,
  fields = NULL
)
```

**Arguments**

objectId	(string) The object ID of the shape the updates are applied to.
shapeProperties	( <a href="#">ShapeProperties</a> ) The shape properties to update.
fields	(string) The fields that should be updated. At least one field must be specified. The root shapeProperties is implied and should not be specified. A single "*" can be used as short-hand for listing every field. For example to update the shape background solid fill color, set fields to "shapeBackgroundFill.solidFill.color". To reset a property to its default value, include its field name in the field mask but leave the field itself unset.

**Details**

Update the properties of a Shape.

**Value**

UpdateShapePropertiesRequest object

---

 UpdateSlidePropertiesRequest

*UpdateSlidePropertiesRequest Object*


---

**Description**

UpdateSlidePropertiesRequest Object



**Usage**

```
UpdateSlidePropertiesRequest(
  objectId = NULL,
  slideProperties = NULL,
  fields = NULL
)
```

**Arguments**

**objectId** (string) The object ID of the slide the update is applied to.

**slideProperties** ([SlideProperties](#)) The slide properties to update.

**fields** (string) The fields that should be updated. At least one field must be specified. The root 'slideProperties' is implied and should not be specified. A single "\*" can be used as short-hand for listing every field. For example to update whether a slide is skipped, set fields to "isSkipped". To reset a property to its default value, include its field name in the field mask but leave the field itself unset.

**Details**

Updates the properties of a Slide.

**Value**

UpdateSlidePropertiesRequest object

---

UpdateSlidesPositionRequest

*UpdateSlidesPositionRequest Object*

---

**Description**

UpdateSlidesPositionRequest Object

**Usage**

```
UpdateSlidesPositionRequest(slideObjectIds = NULL, insertionIndex = NULL)
```

**Arguments**

**slideObjectIds** (list of [string](#) objects) The IDs of the slides in the presentation that should be moved. The slides in this list must be in existing presentation order, without duplicates.

**insertionIndex** (integer) The index where the slides should be inserted, based on the slide arrangement before the move takes place. Must be between zero and the number of slides in the presentation, inclusive.

**Details**

Updates the position of slides in the presentation.

**Value**

UpdateSlidesPositionRequest object

---

UpdateTableBorderPropertiesRequest

*UpdateTableBorderPropertiesRequest Object*

---

**Description**

UpdateTableBorderPropertiesRequest Object

**Usage**

```
UpdateTableBorderPropertiesRequest(
    objectId = NULL,
    tableRange = NULL,
    borderPosition = NULL,
    tableBorderProperties = NULL,
    fields = NULL
)
```

**Arguments**

- |                       |  |
|-----------------------|--|
| objectId              | (string) The object ID of the table.   |
| tableRange            | ( <a href="#">TableRange</a> ) The table range representing the subset of the table to which the updates are applied. If a table range is not specified, the updates will apply to the entire table.   |
| borderPosition        | (string) The border position in the table range the updates should apply to. If a border position is not specified, the updates will apply to all borders in the table range. <ul style="list-style-type: none"> <li>• ALL: All borders in the range.</li> <li>• BOTTOM: Borders at the bottom of the range.</li> <li>• INNER: Borders on the inside of the range.</li> <li>• INNER_HORIZONTAL: Horizontal borders on the inside of the range.</li> <li>• INNER_VERTICAL: Vertical borders on the inside of the range.</li> <li>• LEFT: Borders at the left of the range.</li> <li>• OUTER: Borders along the outside of the range.</li> <li>• RIGHT: Borders at the right of the range.</li> <li>• TOP: Borders at the top of the range.</li> </ul> |
| tableBorderProperties | ( <a href="#">TableBorderProperties</a> ) The table border properties to update.   |

**fields** (string) The fields that should be updated. At least one field must be specified. The root `tableBorderProperties` is implied and should not be specified. A single "\*" can be used as short-hand for listing every field. For example to update the table border solid fill color, set `fields` to `"tableBorderFill.solidFill.color"`. To reset a property to its default value, include its field name in the field mask but leave the field itself unset.

### Details

Updates the properties of the table borders in a Table.

### Value

UpdateTableBorderPropertiesRequest object

---

UpdateTableCellPropertiesRequest

*UpdateTableCellPropertiesRequest Object*

---

### Description

UpdateTableCellPropertiesRequest Object

### Usage

