WebCenter JavaScript API 24.03

This API can be used in the Execute JavaScript workflow node and JavaScript rule actions. Some methods are only available in one of the two, but are indicated as such below.

Interface Summary

ROMD

This interface represents the gateway to the ROMD data of a document.

ReferenceProperty

This interface describes a document reference property.

Specification

This interface describes a task specification value.

Task

This interface describes a task.

TextContent

This interface represents a text content document.

Interface ACADCanvas

This interface represents an ArtiosCAD canvas document.

Methods

getArea

`public double` **`getArea`**`()`

Get the area of this ACAD canvas document.

Returns:

The area as number (sq mm).

getBlankHeight

`public double` **`getBlankHeight`**`()`

Get the blank height of this ACAD canvas document.

Returns:

The blank height as number (mm)

getBlankLength

`public double` **`getBlankLength`**`()`

Get the blank length of this ACAD canvas document.

Returns:

The blank length as number (mm).

---

getBoardFlute

`public String` **`getBoardFlute`**`()`

Get the board flute of this ACAD canvas document.

Returns:

The board flute.

---

getBoardName

`public String` **`getBoardName`**`()`

Get the name of the board of this ACAD canvas document.

Returns:

The name of the board.

---

getCaliper

`public double` **`getCaliper`**`()`

Get the caliper of this ACAD canvas document.

Returns:

The caliper as number (mm).

---

getDepth

`public double` **`getDepth`**`()`

Get the depth of this ACAD canvas document.

Returns:

The depth as number (mm).

---

getGrainDirection

`public String **getGrainDirection**()`

Get the grain/flute direction of this ACAD canvas document.

Returns:

The grain direction. 1 for vertical, 2 for horizontal.

---

getLength

`public double **getLength**()`

Get the length of this ACAD canvas document.

Returns:

The length as number (mm).

---

getPartNames

`public String[] **getPartNames**()`

Get the part names of this ACAD canvas document.

Returns:

The names of the parts as array of strings.

---

getRuleLength

`public double **getRuleLength**()`

Get the rule length of this ACAD canvas document.

Returns:

The rule length as number (mm).

---

getWidth

`public double **getWidth**()`

Get the width of this ACAD canvas document.

Returns:

The width as number (mm).

Interface ACADDesign

This interface represents an ArtiosCAD design document.

Methods

getArea

`public double` **`getArea`**`()`

Get the area of this ACAD design document.

Returns:

The area as number (sq mm).

getBlankHeight

`public double` **`getBlankHeight`**`()`

Get the blank height of this ACAD design document.

Returns:

The blank height as number (mm)

getBlankLength

`public double` **`getBlankLength`**`()`

Get the blank length of this ACAD design document.

Returns:

The blank length as number (mm).

getBoardFlute

`public String` **`getBoardFlute`**`()`

Get the board flute of this ACAD design document.

Returns:

The board flute.

getBoardName

`public String **getBoardName**()`

Get the name of the board of this ACAD design document.

Returns:

The name of the board.

---

getCaliper

`public double **getCaliper**()`

Get the caliper of this ACAD design document.

Returns:

The caliper as number (mm).

---

getDepth

`public double **getDepth**()`

Get the depth of this ACAD design document.

Returns:

The depth as number (mm).

---

getGrainDirection

`public String **getGrainDirection**()`

Get the grain/flute direction of this ACAD design document.

Returns:

The grain direction. 1 for vertical, 2 for horizontal.

---

getLength

`public double **getLength**()`

Get the length of this ACAD design document.

Returns:

The length as number (mm).

---

getRuleLength

`public double **getRuleLength**()`

Get the rule length of this ACAD design document.

Returns:

The rule length as number (mm).

---

getWidth

`public double **getWidth**()`

Get the width of this ACAD design document.

Returns:

The width as number (mm).

Interface ACADMFG

This interface represents an ArtiosCAD manufacturing document.

Methods

getBoardFlute

`public String **getBoardFlute**()`

Get the board flute of this ACAD MFG document.

Returns:

The board flute.

---

getBoardName

`public String **getBoardName**()`

Get the name of the board of this ACAD MFG document.

Returns:

The name of the board.

---

getCaliper

`public double **getCaliper**()`

Get the caliper of this ACAD MFG document.

Returns:

The caliper as number (mm).

---

getGrainDirection

`public String **getGrainDirection**()`

Get the grain/flute direction of this ACAD MFG document.

Returns:

The grain direction. 1 for vertical, 2 for horizontal.

---

getMachines

`public String[] **getMachines**()`

Get the machines of this ACAD MFG document.

Returns:

The names of the machines as array.

---

getPrinItems

`public String[] **getPrinItems**()`

Get the print items of this ACAD MFG document.

Returns:

The names of the print items as array.

---

getRunLength

`public int **getRunLength**()`

Get the run length of this ACAD MFG document.

Returns:

The run length as number.

---

getSheetLength

```
public double getSheetLength()
```

Get the sheet length of this ACAD MFG document.

Returns:

The sheet length as number (mm).

---

getSheetWidth

```
public double getSheetWidth()
```

Get the sheet width of this ACAD MFG document.

