

Translation Development Kit Guide

HelloTranslate comes with a connectivity framework. After defining a connection to your content, you can navigate the repository with our content browser and pickup content for translation manually or automate the selection based on metadata.

HelloTranslate Translation Development Kit is a way how to integrate your content into this connectivity framework and use all the described features with your content. To integrate your content, you can choose your favorite technology to build a web service capable of

- delivering the metadata of your content
- navigating your content repository
- downloading binary content
- uploading translations

Your web service will expose an endpoint and HelloTranslate will send requests to this endpoint with JSON payload to specify the operation to perform and its parameters. The web service will respond with a JSON response describing the metadata of the content, deliver the binary content or simply confirm the translation creation.

JSON RPC 2.0 Protocol

To delegate some of the content operations to your custom service, HelloTranslate connectivity framework uses a JSON RPC 2.0 Protocol. It is a stateless, light-weight remote procedure call (RPC) protocol and it uses [JSON \(RFC 4627\)](#) as data format. You can check the full [JSON RPC Specification](#).

Requests

An RPC call is represented by sending a Request object to your service. The Request object has the following members:

- **jsonrpc** - A String specifying the version of the JSON-RPC protocol. Will be exactly "2.0".
- **id** - A String identifier of the request, eg. "4ab6901a-a749-4026-afd1-2e13307fa283". Your service must reply with the same value in the Response object, if included.
- **method** - A String containing the name of the method to be invoked, eg. "entity.get".
- **params** - A nested object that holds the parameters to be used during the invocation of the method.

Responses

The Response is expressed as a single JSON Object, with the following members:

- **jsonrpc** - A String specifying the version of the JSON-RPC protocol. MUST be exactly "2.0".
- **id** - This member is required, and it must be the same as the value of the id member in the Request Object.
- **result** - This member is required on success. This member must not exist if there was an error invoking the method. The value of this member is determined by the method invoked.
- **error** - This member is required on error. This member must not exist if there was no error triggered during invocation. The value for this member must be an Object as defined below.

Either the result member or error member must be included, but both members must not be included.

Error Objects

When a RPC call encounters an error, the response object must contain the error member with a value that is an object with the following members:

- **code** - A Number that indicates the error type that occurred. This must be an integer.
- **message** - A String providing a short description of the error.
- **data** - A Primitive or Structured value that contains additional information about the error. This may be omitted.

HelloTranslate JSON RPC implementation

HelloTranslate uses only a subset of the JSON RPC standard:

- The transport layer to be used to exchange requests and responses is HTTP(S).
- In absence of technical failures your services always returns an HTTP 200 (OK) state with a response. Any other state code signals the service is defective or malfunctioning.
- The protocol is synchronous. HelloTranslate will send a request to execute a method, then wait for the response to be ready. A timeout is set to limit the wait in case of malfunctions in the external service.
- Notifications (requests without an expected response) are not used.
- The parameters for the method invocation must be transferred as named values: the positional format is *not* supported.

Configuring connection to your content

HelloTranslate offers a user interface to configure a connection to your content.

You need to provide the URL of your service, and you can define a `config` JSON object. HelloTranslate will send the `config` to your service with each request.

You can use the `config` e.g., for an authentication mechanism, or to specify the part of the content repository you want to access, etc. It is completely up to you.

Your service is responsible for validating the configuration it receives.

The `config` does not support array values or nested objects.

Describing entity metadata

HelloTranslate uses generic metadata model, which allows connecting all kinds of different content repositories.

Content metadata are always represented as JSON, which contains the following members:

Field	Type	Description
id	URL	A URL path that points to the entity.
xdip	URL	A URL that is used internally to point to the entity.
kind	String	A simple content type string that describes the best match for the entity.
original	Object	An object containing all decorators as they were found in the content repository.
modified	Object	An object containing all decorators as they will be represented in the content repository. This field is identical to <code>original</code> on all entities that are retrieved from the api. It is used only to update the metadata of entity.