```
UpdateTableCellPropertiesRequest(
    objectId = NULL,
    tableRange = NULL,
    tableCellProperties = NULL,
    fields = NULL
)
```

### Arguments

**objectId** (string) The object ID of the table.

**tableRange** ([TableRange](#)) The table range representing the subset of the table to which the updates are applied. If a table range is not specified, the updates will apply to the entire table.

**tableCellProperties** ([TableCellProperties](#)) The table cell properties to update.

**fields** (string) The fields that should be updated. At least one field must be specified. The root `tableCellProperties` is implied and should not be specified. A single "\*" can be used as short-hand for listing every field. For example to update the table cell background solid fill color, set `fields` to `"tableCellBackgroundFill.solidFill.color"`. To reset a property to its default value, include its field name in the field mask but leave the field itself unset.

**Details**

Update the properties of a TableCell.

**Value**

UpdateTableCellPropertiesRequest object

---

UpdateTableColumnPropertiesRequest

*UpdateTableColumnPropertiesRequest Object*

---

**Description**

UpdateTableColumnPropertiesRequest Object

**Usage**

```
UpdateTableColumnPropertiesRequest(
  objectId = NULL,
  columnIndices = NULL,
  tableColumnProperties = NULL,
  fields = NULL
)
```

**Arguments**

objectId	(string) The object ID of the table.
columnIndices	(list of <a href="#">integer</a> objects) The list of zero-based indices specifying which columns to update. If no indices are provided, all columns in the table will be updated.
tableColumnProperties	( <a href="#">TableColumnProperties</a> ) The table column properties to update. If the value of table_column_properties#column_width in the request is less than 406,400 EMU (32 points), a 400 bad request error is returned.
fields	(string) The fields that should be updated. At least one field must be specified. The root tableColumnProperties is implied and should not be specified. A single "*" can be used as short-hand for listing every field. For example to update the column width, set fields to "column_width". If "column_width" is included in the field mask but the property is left unset, the column width will default to 406,400 EMU (32 points).

**Details**

Updates the properties of a Table column.

**Value**

UpdateTableColumnPropertiesRequest object

---

UpdateTableRowPropertiesRequest

*UpdateTableRowPropertiesRequest Object*

---

## Description

UpdateTableRowPropertiesRequest Object

## Usage

```
UpdateTableRowPropertiesRequest(  
  objectId = NULL,  
  rowIndices = NULL,  
  tableRowProperties = NULL,  
  fields = NULL  
)
```

## Arguments

objectId	(string) The object ID of the table.
rowIndices	(list of <a href="#">integer</a> objects) The list of zero-based indices specifying which rows to update. If no indices are provided, all rows in the table will be updated.
tableRowProperties	( <a href="#">TableRowProperties</a> ) The table row properties to update.
fields	(string) The fields that should be updated. At least one field must be specified. The root <code>tableRowProperties</code> is implied and should not be specified. A single "*" can be used as short-hand for listing every field. For example to update the minimum row height, set fields to "min_row_height". If "min_row_height" is included in the field mask but the property is left unset, the minimum row height will default to 0.

## Details

Updates the properties of a Table row.

## Value

UpdateTableRowPropertiesRequest object

---

 UpdateTextStyleRequest

*UpdateTextStyleRequest Object*


---

## Description

UpdateTextStyleRequest Object

## Usage