Returns:

The sheet width as number (mm).

Interface API

This interface describes the JavaScript API.

Methods

constructURL

```
public String constructURL(String url,
                           Map parameters)
```

Parameterize a URL by replacing {placeholders} with values in a safe way. Use this function to create URLs that contain variables/data from webcenter/user. Example: constructUrl("/projects/{projectName}", {projectName: "a Project Name"}) -> /projects/a%20Project%20Name

Parameters:

url - The URL potentially containing placeholders.
parameters - A Map with variables that should be replaced.

---

doExternalApiCall

```
public ExternalAPIResult doExternalApiCall(String name,
                                           String httpMethod,
                                           String path)
```

Execute an external API call. Secrets should be configured in the External API Connection configuration page! Warning: when using data in the path of the request use API.constructURL.

Parameters:

name - The name of the configuration that you want to use.
httpMethod - The httpMethod that you want to execute. Example: GET, POST, PUT, PATCH, DELETE.
path - The path that you want to add to the baseUrl configured in the external API Connection. Example: baseUrl : https://httpbin.org , path : /bearer will result in a call to https://httpbin.org/bearer.

Returns:

An object with status code, headers and body.

---

doExternalApiCall

```
public ExternalAPIResult doExternalApiCall(String name,
                                           String httpMethod,
                                           String path,
                                           Map queryParameters)
```

Execute an external API call. Warning: do not put secrets in the queryParameters. Secrets should be configured in the External API Connection configuration page! Warning: when using data in the path of the request use API.constructURL.

Parameters:

name - The name of the External API Connection that you want to use.
httpMethod - The httpMethod that you want to execute. Example: GET, POST, PUT, PATCH, DELETE.
path - The path that you want to add to the baseUrl configured in the external API Connection. Example: baseUrl : https://httpbin.org , path : /bearer will result in a call to https://httpbin.org/bearer.
queryParameters - A map of query parameters that you want to add to the URL.

Returns:

An object with status code, headers and body.

---

doExternalApiCall

```
public ExternalAPIResult doExternalApiCall(String name,
                                           String httpMethod,
                                           String path,
                                           Map queryParameters,
                                           Map headers)
```

Execute an external API call. Warning: do not put secrets in the queryParameters, headers. Secrets should be configured in the External API Connection configuration page! Warning: when using data in the path of the request use API.constructURL.

Parameters:

name - The name of the External API Connection that you want to use.
httpMethod - The httpMethod that you want to execute. Example: GET, POST, PUT, PATCH, DELETE.
path - The path that you want to add to the baseUrl configured in the external API Connection. Example: baseUrl : https://httpbin.org , path : /bearer will result in a call to https://httpbin.org/bearer.
queryParameters - A map of query parameters that you want to add to the URL.
headers - A map of headers that you want to add to the request.

Returns:

An object with status code, headers and body.

---

doExternalApiCall

```
public ExternalAPIResult doExternalApiCall(String name,
                                           String httpMethod,
                                           String path,
                                           Map queryParameters,
                                           Map headers,
                                           Integer timeout)
```

Execute an external API call. Warning: do not put secrets in the queryParameters, headers. Secrets should be configured in the External API Connection configuration page! Warning: when using data in the path of the request use API.constructURL.

Parameters:

name - The name of the External API Connection that you want to use.
httpMethod - The httpMethod that you want to execute. Example: GET, POST, PUT, PATCH, DELETE.
path - The path that you want to add to the baseUrl configured in the external API Connection. Example: baseUrl : https://httpbin.org , path : /bearer will result in a call to https://httpbin.org/bearer.
queryParameters - A map of query parameters that you want to add to the URL.
headers - A map of headers that you want to add to the request.
timeout - A timeout in seconds for making the external API call, minimum 1 maximum 300 seconds.

Returns:

An object with status code, headers and body.

---

doExternalApiCallWithBody

```
public ExternalAPIResult doExternalApiCallWithBody(String name,
                                                   String httpMethod,
                                                   String path,
                                                   String requestBodyType,
                                                   Object requestBody)
```

Execute an external API call. Warning: do not put secrets in the queryParameters, headers. Secrets should be configured in the External API Connection configuration page! Warning: when using data in the path of the request use API.constructURL.

Parameters:

name - The name of the External API Connection that you want to use.
httpMethod - The httpMethod that you want to execute. Example: GET, POST, PUT, PATCH, DELETE.
path - The path that you want to add to the baseUrl configured in the external API Connection. Example: baseUrl : https://httpbin.org , path : /bearer will result in a call to https://httpbin.org/bearer.
requestBodyType - The type of the request body parameter: FORM, JSON, XML.
requestBody - The body to be added to the request.

Returns:

An object with status code, headers and body.

---

doExternalApiCallWithBody

```
public ExternalAPIResult doExternalApiCallWithBody(String name,
                                                   String httpMethod,
                                                   String path,
                                                   String requestBodyType,
                                                   Object requestBody,
                                                   Map queryParameters)
```

Execute an external API call with a request body (JSON / form data or an XML string). Warning: do not put secrets in the queryParameters, headers or bodyAsString. Secrets should be configured in the External API Connection configuration page! Warning: when using data in the path of the request use API.constructURL.