XDIPs and IDs

To locate documents in a repository, an internal protocol called XDIP is used. The XDIP and the request URLs are connector-specific. XDIPs follow the URL syntax

- protocol is `xdip`
- hostname is a configurationID (identification of the connection you've created above)
- path identifies the entity in the repository

For example

- `xdip://our-cms/test-folder/test-file.xml`
- `xdip://our-cms/84e2a30b-1aa5-4291-af8a-06b5fc204c65`

Decorators

Within an entity, the 'original' and 'modified' sections contain decorators. Decorators are named sets of closely related fields. A collection of decorators is provided to represent most common metadata fields. Fields must be of one of the following types:

Type	Description
BOOLEAN	Either true or false.
NUMBER	An integer or decimal number.
STRING	A textual value. A blank string value means: "intentionally left blank".
URL	A URL formatted string value.
DATE	An ISO date formatted string value. Format: yyyy-MM-ddTHH:mm:ss.sssTZD. The date format complies with https://www.w3.org/TR/NOTE-datetime .
ENUM	A STRING with limited possible values.
OBJECT	A nested object containing key-value pairs.

For each field an indicator is given whether the field is required or optional. Required fields cannot be omitted and cannot have value 'null'. Optional fields can be set to null or may be omitted entirely.

Container decorator

The `container` decorator signifies an entity that can have children. If the decorator is not present on an entity, the entity cannot have children. For example, this decorator is not present on an entity representing a file in a file system.

When the container decorator is present, the entity can have children. For example, this decorator is present on an entity representing a folder in a file system.

Field Name	Description	Type	Required
<code>hasChildren</code>	Set to <code>true</code> if this entity might have children. Set to <code>false</code> if it is certain that the object does not have any children.	BOOLEAN	<code>true</code>

```
{
  "container": {
    "hasChildren": true
  }
}
```

ContentType decorator

The `contentType` decorator contains the content type used by the target system. The exact value differs for every implementation. Feel free to use any types that describe your types of content well.

Field Name	Description	Type	Required
<code>displayName</code>	The human readable name of the content type.	STRING	
<code>systemName</code>	The technical name of the content type. Can also be a system specific unique id or name.	STRING	true

```
{
  "contentType": {
    "systemName": "86797349759843867824",
    "displayName": "FileMetadata"
  }
}
```

Created decorator

The `created` decorator signifies when an entity was created. This decorator is important for the automation feature of HelloTranslate when we look for a content that is new or has been changed.

Field Name	Description	Type	Required
<code>date</code>	The creation date of the entity. Has millisecond accuracy. The time zone is formatted as <code>Z</code> for UTC or <code>+02:00</code> otherwise.	DATE	true

```
{
  "created": {
    "date": "2017-08-22T14:29:53.000+02:00"
  }
}
```

File decorator

The `file` decorator contains various pieces of information related to files on a file system. This decorator is mandatory for all entities with binary content. Not specifying this decorator will cause HelloTranslate not to download the content of this entity.

Field Name	Description	Type	Required
<code>rawExtension</code>	The extension of the file as found on the filesystem. Contains the empty string ("") when the file does not have an extension.	STRING	true
<code>size</code>	The size of the file in bytes.	NUMBER	true

```
{
  "file": {
    "rawExtension": "JPG",
    "size": 12697
  }
}
```

Language decorator

The `language` decorator contains information about the language of the entity's content. Whenever a *language tag* is mentioned, we refer to strings compliant with [BCP-47](#).

Field Name	Description	Type	Required
<code>tag</code>	The language tag for this entity.	STRING	true
<code>translationOf</code>	A reference (id) to the entity that is the master from which this one was translated.	STRING	

```
{
  "tag": "it-IT"
}
```

MimeType decorator

The `mimeType` decorator contains a reference to the type of media that is contained by the entity.

Field Name	Description	Type	Required
<code>type</code>	The MIME Type of the entity's contents.	STRING	true

```
{
  "mimeType": {
    "type": "text/plain"
  }
}
```

Modified decorator

The `modified` decorator signifies when an entity was last modified. This decorator is important for the automation feature of HelloTranslate when we look for a content that has changed.