```
UpdateTextStyleRequest(
  objectId = NULL,
  cellLocation = NULL,
  style = NULL,
  textRange = NULL,
  fields = NULL
)
```

## Arguments

objectId	(string) The object ID of the shape or table with the text to be styled.
cellLocation	( <a href="#">TableCellLocation</a> ) The location of the cell in the table containing the text to style. If object_id refers to a table, cell_location must have a value. Otherwise, it must not.
style	( <a href="#">TextStyle</a> ) The style(s) to set on the text. If the value for a particular style matches that of the parent, that style will be set to inherit. Certain text style changes may cause other changes meant to mirror the behavior of the Slides editor. See the documentation of <a href="#">TextStyle</a> for more information.
textRange	( <a href="#">Range</a> ) The range of text to style. The range may be extended to include adjacent newlines. If the range fully contains a paragraph belonging to a list, the paragraph's bullet is also updated with the matching text style.
fields	(string) The fields that should be updated. At least one field must be specified. The root style is implied and should not be specified. A single "*" can be used as short-hand for listing every field. For example, to update the text style to bold, set fields to "bold". To reset a property to its default value, include its field name in the field mask but leave the field itself unset.

## Details

Update the styling of text in a Shape or Table.

## Value

UpdateTextStyleRequest object

---

UpdateVideoPropertiesRequest

*UpdateVideoPropertiesRequest Object*

---

### Description

UpdateVideoPropertiesRequest Object

### Usage

```
UpdateVideoPropertiesRequest(  
    objectId = NULL,  
    videoProperties = NULL,  
    fields = NULL  
)
```

### Arguments

objectId	(string) The object ID of the video the updates are applied to.
videoProperties	( <a href="#">VideoProperties</a> ) The video properties to update.
fields	(string) The fields that should be updated. At least one field must be specified. The root videoProperties is implied and should not be specified. A single "*" can be used as short-hand for listing every field. For example to update the video outline color, set fields to "outline.outlineFill.solidFill.color". To reset a property to its default value, include its field name in the field mask but leave the field itself unset.

### Details

Update the properties of a Video.

### Value

UpdateVideoPropertiesRequest object

---

Video

*Video Object*

---

### Description

Video Object

### Usage

```
Video(url = NULL, source = NULL, id = NULL, videoProperties = NULL)
```

**Arguments**

url	(string) An URL to a video. The URL is valid as long as the source video exists and sharing settings do not change.
source	(string) The video source. <ul style="list-style-type: none"> <li>SOURCE_UNSPECIFIED: The video source is unspecified.</li> <li>YOUTUBE: The video source is YouTube.</li> <li>DRIVE: The video source is Google Drive.</li> </ul>
id	(string) The video source's unique identifier for this video.
videoProperties	( <a href="#">VideoProperties</a> ) The properties of the video.

**Details**

A PageElement kind representing a video.

**Value**

Video object

---

VideoProperties	<i>VideoProperties Object</i>
-----------------	-------------------------------

---

**Description**

VideoProperties Object

**Usage**

```
VideoProperties(
  outline = NULL,
  autoPlay = NULL,
  start = NULL,
  end = NULL,
  mute = NULL
)
```

**Arguments**

outline	( <a href="#">Outline</a> ) The outline of the video. The default outline matches the defaults for new videos created in the Slides editor.
autoPlay	(boolean) Whether to enable video autoplay when the page is displayed in present mode. Defaults to false.



start	(integer) The time at which to start playback, measured in seconds from the beginning of the video. If set, the start time should be before the end time. If you set this to a value that exceeds the video's length in seconds, the video will be played from the last second. If not set, the video will be played from the beginning.
end	(integer) The time at which to end playback, measured in seconds from the beginning of the video. If set, the end time should be after the start time. If not set or if you set this to a value that exceeds the video's length, the video will be played until its end.
mute	(boolean) Whether to mute the audio during video playback. Defaults to false.

**Details**

The properties of the Video.

**Value**

VideoProperties object

---

WeightedFontFamily      *WeightedFontFamily Object*

---

**Description**

WeightedFontFamily Object

**Usage**