Parameters:

name - The name of the External API Connection that you want to use.
httpMethod - The httpMethod that you want to execute. Example: GET, POST, PUT, PATCH, DELETE.
path - The path that you want to add to the baseUrl configured in the external API Connection. Example: baseUrl : https://httpbin.org , path : /bearer will result in a call to https://httpbin.org/bearer.
requestBodyType - The type of the request body parameter: FORM, JSON, XML.
requestBody - The body to be added to the request.
queryParameters - A map of query parameters that you want to add to the URL.

Returns:

An object with status code, headers and body.

---

doExternalApiCallWithBody

```
public ExternalAPIResult doExternalApiCallWithBody(String name,
                                                   String httpMethod,
                                                   String path,
                                                   String requestBodyType,
                                                   Object requestBody,
                                                   Map queryParameters,
                                                   Map headers)
```

Execute an external API call with a request body (JSON / form data or an XML string). Warning: do not put secrets in the queryParameters, headers or bodyAsString. Secrets should be configured in the External API Connection configuration page! Warning: when using data in the path of the request use API.constructURL.

Parameters:

name - The name of the External API Connection that you want to use.
httpMethod - The httpMethod that you want to execute. Example: GET, POST, PUT, PATCH, DELETE.
path - The path that you want to add to the baseUrl configured in the external API Connection. Example: baseUrl : https://httpbin.org , path : /bearer will result in a call to https://httpbin.org/bearer.
requestBodyType - The type of the request body parameter: FORM, JSON, XML.
requestBody - The body to be added to the request.
queryParameters - A map of query parameters that you want to add to the URL.
headers - A map of headers that you want to add to the request.

Returns:

An object with status code, headers and body.

doExternalApiCallWithBody

```
public ExternalAPIResult doExternalApiCallWithBody(String name,
                                                   String httpMethod,
                                                   String path,
                                                   String requestBodyType,
                                                   Object requestBody,
                                                   Map queryParameters,
                                                   Map headers,
                                                   Integer timeout)
```

Execute an external API call with a request body (JSON / form data or an XML string). Warning: do not put secrets in the queryParameters, headers or bodyAsString. Secrets should be configured in the External API Connection configuration page! Warning: when using data in the path of the request use API.constructURL.

Parameters:

name - The name of the External API Connection that you want to use.
httpMethod - The httpMethod that you want to execute. Example: GET, POST, PUT, PATCH, DELETE.
path - The path that you want to add to the baseUrl configured in the external API Connection. Example: baseUrl : https://httpbin.org , path : /bearer will result in a call to https://httpbin.org/bearer
requestBodyType - The type of the request body parameter: FORM, JSON, XML.
requestBody - The body to be added to the request.
queryParameters - A map of query parameters that you want to add to the URL.
headers - A map of headers that you want to add to the request.
timeout - A timeout in seconds for making the external API call, minimum 1 maximum 300 seconds.

Returns:

An object with status code, headers and body.

---

getDocumentVersions

```
public DocumentVersion[] getDocumentVersions()
                                                    throws
UnsupportedOperationException
```

Get the input document versions. This is only available in the Execute JavaScript workflow node.

Returns:

An array of objects representing the input document versions.

Throws:

UnsupportedOperationException - If this method is called outside an Execute JavaScript workflow node.

---

getListData

```
public String getListData(String listName,
                          String resultColumnName,
                          String lookupColumnName,
                          String lookupValue)
```

Get data from the given list. The lookupValue(s) are searched in the lookupColumnName and the corresponding values from the resultColumnName are returned as unique set in String format: if there are multiple values, the result is concatenated with " | " just like multi-value attributes. Note: only the first 10000 items are returned.

Parameters:

listName - The name of the list in the system.
resultColumnName - The name of the list column to get the results from.
lookupColumnName - The name of the list column in which to search the lookupValue.
lookupValue - The value(s) that have to be searched in the lookupColumnName. This field can be multi-value (a string concatenated with " | ") to lookup multiple values from the same column.

Returns:

The unique set of values looked up in the list according to the given parameters. Multiple values are concatenated with " | ".

---

getListData

```
public String getListData(String listName,
                          String resultColumnName,
                          Map lookupParams)
```

Get data from the given list. The lookupParams map has the lookupColumnName as key, and lookupValue(s) as value. The lookupValue(s) are searched in the lookupColumnName and the corresponding values from the resultColumnName are returned as unique set in String format: if there are multiple values, the result is concatenated with " | " just like multi-value attributes. Note: only the first 10000 items are returned.

Parameters:

listName - The name of the list in the system.
resultColumnName - The name of the list column to get the results from.
lookupParams - The map that contains "lookupColumnName: lookupValue(s)" pairs. Multiple values are concatenated with " | ".

Returns:

The unique set of values looked up in the list according to the given parameters. Multiple values are concatenated with " | ".

---

getMarkedDocumentVersions

`public DocumentVersion[] getMarkedDocumentVersions(String mark) throws UnsupportedOperationException`

Get the list of document versions with the given mark. This is only available in the Execute JavaScript workflow node.

