TMB API

This is a list of all the available APIs of the Time Machine Box for collections and items data retrieving.

We have extended the IIIF Toolkit Omeka plugin API to add several request types to get collections, manifests, sequences.

Description
This function returns the list of all the collections loaded in Omeka.

REST action method
GET

Link
http://iiif-server.mind-ware.it/omeka/api/oa/collections?key=YOUR_OMEKA_API_KEY

Response
Json Manifest of all collections

Example

```
{
      "@context": "http://iiif.io/api/presentation/2/context.json",
      "@id": "http://iiif-server.mind-ware.it/omeka/api/oa",
      "@type": "sc:Collection",
      "label": "Top Level Collection",
      "manifests": [
                  {
                        "@id":
"http://iiif-server.mind-ware.it/omeka/oa/collections/1/manifest.json",
                        "@type": "sc:Manifest",
                        "label": "Quaderno dei trasporti detto Fia E - 1526"
                  },
                  {
                        "@id":
"http://iiif-server.mind-ware.it/omeka/oa/collections/2/manifest.json",
                        "@type": "sc:Manifest",
                        "label": "S. Nicolò - 439 - 1"
                  },
                  ...
      ]
}
```

Description
This function returns the manifest of a collection loaded in Omeka.

REST action method
GET

Link
http://iiif-server.mind-ware.it/omeka/api/oa/collections/{collection-id}?key=YOUR_OMEKA_API_KEY

Response
Json Manifest of collection with id *collection-id*

Example
```
{
"@context": "http://www.shared-canvas.org/ns/context.json",
      "@id":
"http://iiif-server.mind-ware.it/omeka/oa/collections/2/manifest.json",
      "@type": "sc:Manifest",
      "label": "439-1",
      "sequences": [
      {
                  "@id":
"http://iiif-server.mind-ware.it/omeka/oa/collections/2/sequence.json",
                  "@type": "sc:Sequence",
                  "label": "",
                  "canvases": [...],
                  "metadata": [...]
      }
}
```

Description
This function returns the manifest of an item loaded in Omeka.

REST action method
GET

Link
http://iiif-server.mind-ware.it/omeka/api/oa/items/{item-id}?key=YOUR_OMEKA_API_KEY

Response
Json Manifest of item with id *item-id*

Example
```
{
"@context": "http://www.shared-canvas.org/ns/context.json",
"@id": "http://iiif-server.mind-ware.it/omeka/oa/items/2/manifest.json",
      "@type": "sc:Manifest",
      "label": "Quaderno dei trasporti detto Fia E - 1526",
      "sequences": [
      {
            "@id":
"http://iiif-server.mind-ware.it/omeka/oa/items/2/sequence.json",
            "@type": "sc:Sequence",
            "label": "",
            "canvases": [...],
            "metadata": [...],
            …
      }
}
```

IIIF Image API

IIIF Image API URLs for an image request are called this way:

http[s]://{server}/{digilib-webapp}/Scaler/{iiif-prefix}/{identifier}/{region}/{size}/{rotation}/{quality}.{format}

where *digilib-webapp* is the name of the digilib web application in the servlet container.

The value of iiif-prefix is defined by the iiif-prefix parameter in digilib-config. The default value is “IIIF”.

The identifier part of the URL must not contain any slashes. Since the identifier is mapped to the digilib *fn-parameter*, which is a filesystem path that likely contains slashes separating subdirectories, all occurrences of a slash have to be replaced by the value of the *iiif-slash-replacement* parameter in *digilib-config*. The default value of the replacement string is “!”, so the *fn-path* “books/book1/page0002” becomes the identifier “books!book1!page0002”.

For a definition of the other parameters region, size, rotation, quality, and format please read the IIIF Image API docs.

Example
To view the **5be182a072661e54169f363fd24ac45b.jpg** image at full resolution the url will be:

http://iiif-server.mind-ware.it:8080/digilib/Scaler/IIIF/!5be182a072661e54169f363fd24ac45b.jpg/full/full/0/default.jpg

IIIF Presentation API

Digilib provides the optional Manifester servlet that generates simple IIIF Presentation API version 2 manifests that can be used with any IIIF viewer to navigate a directory full of images using functions as a book-reader or light-table.
The Manifester servlet URLs have the form:

http[s]://{server}/{digilib-webapp}/Manifester/{iiif-prefix}/{identifier}

(!) This feature is not used because omeka save images in one folder, so th manifest of any collection can be retrieved directly through Omeka.