```
WeightedFontFamily(fontFamily = NULL, weight = NULL)
```

**Arguments**

fontFamily	(string) The font family of the text. The font family can be any font from the Font menu in Slides or from <b>Google Fonts</b> . If the font name is unrecognized, the text is rendered in Arial.
weight	(integer) The rendered weight of the text. This field can have any value that is a multiple of 100 between 100 and 900, inclusive. This range corresponds to the numerical values described in the CSS 2.1 Specification, <b>section 15.6</b> , with non-numerical values disallowed. Weights greater than or equal to 700 are considered bold, and weights less than 700 are not bold. The default value is 400 ("normal").

**Details**

Represents a font family and weight used to style a TextRun.

**Value**

WeightedFontFamily object

---

WordArt	<i>WordArt Object</i>
---------	-----------------------

---

**Description**

WordArt Object

**Usage**

```
WordArt(renderedText = NULL)
```

**Arguments**

renderedText (string) The text rendered as word art.

**Details**

A PageElement kind representing word art.

**Value**

WordArt object

---

WriteControl	<i>WriteControl Object</i>
--------------	----------------------------

---

**Description**

WriteControl Object

**Usage**

```
WriteControl(requiredRevisionId = NULL)
```

**Arguments**

requiredRevisionId  
 (string) The revision ID of the presentation required for the write request. If specified and the required revision ID doesn't match the presentation's current revision ID, the request is not processed and returns a 400 bad request error. When a required revision ID is returned in a response, it indicates the revision ID of the document after the request was applied.

**Details**

Provides control over how write requests are executed.

**Value**

WriteControl object

# Index

- \* **BatchUpdatePresentationRequest functions**
  - presentations.batchUpdate, 72
- \* **Presentation functions**
  - presentations.create, 73
- \* **auth functions**
  - ladder\_auth, 42
  - ladder\_auth\_configure, 44
  - ladder\_deauth, 46
- \* **low-level API functions**
  - ladder\_has\_token, 47
  - ladder\_token, 47
- AffineTransform, 5, 66, 67, 89, 126
- Autofit, 6, 99
- AutoText, 7, 115
- BatchUpdatePresentationRequest, 7, 73
- BatchUpdatePresentationResponse, 8
- Bullet, 9, 68
- choose\_slides, 9
- ColorScheme, 10, 67
- ColorStop, 10, 76
- CreateImageRequest, 11, 84
- CreateImageResponse, 12, 87
- CreateLineRequest, 12, 85
- CreateLineResponse, 13, 87
- CreateParagraphBulletsRequest, 14, 85
- CreateShapeRequest, 16, 84
- CreateShapeResponse, 22, 87
- CreateSheetsChartRequest, 23, 85
- CreateSheetsChartResponse, 24, 87
- CreateSlideRequest, 24, 84
- CreateSlideResponse, 25, 87
- CreateTableRequest, 26, 84
- CreateTableResponse, 27, 87
- CreateVideoRequest, 27, 84
- CreateVideoResponse, 28, 87
- CropProperties, 29, 39
- DeleteObjectRequest, 30, 84
- DeleteParagraphBulletsRequest, 31, 85
- DeleteTableColumnRequest, 31, 84
