Digital Cultural Heritage APIs

Galina Bogdanova[1][0000-0002-5463-4274], Todor Todorov[1, 2][0000-0002-2443-4618],
Nikolay Noev[1][0000-0003-3290-1439]

[1] Institute of mathematics and informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria
[2] St. Cyril and St. Methodius University of Veliko Tarnovo, Veliko Tarnovo, Bulgaria
`galina@math.bas.bg, todor@math.bas.bg, nickey@math.bas.bg`

Abstract. In the paper we present services provided by some of the popular cultural heritage application programming interfaces (API). We show how they could be used from third applications software applications. Finally, we develop tools that use this APIs in order to search and visualize data.

Keywords: Digital Cultural Heritage, API, Europeana.

1   Introduction

Application programming interface (API) contains services for software development (Jacobson, Brail, & Woods, 2011), (Richardson, Amundsen, & Ruby, 2013). It contains well defined functions for interaction between components. Web API contains functionality available through public endpoints. In most cases request and response messages are in JSON or XML format.

In recent years cultural heritage is bring to another level with the usage of digital technologies (Bogdanova, Todorov, & Noev, 2018), (Kremers, 2019). Now people have wider opportunities to use cultural materials and the institutions can interact with new users and develop new content for research and education.

Most of the popular digital cultural heritage libraries offer their digitized content for usage via specially developer APIs (British Museum API[1], Colada[2], Europeana[3], Harvard Art Museums API[4], New York Public Library Digital Collections[5], Rijksmuseum API[6]). In our research we make an overview of the services provided by the APIs of two of the biggest digital libraries Europeana [7] and New York public library (NYPL) [9]. We also present our custom software applications that use these APIs in order to retrieve and represent digital cultural heritage content.

---

[1] British Museum API. https://www.programmableweb.com/api/british-museum

[2] Colada. https://github.com/netsensei/europeana

[3] Europeana. API, https://pro.europeana.eu/resources/apis

[4] Harvard Art Museums API. https://www.harvardartmuseums.org/collections/api

[5] New York Public Library Digital Collections. http://api.repo.nypl.org/

[6] Rijksmuseum API. https://www.rijksmuseum.nl/en/api?lang=en

2 Overview of Cultural Heritage APIs

2.1 Europeana

Europeana developed a digital library that contains a rich number of digitized cultural heritage materials. In recent years a lot of cultural institutions provide digital data to Europeana.

Europeana APIs provide functions for application development that uses the rich collections of objects from the biggest major museums and galleries. It includes over 50 million cultural heritage items from over 3,500 cultural institutions across Europe.

In order to use Europeana APIs we should have a valid API-key. It's provided to the user for free but after under special terms of use. API-keys are for personal usage and should not be provided to third. A separate API-key is needed for each application.

Next we will consider usage of two of the most popular Europeana APIs.

Europeana Record API

The Record API is a web interface to the Europeana data. HTTPS protocol is used for requests to the Record API in the following format:

https://www.europeana.eu/api/v2/record/[RECORD_ID].[FORMAT]

RECORD_ID is the identifier of the record in the internal dataset.
FORMAT is the requested data output formats.
The API supports three serialization formats: JSON, JSON-LD and RDF/XML.

JSON response contains fields that present information about the process of the request and data.
apikey – user authentication key
success – status of the request
statsDuration – the time to process the request
requestNumber – the number of request by the user for the last 24 hours
Object – Data requested in the form of Entity Data Model (EDM) record

Search API

It can be used to query metadata records and media in the Europeana library. It can be used to search with keywords. More advanced queries are performed with Boolean searches and results advanced filtering. Finally, objects can be refined by their copyrights or source language.

Search API HTTPS requests have the following URL format:

https://www.europeana.eu/api/v2/search.json

There are a lot of parameters that could be passed to the request but most important is "**qf**" parameter. It contains the search query. It could contain simple keyword or more complex Boolean operators, Time ranges, Regular expressions etc.

Successful response contains also many metadata fields but the most important is "**items**" array which is a collection of searched objects.

2.2 NYPL Digital Collections API

The New York Public Library has a hundred years history of collecting unique material from different cultural heritage areas. Most of the Library's digitized collections are available through software service and APIs.

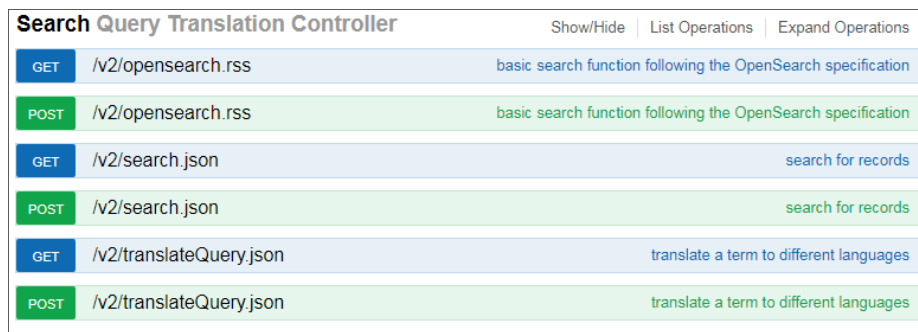To access the API we should register for free and then make request providing our username and password:

http://username:password@api.repo.nypl.org/api/v1/items/search?q=cats&publicDomainOnly=true

The response format could be JSON or XML. Default data format is JSON. In order to receive data in XML we should add '.xml' to the end of URL. Using "**page**" and "**per_page**" we can refine results pagination. "**PublicDomainOnly**" parameter limits results from the API to materials that have no copyright issues.