Parameters:

mark - The name of the mark as set by the Mark node earlier in the workflow.

Returns:

An array of objects representing the marked document versions.

Throws:

UnsupportedOperationException - If this method is called outside an Execute JavaScript workflow node.

---

getProject

`public Project getProject()`

Get the current project.

Returns:

The object representing the current project.

---

getTask

`public Task getTask() throws UnsupportedOperationException`

Get the current workflow task. This is only available in the Execute JavaScript workflow node.

Returns:

The object representing the current workflow task.

Throws:

UnsupportedOperationException - If this method is called outside an Execute JavaScript workflow node.

---

setOuputDocumentVersions

```
public void setOuputDocumentVersions(DocumentVersion[] outputDocuments)
            throws UnsupportedOperationException
```

Deprecated this method contained a typo in the name for new scripts use {@link #setOutputDocumentVersions(DocumentVersion[])} instead

Parameters:

outputDocuments -

Throws:

UnsupportedOperationException - If this method is called outside an Execute JavaScript workflow node.

---

setOutputDocumentVersions

```
public void setOutputDocumentVersions(DocumentVersion[] outputDocuments)
            throws UnsupportedOperationException
```

Set the list of output document versions. This is the list of documents that should be put on green output pin. If this method is not called, the input documents become the output documents. Duplicate document versions are filtered out. This is only available in the Execute JavaScript workflow node.

Parameters:

outputDocuments - An array of document version objects to put on the output pin.

Throws:

UnsupportedOperationException - If this method is called outside an Execute JavaScript workflow node.

Interface Attribute

This interface describes an attribute value.

Methods

addDocumentReference

```
public DocumentReference addDocumentReference(DocumentVersion documentVersion)
                                                                    throws
UnsupportedOperationException
```

Add a new document reference in case this is a document reference attribute.

Parameters:

documentVersion - The document version to add as reference.

Returns:

A document reference object that is added. Afterwards you can set the reference properties on this object.

Throws:

UnsupportedOperationException - If this is not a document reference attribute.

---

getDocumentReferences

```
public DocumentReference[] getDocumentReferences()
                                                    throws
UnsupportedOperationException
```

Get the document references in case this is a document reference attribute.

Returns:

An array of document reference (value) objects if this attribute is a document reference, otherwise an error is thrown. In case there is no value, this method returns an empty array.

Throws:

UnsupportedOperationException - If this is not a document reference attribute.

---

getName

```
public String getName()
```

Get the name of the attribute.

---

getType

```
public String getType()
```

Get the type of the attribute.

The list of possible types are (without the text between brackets):
- Text
- Integer
- Float
- Distance (mm)
- Area (sq cm)
- Volume (cu cm)
- Weight (gr)
- Distance1 (m)
- Area1 (sq mm)
- Area2 (sq m)
- Volume1 (cu dm)
- Volume2 (cu m)
- Weight1 (kg)
- DateTime
- RichText
- DocumentReference

Returns:

A String representing the type of attribute. See above list.

---

getValue

```
public String getValue()
```

Get the value of the attribute.

Returns:

The value of the attribute as String. These are the name(s) of the referenced document(s) for document reference attributes.

---

removeDocumentReference

```
public void removeDocumentReference(DocumentReference documentReference)
            throws UnsupportedOperationException
```

Remove a document reference in case this is a document reference attribute.

Parameters:

documentReference - The document reference to remove.

Throws:

UnsupportedOperationException - If this is not a document reference attribute

setValue

`public void **setValue**(String newValue)`

Set a new value for the attribute.

Parameters:

newValue - The new (String) value for the attribute.

Interface Cape

This interface represents a cape document.

Methods

getPalletNames

`public String[] **getPalletNames**()`

Get the names of the pallets of this cape document.

Returns:

The pallet names as array.

getProductCode

`public String **getProductCode**()`

Get the product code of this cape document.

Returns:

The product code.

getProductName

`public String **getProductName**()`

Get the product name of this cape document.

Returns:

The product name.

getUserText

`public String[] getUserText()`

Get the user text of this cape document.

Returns:

The user text as array.

Interface DocumentReference

This interface describes a document reference.

Methods

getDocumentVersion

`public DocumentVersion getDocumentVersion()`

Get the document version of this document reference.

Returns:

The document version of this document reference.

getReferenceProperties

`public ReferenceProperty[] getReferenceProperties()`

Get the reference properties of this document reference.

Returns:

An array of reference property objects.

getReferenceProperty

`public ReferenceProperty getReferenceProperty(String referencePropertyName)`

Get the reference property with the given name of this document reference.

Parameters:

referencePropertyName - The name of the reference property.

Returns:

An object representing the reference property with the given name or null if not found.

Interface DocumentVersion

This interface describes a document version. Only changes to the latest version will be saved.

Methods

addCharacteristicByPath

```
public void addCharacteristicByPath(String path)
            throws IllegalArgumentException
```

Add a document characteristic by path.