- DeleteTableRowRequest, 32, 84
- DeleteTextRequest, 33, 84
- Dimension, 33, 54, 61, 69, 89, 100, 107, 112–114, 117
- DuplicateObjectRequest, 34, 85
- DuplicateObjectResponse, 35, 87
- gargle::AuthState, 45
- gargle::gargle\_oauth\_client\_from\_json(), 45
- gargle::gargle\_options, 44
- gargle::token\_email(), 48
- gargle::token\_fetch(), 42, 44
- gargle::token\_tokeninfo(), 48
- gargle::token\_userinfo(), 48
- gargle\_oauth\_cache(), 43
- gargle\_oauth\_client\_type(), 43
- gargle\_oauth\_email(), 43
- gargle\_oob\_default(), 43
- Group, 35, 66
- GroupObjectsRequest, 36, 86
- GroupObjectsResponse, 37, 87
- htr, 47
- htr::config(), 43, 47
- Image, 37, 66
- ImageProperties, 38, 38, 100, 122
- InsertTableColumnsRequest, 39, 84
- InsertTableRowsRequest, 40, 84
- InsertTextRequest, 41, 84
- integer, 132, 133
- jsonlite::fromJSON(), 43, 45
- ladder\_api\_key (ladder\_auth\_configure), 44
- ladder\_api\_key(), 46

- ladder\_auth, [42](#), [45](#), [46](#)
- ladder\_auth(), [45](#), [47](#)
- ladder\_auth\_configure, [44](#), [44](#), [46](#)
- ladder\_auth\_configure(), [44](#), [46](#)
- ladder\_deauth, [44](#), [45](#), [46](#)
- ladder\_deauth(), [45](#), [47](#)
- ladder\_has\_token, [47](#), [47](#)
- ladder\_oauth\_client
  - (ladder\_auth\_configure), [44](#)
- ladder\_token, [47](#), [47](#)
- ladder\_user, [48](#)
- LayoutPlaceholderIdMapping, [25](#), [48](#)
- LayoutProperties, [49](#), [63](#)
- LayoutReference, [25](#), [50](#)
- Line, [51](#), [66](#)
- LineConnection, [52](#), [55](#)
- LineFill, [53](#), [54](#)
- LineProperties, [51](#), [53](#), [123](#)
- Link, [39](#), [55](#), [55](#), [98](#), [117](#)
- List, [56](#), [114](#)
  
- MasterProperties, [57](#), [63](#)
- MergeTableCellsRequest, [57](#), [85](#)
  
- NestingLevel, [56](#), [58](#)
- NotesProperties, [58](#), [63](#)
  
- OpaqueColor, [10](#), [59](#), [60](#), [89](#), [102](#)
- OptionalColor, [60](#), [117](#)
- Outline, [39](#), [61](#), [98](#), [103](#), [136](#)
- OutlineFill, [61](#), [62](#)
  
- Page, [63](#), [71](#), [72](#), [101](#)
- PageBackgroundFill, [64](#), [67](#)
- PageElement, [36](#), [63](#), [65](#)
- PageElementProperties, [11](#), [13](#), [16](#), [23](#), [26](#), [28](#), [66](#)
- PageProperties, [64](#), [67](#), [126](#)
- ParagraphMarker, [68](#), [115](#)
- ParagraphStyle, [68](#), [68](#), [127](#)
- Placeholder, [38](#), [49](#), [70](#), [96](#)
- Presentation, [71](#), [73](#)
- presentations.batchUpdate, [72](#)
- presentations.create, [73](#)
- presentations.get, [73](#)
- presentations.pages.get, [74](#)
- presentations.pages.getThumbnail, [74](#)
  
- Range, [14](#), [31](#), [33](#), [75](#), [127](#), [134](#)
  
- Recolor, [39](#), [76](#)
- RefreshSheetsChartRequest, [77](#), [85](#)
- ReplaceAllShapesWithImageRequest, [78](#), [85](#)
- ReplaceAllShapesWithImageResponse, [79](#), [87](#)
- ReplaceAllShapesWithSheetsChartRequest, [80](#), [85](#)
- ReplaceAllShapesWithSheetsChartResponse, [81](#), [87](#)