The result contains a collection of request objects. We can retrieve brief information for them using the fields "**title**", "**typeOfResource**", "**itemLink**", "**dateDigitized**". There are also other fields with more explicit information.

3 Software Implementations

There are a lot of predefined software libraries from APIs providers or from other developers that can be useful in order to automate usage of API services. Europeana provides a Swagger API console which can be used to overview of the Europeana Search & Record API. All types of searches and queries can be tested with it.

Fig. 1. Europeana console

On Figure 2 is shown a CH Context widget for WordPress that we use in a cultural heritage site. The widget is JavaScript based and is available for free. It uses jQuery selectors to define items or tags of currently displayed post in a website on which it is embedded. These items are used to query Europeana library for related objects. On the

website from the example we choose to show a preview of five of these objects and a link to Europeana search site for more results.
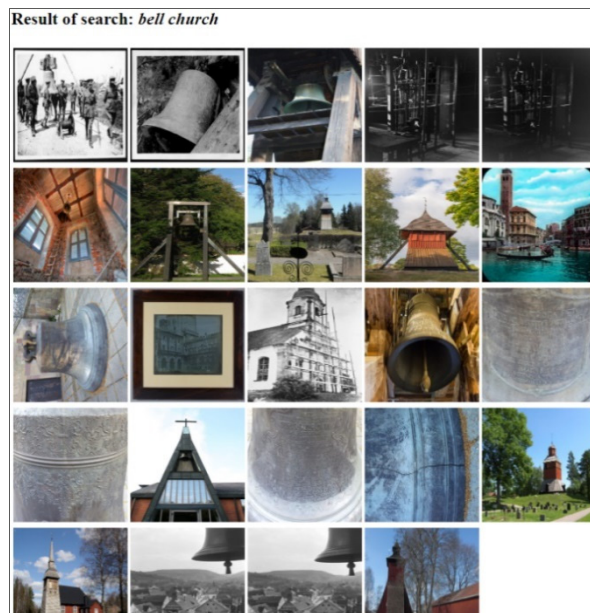

Fig. 2. CH Context widget

Fig. 3. Europeana search

We use Europeana PHP library and Laravel framework in order to request Europeana digital library and to visualize results. On Figure 3 is shown result of search using one of our template engines. On Figure 4 and 5 is shown part of PHP and jQuery code of our applications. We also test our templates on NYPL Digital Collections API.

```php
<?php
$apiKey = "api2demo";
$payload = new Colada\Europeana\Payload\SearchPayload();
$payload->addQuery("bell church");

try
{
    $client = new Guzzle\Client()
    $apiClient = new Colada\Europeana\Transport\ApiClient($apiKey, $client);

    $payloadResponse = $apiClient->send($payload);

    $items = $payloadResponse->getItems();
    foreach ($items as $item) {
        $item->getTitle();
        $item->getType();
    }

} catch (new Colada\Europeana\Exception\EuropeanaException $e) {

}
?>
```

Fig. 4. Europeana API usage PHP code

```javascript
$( document ).ready(function() {
    var apiCall = 'https://www.europeana.eu/api/v2/search.json?wskey=api2demo' +
    '&query=bell%20church&start=1&rows=24&profile=rich&theme=photography&thumbnail=true';

    $.getJSON(apiCall, function (json) {
        // here we will do something with the response
        var totalcount = json.totalResults;
        //var counthtml = '<h2>Total results: ' + totalcount + '</h2>';
        var resulthtml = '<h2>Result of search: <em>bell church </em></h2>';
        $('#content').append(resulthtml);
        var html = '<table> <tr>';
        var objecthtml = '';
        $('#content').append(html);
        $.each(json.items, function (i, item) {
            var title = item.title;
            var link = item.guid;
            var thumbnail = item.edmPreview;
            objecthtml += '<td>';
            objecthtml += '<a href="' + link + '" class="thumbnail" title="'
            + title + '" target="_blank"><img src="' + thumbnail + '" alt="'
            + title + '"></a> ';
            objecthtml += '</td>';
            if ((i+1) % 5 == 0){
                html += objecthtml;
                html += '</tr><tr>';
                objecthtml = '';
            }
        });
        if ((json.itemsCount % 5) > 0){
            html += objecthtml;
            html += '</tr><tr>';
        }
        html += ' </tr></table>';
        $('#content').append(html)
    });
});
```

Fig. 5. Europeana API usage jQuery code

4   Conclusions

In the paper is made and overview of some popular cultural heritage APIs. Presented services are only part of the rich collection of functions provided by these APIs. In the future we will continue to examine other services of these and other APIs.

Acknowledgements.

References

Bogdanova, G., Todorov, T., & Noev, N. (2018). Effective methods for organization and presentation of digitized cultural heritage. *International Journal of Mechanical Engineering and Technology (IJMET), Vol. 9(4)*, 991–1000.

Jacobson, D., Brail, G., & Woods, D. (2011). *APIs: A Strategy Guide.* O'Reilly Media.

Kremers, H. (2019). *Digital Cultural Heritage.* Springer.

Richardson, L., Amundsen, M., & Ruby, S. (2013). *RESTful Web APIs: Services for a Changing World.* O'Reilly Media.