Parameters:

path - The path of the characteristic to add in the document in the format: "\\tree\\subtree\\subsubtree". Note: a backslash needs to be escaped in a String, that means you have to enter "\\\\tree\\\\subtree\\\\subsubtree" to make it work properly.

Throws:

IllegalArgumentException - If the path cannot be found in the system.

createColorDocumentThumbnail

```
public void createColorDocumentThumbnail()
            throws IllegalArgumentException
```

Create a new document thumbnail based on the document color attributes.

Throws:

IllegalArgumentException - If the color assets are missing on the system or the document version doesn't have the color attributes assigned.

getApprovalStatus

```
public String getApprovalStatus()
```

Get the approval status of the document version.

Returns:

The approval status of the document version or an empty string if approval was never started. For example "Pending" or "Approved".

getAttribute

`public Attribute getAttribute(String attributeName) throws IllegalArgumentException`

Get the document version's attribute (value) with the given name.

Parameters:

attributeName - The name of the attribute.

Returns:

An attribute object for the attribute value of the attribute with the given name or null if the attribute does not exist for this document version.

Throws:

IllegalArgumentException - If the provided attribute name is null.

---

getAttributeCategoryName

`public String getAttributeCategoryName()`

Get the name of the attribute category of this document version.

Returns:

The name of the attribute category of this document version or an empty String if there is no document attribute category.

---

getAttributes

`public Attribute[] getAttributes()`

Get an array of all attributes of the document version.

Returns:

An array with attribute objects representing the attributes of this document version.

---

getCharacteristicPaths

`public String[] getCharacteristicPaths()`

Get a list of characteristics set on the document.

Returns:

An array of the paths of all the characteristics of this document in the format "\\tree\\subtree\\subsubtree" or an empty array if none are set.

---

getDescription

`public String **getDescription**()`

Get the description of the document.

Returns:

The description of the document.

---

getDocumentID

`public String **getDocumentID**()`

Get the document ID of this document version.

Returns:

The document ID of this document version.

---

getDocumentType

`public String **getDocumentType**()`

Get the document type of the document.

The list of possible types are:
- ArtiosCAD Design
- ArtiosCAD Manufacturing
- ArtiosCAD 3D
- Graphics and Images
- Other
- Graphic Step and Repeat
- Digital Film
- Cape
- Desktop Graphics
- Image Graphics
- Unknown Graphics
- Page List
- Zipped Collada
- Collada
- CHILI Package
- Text Content
- ArtiosCAD Canvas
- Component

Returns:

The document type name of the file. See above list.

---

getLatestVersion

`public DocumentVersion getLatestVersion()`

Get the latest version of this document version.

Returns:

The object representing the latest document version or itself if it is already the latest version.

---

getName

`public String getName()`

Get the name of the document.

Returns:

The name of the document.

---

getROMD

`public ROMD getROMD()`

Get ROMD data of this document version. That gives you access to document type specific information.

Returns:

The ROMD object for this document version.

---

getRevisionLetter

`public String getRevisionLetter()`

Get the revision letter of this document version.

Returns:

The revision letter of this document version (starting from A). If there is no revision, this returns the empty string.

---

getVersionCounter

`public int getVersionCounter()`

Get the version number of this document version.

Returns:

The version number of this document version (starting from 1).

getVersionID

```
public String getVersionID()
```

Get the version ID of this document version.

Returns:

The document version ID of this document version.

hasCharacteristicPath

```
public boolean hasCharacteristicPath(String path)
               throws IllegalArgumentException
```

Check whether a characteristic is set in the document by path.

Parameters:

path - The path of the characteristic to check in the format: "\\tree\\subtree\\subsubtree". Note: a backslash needs to be escaped in a String, that means you have to enter "\\\\tree\\\\subtree\\\\subsubtree" to make it work properly.

Returns:

True if the characteristic is set on the document, false otherwise.

Throws:

IllegalArgumentException - If the path cannot be found in the system.

removeCharacteristicByPath

```
public void removeCharacteristicByPath(String path)
            throws IllegalArgumentException
```

Remove a document characteristic by path.

Parameters:

path - The path of the characteristic to remove from the document in the format: "\\tree\\subtree\\subsubtree". Note: a backslash needs to be escaped in a String, that means you have to enter "\\\\tree\\\\subtree\\\\subsubtree" to make it work properly.

Throws:

IllegalArgumentException - If the path cannot be found in the system.

setDescription

```
public void setDescription(String newDescription)
            throws IllegalArgumentException
```

Set the description of the document.

Parameters:

newDescription - The new description of the document.

Throws:

IllegalArgumentException - If the provided document description is null.

---

setName

```
public void setName(String newName)
            throws IllegalArgumentException
```

Set the name of the document.

Parameters:

newName - The new name of the document.

Throws:

IllegalArgumentException - If the provided document name is null.

Interface ExternalAPIResult

This interface represents the result of an external API call.

Methods

getBodyAsJSON

```
public JSONObject getBodyAsJSON()
```

Returns:

Response body as JSON object.

---

getBodyAsText

`public String` **`getBodyAsText`**`()`

Returns:

Response body as string.

---

getHeaders

`public Map` **`getHeaders`**`()`

Returns:

Response headers from the API request as Map.