- ReplaceAllTextRequest, [82](#), [84](#)
- ReplaceAllTextResponse, [82](#), [87](#)
- ReplaceImageRequest, [83](#), [86](#)
- Request, [7](#), [84](#)
- RerouteLineRequest, [86](#), [86](#)
- Response, [8](#), [87](#)
- RgbColor, [59](#), [88](#), [119](#)
  
- Shadow, [39](#), [88](#), [98](#), [103](#)
- Shape, [66](#), [90](#)
- ShapeBackgroundFill, [97](#), [98](#)
- ShapeProperties, [96](#), [98](#), [128](#)
- SheetsChart, [66](#), [99](#)
- SheetsChartProperties, [99](#), [100](#)
- Size, [66](#), [67](#), [71](#), [100](#), [104](#)
- SlideProperties, [63](#), [101](#), [129](#)
- SolidFill, [53](#), [62](#), [65](#), [97](#), [102](#), [106](#), [110](#)
- SpeakerSpotlight, [66](#), [102](#)
- SpeakerSpotlightProperties, [102](#), [103](#)
- StretchedPictureFill, [65](#), [103](#)
- string, [34](#), [36](#), [79](#), [80](#), [82](#), [120](#), [125](#), [129](#)
- SubstringMatchCriteria, [78](#), [80](#), [82](#), [104](#)
  
- Table, [66](#), [105](#)
- TableBorderCell, [106](#), [108](#)
- TableBorderFill, [106](#), [107](#)
- TableBorderProperties, [106](#), [107](#), [130](#)
- TableBorderRow, [105](#), [108](#)
- TableCell, [108](#), [113](#)
- TableCellBackgroundFill, [109](#), [111](#)
- TableCellLocation, [14](#), [31–33](#), [40](#), [41](#), [106](#), [109](#), [110](#), [112](#), [127](#), [134](#)
- TableCellProperties, [109](#), [111](#), [131](#)
- TableColumnProperties, [105](#), [112](#), [132](#)
- TableRange, [57](#), [112](#), [121](#), [130](#), [131](#)
- TableRow, [105](#), [113](#)
- TableRowProperties, [113](#), [113](#), [133](#)
- TextContent, [96](#), [109](#), [114](#)
- TextElement, [114](#), [115](#)

TextRun, [115](#), [116](#)  
TextStyle, [7](#), [9](#), [58](#), [116](#), [116](#), [134](#)  
ThemeColorPair, [10](#), [118](#)  
Thumbnail, [119](#)  
Token2.0, [43](#)

UngroupObjectsRequest, [86](#), [120](#)  
UnmergeTableCellsRequest, [85](#), [121](#)  
UpdateImagePropertiesRequest, [85](#), [121](#)  
UpdateLineCategoryRequest, [86](#), [122](#)  
UpdateLinePropertiesRequest, [85](#), [123](#)  
UpdatePageElementAltTextRequest, [86](#),  
[124](#)  
UpdatePageElementsZOrderRequest, [86](#),  
[124](#)  
UpdatePageElementTransformRequest, [84](#),  
[125](#)  
UpdatePagePropertiesRequest, [85](#), [126](#)  
UpdateParagraphStyleRequest, [85](#), [127](#)  
UpdateShapePropertiesRequest, [85](#), [128](#)  
UpdateSlidePropertiesRequest, [86](#), [128](#)  
UpdateSlidesPositionRequest, [84](#), [129](#)  
UpdateTableBorderPropertiesRequest, [85](#),  
[130](#)  
UpdateTableCellPropertiesRequest, [85](#),  
[131](#)  
UpdateTableColumnPropertiesRequest, [85](#),  
[132](#)  
UpdateTableRowPropertiesRequest, [85](#),  
[133](#)  
UpdateTextStyleRequest, [85](#), [134](#)  
UpdateVideoPropertiesRequest, [85](#), [135](#)

Video, [66](#), [135](#)  
VideoProperties, [135](#), [136](#), [136](#)

WeightedFontFamily, [118](#), [137](#)  
WordArt, [66](#), [138](#)  
WriteControl, [7](#), [8](#), [138](#)