Field Name	Description	Type	Required
<code>date</code>	The last modification date of the entity. Has millisecond accuracy. The time zone is formatted as <code>Z</code> for UTC or <code>+02:00</code> otherwise.	DATE	true

```
{
  "modified": {
    "date": "2017-08-22T14:29:53.000+02:00"
  }
}
```

Name decorator

The `name` decorator represents both a technical name and a human readable name for an entity. This could be anything from the title of a web page to the filename of a document.

Field Name	Description	Type	Required
<code>displayName</code>	The human readable name of the entity.	STRING	
<code>systemName</code>	The technical name of the entity. Can also be a system specific unique id or name.	STRING	true

```
{
  "name": {
    "displayName": "How to integrate HelloTranslate",
    "systemName": "how-to-integrate-hellotranslate"
  }
}
```

Parent decorator

The `parent` decorator contains a reference to the entity's parent.

Field Name	Description	Type	Required
<code>id</code>	The URL path that points to the entity's parent.	STRING	true

```
{  
  "parent": {  
    "id": "/59e4b5a6857aba0005a28228/images/2017"  
  }  
}
```

Navigating content repository

To implement navigation to your content you need to make your service ready for the requests like this

```
{
  "jsonrpc": "2.0",
  "id": "84e2a30b-1aa5-4291-af8a-06b5fc204c65",
  "method": "entity.get",
  "params": {
    "config": {
      "respository": "tests"
    },
    "xdip": "xdip://my-connector/",
    "requestParameters": {
      "projectionScopes": [
        "path_children_reference"
      ],
      "projectionIncludes": [],
      "projectionExcludes": [],
      "offset": null,
      "limit": null
    }
  }
}
```

and this

```
{
  "jsonrpc": "2.0",
  "id": "4ab6901a-a749-4026-afd1-2e13307fa283",
  "method": "entity.get",
  "params": {
    "config": {
      "respository": "tests"
    },
    "xdip": "xdip://my-connector/",
    "requestParameters": {
      "projectionScopes": [
        "path_children_entity"
      ],
      "projectionIncludes": [],
      "projectionExcludes": [],
      "offset": null,
      "limit": null
    }
  }
}
```

With these JSON RPC requests HelloTranslate is exploring the structure of your content. Both the requests ask for list of children of the certain container.

The method will be “entity.get”.

The parameter `requestParameters.projectionScopes` tells you to return children of an entity. The scope `“path_children_reference“` requests only the list of ids of the entities while the scope `“path_children_entity“` requests the full metadata of the entities.

The parameter `xdip` identifies the entity whose children HelloTranslate needs to retrieve.

As described earlier, the parameter `config` is the exact copy of the object you defined when configuring the connection to your content.

The `result` element of the response object is populated with different information depending on the requested scopes. Eg. the response to the second request can look like this

```
{
  "jsonrpc": "2.0",
  "id": "84e2a30b-1aa5-4291-af8a-06b5fc204c65",
  "result": {
    "path_children_entity": [
      {
        "id": "media",
        "kind": "Folder",
        "original": {
          "container": {
            "hasChildren": true
          },
          "contentType": {
            "systemName": "Folder"
          },
          "created": {
            "date": "2022-04-20T10:00:50.770Z"
          },
          "modified": {
            "date": "2022-04-20T10:00:50.770Z"
          },
          "name": {
            "systemName": "media"
          },
          "parent": {
            "id": "xdip://my-connector/"
          }
        }
      },
      {
        "id": "pages",
        "kind": "Folder",
        "original": {
          "container": {
            "hasChildren": true
          },
          "contentType": {
            "systemName": "Folder"
          },
          "created": {
            "date": "2022-04-20T10:06:32.038Z"
          }
        }
      }
    ]
  }
}
```

```
        "modified": {
          "date": "2022-04-20T10:06:32.038Z"
        },
        "name": {
          "systemName": "pages"
        },
        "parent": {
          "id": "xdip://my-connector/"
        }
      }
    },
    {
      "id": "posts",
      "kind": "Folder",
      "original": {
        "container": {
          "hasChildren": true
        },
        "contentType": {
          "systemName": "Folder"
        },
        "created": {
          "date": "2022-04-20T10:06:32.040Z"
        },
        "modified": {
          "date": "2022-04-20T10:06:32.040Z"
        },
        "name": {
          "systemName": "posts"
        },
        "parent": {
          "id": "xdip://my-connector/"
        }
      }
    },
    {
      "id": "test-file.json",
      "kind": "File",
      "original": {
        "contentType": {
          "systemName": "File"
        },
        "created": {
          "date": "2022-04-20T09:52:08.131Z"
        },
        "language": {
          "tag": "en-US"
        },
        "mimeType": {
          "type": "application/json"
        },
        "file": {
          "rawExtension": "json",
          "size": 103
        },
        "modified": {
          "date": "2022-04-20T09:52:08.131Z"
        }
      }
    }
  ]
}
```

```
    },
    "name": {
      "systemName": "test-file.json"
    },
    "parent": {
      "id": "xdip://my-connector/"
    }
  }
}
]
```