---

getStatusCode

`public int` **`getStatusCode`**`()`

Returns:

Status Code returned by the API.

Interface Folder

This interface describes a folder in a project.

Methods

getDocumentVersions

`public DocumentVersion[]` **`getDocumentVersions`**`()`

Get the documents in this folder.

Returns:

An Array of all documents in this folder or an empty array if there are none in this folder.

---

getName

`public String getName()`

Get the name of the folder.

Returns:

The name of the folder.

Interface Graphic

This interface represents a graphic document.

Methods

getBarcodeCodes

`public String[] getBarcodeCodes()`

Get the barcode codes of this graphic document.

Returns:

The barcode codes as array. For example: ["0000000000000"] for an EAN13 barcode.

getDesignFiles

`public String[] getDesignFiles()`

Get the design files of this graphic document. This could for example be the path of a .ard file.

Returns:

The design files as array.

getExternalRefs

`public String[] getExternalRefs()`

Get the external references of this graphic document. This could for example be the path of the reference to a .ard file.

Returns:

The external references as array.

getFontNames

`public String[] getFontNames()`

Get the fonts of this graphic document.

Returns:

The font names as array. For example: ["Helvetica", "Helvetica Condensed"]

---

getGraphicsName

`public String getGraphicsName()`

Get the graphics name of this graphic document.

Returns:

The graphics name.

---

getGraphicsProject

`public String getGraphicsProject()`

Get the graphics project of this graphic document.

Returns:

The graphics project.

---

getHeight

`public double getHeight()`

Get the height of this graphic document.

Returns:

The height as number (mm).

---

getInks

`public String[] getInks()`

Get the inks of this graphic document.

Returns:

The inks (concatenated ink name and ink group) as array.

---

getWidth

`public double **getWidth**()`

Get the width of this graphic document.

Returns:

The width as number (mm).

Interface Image

This interface represents an image document.

Methods

getHRes

`public double **getHRes**()`

Get the horizontal resolution of this image document.

Returns:

The horizontal resolution as number (pixels/mm).

getHeight

`public double **getHeight**()`

Get the height of this image document.

Returns:

The height as number (mm).

getInks

`public String[] **getInks**()`

Get the inks of this image document.

Returns:

The inks (concatenated ink name and ink group) as array.

getVRes

```
public double getVRes()
```

Get the vertical resolution of this image document.

Returns:

The vertical resolution as number (pixels/mm).

---

getWidth

```
public double getWidth()
```

Get the width of this image document.

Returns:

The width as number (mm).

Interface Project

This interface describes a project.

Methods

addCharacteristicByPath

```
public void addCharacteristicByPath(String path)
            throws IllegalArgumentException
```

Add a project characteristic by path.

Parameters:

path - The path of the characteristic to add in the project in the format: "\\tree\\subtree\\subsubtree". Note: a backslash needs to be escaped in a String, that means you have to enter "\\\\tree\\\\subtree\\\\subsubtree" to make it work properly.

Throws:

IllegalArgumentException - If the path cannot be found in the system.

---

getAttribute

`public Attribute getAttribute(String attributeName) throws IllegalArgumentException`

Get the project attribute (value) with the given name.

Parameters:

attributeName - The name of the attribute.

Returns:

An attribute object for the attribute value of the attribute with the given name or null if the attribute does not exist for this project.

Throws:

IllegalArgumentException - If the provided attribute name is null.

---

getAttributeCategoryName

`public String getAttributeCategoryName()`

Get the name of the attribute category of this project.

Returns:

The name of the attribute category of this project or an empty String if there is no project attribute category.

---

getAttributes

`public Attribute[] getAttributes()`

Get an array of all attributes of the project

Returns:

An array with attribute objects representing the attributes of this project.

---

getCharacteristicPaths

`public String[] getCharacteristicPaths()`

Get a list of characteristics set on the project.

Returns:

An array of the paths of all the characteristics of this project in the format "\\tree\\subtree\\subsubtree" or an empty array if none are set.

---

getChildProjects

`public Project[] getChildProjects()`

Get an array of all child projects.

Returns:

An array with project objects which are child projects of this project.

---

getDescription

`public String getDescription()`

Get the description of the project.

Returns:

The description of the project.

---

getDueDate

`public long getDueDate()`

Get the due date of the project.

Returns:

The due date of the project as a long representing the number of milliseconds since January 1, 1970, 00:00:00 GMT.

---

getFolderByName

`public Folder getFolderByName(String folderName) throws IllegalArgumentException`

Get the project folder with the given name.

Parameters:

folderName - The name of the folder. Pass the empty string for selecting the root folder.

Returns:

A folder object for the folder with the given name or null if the folder does not exist for this project.

Throws:

IllegalArgumentException - If the provided folder name is null.

---

getFolders

`public Folder[] getFolders()`

Get an array of all the folders of this project.

Returns:

An array with folder objects representing the folders of this project.

---

getID

`public String getID()`

Get the ID of the project

Returns:

The ID of the project

---

getName

`public String getName()`

Get the name of the project.

Returns:

The name of the project.