Omeka API wrapper

A PHP wrapper has been created for an easy access to the omeka API.
For example you can use it to download a manifest and all the related images of a group of collections that are have been created or updated In Omeka at some point in time.

Omeka API Wrapper Installation

1. Download http://www.timemachinebox.eu/sites/default/files/download/tmb-api.tar.gz
2. Unzip file *tmb-api.tar.gz* and open terminal in the *tmb-api* folder
3. Execute these commands from terminal:
    a. php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
    b. php -r "if (hash_file('SHA384', 'composer-setup.php') === '544e09ee996cdf60ece3804abc52599c22b1f40f4323403c44d44fdfdd586475ca9813a858088ffbc1f233e9b180f061') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
    c. php composer-setup.php
    d. php -r "unlink('composer-setup.php');"
4. After composer installation execute this command for installing dependencies:
    a. php composer.phar install
5. Now you can use the wrapper for easy access of the omeka API. You can view test.php for a direct example.

Omeka API Wrapper built-in functions

List of built-in functions of the *OmekaAPIUtilities* class. To use these functions you have to create an instance of the the *OmekaAPIUtilities* class.
For example:

```
$omekaAPIUtilities = new OmekaAPIUtilities();
$omekaAPIUtilities->getCollections('2017-09-13T05:45:00-05:00',
'2017-09-13T05:45:00-05:00', 'files');
```

Function name
getCollections

Class
OmekaAPIUtilities

Requires
vendor/autoload.php
omeka/omekaAPIUtilities.php

Description
This function returns a set of collections based on creation and edit date.

Params

$added_since	*date of collection creation*
$modified_since	*date of collection editing*
$download_type	*type of download*; it can have these values: **manifest**, **all**, **files**

REST action method

GET

Description

This function use dates in ISO8601 format. The *download_type* param allows you to choose what to download:

- **manifest →** the manifest of the filtered collection
- **all →** the manifest with all images and all images manifests of the filtered collection
- **files →** the manifest with all images of the filtered collection

Example

```
$omekaAPIUtilities->getCollections('2017-09-13T05:45:00-05:00',
'2017-09-13T05:45:00-05:00', 'files');
```

Function name

getItems

Class

OmekaAPIUtilities

Requires

vendor/autoload.php
omeka/omekaAPIUtilities.php

Description

This function returns items based on creation and edit date.

Params

$added_since	*date of collection inserting*
$modified_since	*date of collection editing*
$download_type	*type of download*; it can have these values: **manifest**, **all**, **files**

REST action method

GET

Description

This function use dates in ISO8601. The type of download allows to choose if you want to download:

- **manifest →** all the manifests of the filtered items
- **all →** all the manifests and the images of the filtered items
- **files →** all the images of the filtered items

Example

```
$omekaAPIUtilities->getItems('2017-09-13T05:45:00-05:00',
'2017-09-13T05:45:00-05:00', 'files');
```

Function name
collectionFilesDownload

Class
OmekaAPIUtilities

Requires
vendor/autoload.php
omeka/omekaAPIUtilities.php

Description
This function let you get all the related files of a collection.

Params

$collection_id	*id of the omeka collection*
$download_type	*type of download*; it can have these values: **manifest**, **all**, **files**

REST action method
GET

Description
The type of download allows to choose if you want to download:

- **manifest →** the collection manifest
- **all →** the collection manifest with all images and all images manifests
- **files →** the collection manifest with all images

Example

```
$omekaAPIUtilities->collectionFilesDownload(1, 'files');
```

Function name
itemFilesDownload

Class
OmekaAPIUtilities

Requires

vendor/autoload.php
omeka/omekaAPIUtilities.php

Description
This function let you get the related files of an item.

Params

$item_id	*id of the omeka item*
$download_type	*type of download*; it can have these values: **manifest**, **all**, **files**

REST action method
GET

Description
The download_type param let to choose what to download:

● **manifest** →	the item manifest
● **all** →	the item manifest with all images and all images manifests
● **files** →	the item manifest with all images

Example
```
$omekaAPIUtilities->itemFilesDownload(1, 'files');
```