It is indicating that there are three folders (called media, pages and posts) and one JSON test file in the root of the selected repository.

Downloading Binary Contents

To allow downloading of content you need to make your service ready for requests like this

```
{
  "jsonrpc": "2.0",
  "id": "4ab6901a-a749-4026-afd1-2e13307fa283",
  "method": "entity.get-binary",
  "params": {
    "config": {
      "respository": "tests"
    },
    "xdip": "xdip://my-connector/pages/123"
  }
}
```

The method will be “entity.get-binary“.

The parameter `xdip` identifies the entity whose contents HelloTranslate needs to retrieve.

The parameter `config` is the exact copy of the object you defined when configuring the connection to your content.

As a result of the response object you must send the Base64 encoded binary content of the entity, without line-length limit as in [RFC 4648](#). Like this

```
{
  "jsonrpc" : 2.0,
  "id" : "4ab6901a-a749-4026-afd1-2e13307fa283",
  "result" : "<Base64 encoded contents>"
}
```

Uploading Translations

When uploading translated content to your content repository HelloTranslate will send a JSON RPC request like this

```
{
  "jsonrpc": "2.0",
  "id": "4ab6901a-a749-4026-afd1-2e13307fa283",
  "method": "entity.create",
  "params": {
    "config": {
      "respository": "tests"
    },
    "requestParameters": {
      "projectionScopes": ["entity"]
    },
    "entity": {
      "kind" : "File",
      "original" : {
```

```

        "name" : {
            "systemName" : "notes",
            "displayName" : "notes"
        },
        "language" : {
            "tag" : "fr-FR",
            "translationOf" : "/pages/123"
        }
    },
    "binaryContents": "YmluYXJ5Q29udGVudHM="
}
}

```

The method will be “entity.create“.

The parameter `entity` will describe the metadata of the translation, including the language decorator specifying the language of the translation and the reference to the original content that has been translated.

The parameter `binaryContents` carries the translated content encoded in Base64 format without line length limits as in [RFC 4648](#).

As a result of the response object you must send the full metadata of the newly created entity. Like this

```

{
  "jsonrpc" : 2.0,
  "id" : "4ab6901a-a749-4026-afd1-2e13307fa283",
  "result" : {
    "entity" : {
      "id": "test-file.json",
      "kind": "File",
      "original": {
        "contentType": {
          "systemName": "File"
        },
      },
      "created": {
        "date": "2022-04-20T09:52:08.131Z"
      },
      "language": {
        "tag": "en-US"
      },
      "mimeType": {
        "type": "application/json"
      },
      "file": {
        "rawExtension": "json",
        "size": 103
      },
      "modified": {
        "date": "2022-04-20T09:52:08.131Z"
      },
      "name": {

```

```
        "systemName": "test-file.json"
    },
    "parent": {
        "id": "xdip://my-connector/"
    }
}
}
```