---

getParentProjects

`public Project[] getParentProjects()`

Get an array of all parent projects.

Returns:

An array with project objects which are parent projects of this project.

---

getStatusList

`public ProjectStatus[] getStatusList()`

Get the list of possible project statuses for this project.

Returns:

An array (in display order - 0 based) of project status objects.

---

getStatusName

```
public String getStatusName()
```

Get the project status name.

Returns:

The name of the project status.

---

getTasks

```
public Task[] getTasks()
```

Get an array of all the tasks of this project.

Returns:

An array with task objects representing the tasks of this project.

---

hasCharacteristicPath

```
public boolean hasCharacteristicPath(String path)
               throws IllegalArgumentException
```

Check whether a characteristic is set on the project by path.

Parameters:

path - The path of the characteristic to check in the format: "\\tree\\subtree\\subsubtree". Note: a backslash needs to be escaped in a String, that means you have to enter "\\\\tree\\\\subtree\\\\subsubtree" to make it work properly.

Returns:

True if the characteristic is set on the project, false otherwise.

Throws:

IllegalArgumentException - If the path cannot be found in the system.

---

removeCharacteristicByPath

```
public void removeCharacteristicByPath(String path)
            throws IllegalArgumentException
```

Remove a project characteristic by path.

Parameters:

path - The path of the characteristic to remove from the project in the format: "\\tree\\subtree\\subsubtree". Note: a backslash needs to be escaped in a String, that means you have to enter "\\\\tree\\\\subtree\\\\subsubtree" to make it work properly.

Throws:

IllegalArgumentException - If the path cannot be found in the system or in the project.

setDescription

```
public void setDescription(String newDescription)
            throws IllegalArgumentException
```

Set the description of the project.

Parameters:

newDescription - The new description of the project.

Throws:

IllegalArgumentException - If the provided project description is null.

setDueDate

```
public void setDueDate(long newDueDate)
            throws UnsupportedOperationException
```

Set the new due date of the project.

Parameters:

newDueDate - The new due date for the project as a long representing the number of milliseconds since January 1, 1970, 00:00:00 GMT.

Throws:

UnsupportedOperationException - If the project due date is linked to an attribute.

setName

```
public void setName(String newName)
            throws IllegalArgumentException
```

Set the name of the project.

Parameters:

newName - The new name of the project.

Throws:

IllegalArgumentException - If the provided project name is null or empty.

setStatusByName

```
public void setStatusByName(String newProjectStatusName)
            throws IllegalArgumentException
```

Set a new project status by name.

Parameters:

newProjectStatusName - The new project status name. This is case insensitive.

Throws:

IllegalArgumentException - If the provided project status name is null or if the status cannot be found in the system

Interface ProjectStatus

This interface represents a project status.

Methods

getDisplayOrder

```
public int getDisplayOrder()
```

Get the display order of the project status.

Returns:

The display order as int (0 based).

getLinkedAttributeName

```
public String getLinkedAttributeName()
```

Get the name of the attribute that is linked to this project status.

Returns:

The name of the attribute or the empty string if no attribute link is set.

getName

`public String **getName**()`

Get the name of the project status.

Returns:

The name of the project status.

---

getRGBColor

`public String **getRGBColor**()`

Get the RGB color of the project status.

Returns:

The RGB color of the project status in hex format (e.g. #efefef).

---

getType

`public int **getType**()`

Get the type of project status.

Returns:

The type of the project status as int. 0 = Non-Active, 1 = Active, 2 = Completed, 3 = Archived.

Interface ROMD

This interface represents the gateway to the ROMD data of a document.

Methods

getACADCanvas

`public ACADCanvas getACADCanvas() throws UnsupportedOperationException`

Get the ACAD canvas data.

Returns:

The ACAD canvas object.

Throws:

UnsupportedOperationException - If this is not an ACAD canvas document.

---

getACADDesign

`public ACADDesign getACADDesign() throws UnsupportedOperationException`

Get the ACAD design data.

Returns:

The ACAD design object.

Throws:

UnsupportedOperationException - If this is not an ArtiosCAD design document.

---

getACADMFG

`public ACADMFG getACADMFG() throws UnsupportedOperationException`

Get the ACAD manufacturing data.

Returns:

The ACAD MFG object.

Throws:

UnsupportedOperationException - If this is not an ArtiosCAD manufacturing document.

---

getCape

`public Cape getCape() throws UnsupportedOperationException`

Get the cape data.

Returns:

The cape object.

Throws:

UnsupportedOperationException - If this is not a cape document.

---

getGraphic

`public Graphic getGraphic() throws UnsupportedOperationException`

Get the graphic data.

Returns:

The graphic object.

Throws:

UnsupportedOperationException - If this is not an graphic document.

---

getImage

`public Image getImage() throws UnsupportedOperationException`

Get the image data.

Returns:

The image object.

Throws:

UnsupportedOperationException - If this is not an image document.

---

getTextContent

`public TextContent getTextContent() throws UnsupportedOperationException`

Get the text content data.

Returns:

The text content object.

Throws:

UnsupportedOperationException - If this is not a text content document.

Interface ReferenceProperty

This interface describes a document reference property.

Methods

getName

`public String getName()`

Get the name of the reference property.

getValue

`public String getValue()`

Get the value of the reference property.

Returns:

The value of the reference property as String.

setValue

`public void setValue(String newValue)`

Set a new value for the reference property.

Parameters:

newValue - The new (String) value for the reference property.

Interface Specification

This interface describes a task specification value.

Methods

addDocumentReference

`public DocumentReference addDocumentReference(DocumentVersion documentVersion) throws UnsupportedOperationException`

Add a new document reference in case this is a document reference specification.

Parameters:

documentVersion - The document version to add as reference.

Returns:

A document reference object that is added. Afterwards you can set the reference properties on this object.

Throws:

UnsupportedOperationException - If this is not a document reference specification.

getDocumentReferences

`public DocumentReference[] getDocumentReferences() throws UnsupportedOperationException`

Get the document references in case this is a document reference specification.

Returns:

An array of document reference (value) objects if this specification is a document reference, otherwise an error is thrown. In case there is no value, this method returns an empty array.

Throws:

UnsupportedOperationException - If this is not a document reference specification.

getName

`public String getName()`

Get the name of the specification.

getType

```
public String getType()
```

Get the type of the specification.

The list of possible types are (without the text between brackets):
- Text
- Integer
- Float
- Distance (mm)
- Area (sq cm)
- Volume (cu cm)
- Weight (gr)
- Distance1 (m)
- Area1 (sq mm)
- Area2 (sq m)
- Volume1 (cu dm)
- Volume2 (cu m)
- Weight1 (kg)
- DateTime
- RichText
- DocumentReference

Returns:

A String representing the type of specification. See above list.

---

getValue

```
public String getValue()
```

Get the value of the specification.

Returns:

The value of the specification as String. These are the name(s) of the referenced document(s) for document reference specifications.

---

removeDocumentReference

```
public void removeDocumentReference(DocumentReference documentReference)
            throws UnsupportedOperationException
```

Remove a document reference in case this is a document reference specification.

Parameters:

documentReference - The document reference to remove.

Throws:

UnsupportedOperationException - If this is not a document reference specification

setValue

`public void **setValue**(String newValue)`

Set a new value for the specification.

Parameters:

newValue - The new (String) value for the specification.

Interface Task

This interface describes a task.

Methods

getCompletedDate

`public long **getCompletedDate**()`

Get the completed date of the task.

Returns:

The completed date of the task as a long representing the number of milliseconds since January 1, 1970, 00:00:00 GMT. If the task is not completed yet, 0 is returned.

getCreatedDate

`public long **getCreatedDate**()`

Get the created date of the task.

Returns:

The created date of the task as a long representing the number of milliseconds since January 1, 1970, 00:00:00 GMT.

getDueDate

`public long **getDueDate**()`

Get the due date of the task.

Returns:

The due date of the task as a long representing the number of milliseconds since January 1, 1970, 00:00:00 GMT.

getEstimatedDate

`public long **getEstimatedDate**()`

Get the estimated date of the task.

Returns:

The estimated date of the task as a long representing the number of milliseconds since January 1, 1970, 00:00:00 GMT. If not filled in, 0 is returned.

getID

`public String **getID**()`

Returns:

The ID of current task.

getLeadTime

`public long **getLeadTime**()`

Get the lead time of the task.

Returns:

The lead time of the task in milliseconds.

getName

`public String **getName**()`

Get the name of the task.

Returns:

The name of this task.

getSpecification

`public Specification getSpecification(String specificationName) throws IllegalArgumentException`

Get the task specification (value) with the given name.

Parameters:

specificationName - The name of the specification

Returns:

A specification object for the specification value of the specification with the given name or null if the specification does not exist for this task.

Throws:

IllegalArgumentException - If the provided specification name is null.

---

getSpecifications

`public Specification[] getSpecifications()`

Get an array of all specifications of the task

Returns:

An array with specification objects representing the specifications of this task.

---

getStartedDate

`public long getStartedDate()`

Get the started date of the task.

Returns:

The started date of the task as a long representing the number of milliseconds since January 1, 1970, 00:00:00 GMT. If the task is not started yet, 0 is returned.

---

getStatusName

`public String getStatusName()`

Get the name of the task status of the task.

Returns:

The name of the task status of the task.

---

getTaskTypeName

```
public String getTaskTypeName()
```

Get the name of the task type.

Returns:

The name of the task type of this task.

---

setDueDate

```
public void setDueDate(long newDueDate)
            throws UnsupportedOperationException
```

Set the new due date of the task.

Parameters:

newDueDate - The new due date for the task as a long representing the number of milliseconds since January 1, 1970, 00:00:00 GMT.

Throws:

UnsupportedOperationException - If the task due date is linked to an attribute.

Interface TextContent

This interface represents a text content document.

Methods

getDocumentStatus

```
public String getDocumentStatus()
```

Get the document (approval) status of this text content document.

Returns:

The document (approval) status.

---

getElementType

`public String` **`getElementType`**`()`

Get the element type of this text content document.

Returns:

The element type.