TABLETECH *IATI Registry* Service

Juan García Fernández

Introduction

The *Tabletech IATI Registry* web service allows the client to obtain the data about the monetary aid sent to Sudan (code SD) or any other country of every year, in a period of 5 years back starting from the last year (included). Also, the service has to specify the provider organization names that raises these aids. The data is extracted from the IATI Registry (https://iatiregistry.org/) and the response is cached in aim to reduce the internal request to its API.

Usage

The web service is accessible through the following url:

```
http://test.binatree.com/tabletech/country/{recipient-country-code}
```

Mandatory parameters:

- *recipient-country-code*: The code of the country we want to obtain the data. The countries code list is available here: http://iatistandard.org/201/codelists/Region/

Request Example to obtain the Sudan country data:

```
http://test.binatree.com/tabletech/country/SD
```

The service response

In a correct request, the service returns the data in JSON format with the structure below:

```
{
     "year1" : {
       Provider-org-name1 : 1000
       Provider-org-name2 : 100
       ...
       Provider-org-nameN : 1
    },
     "year" : {
       Provider-org-name1 : 1000
       ...
       Provider-org-nameN : 1
    },
    ...
     "year5" : {
       Provider-org-name1 : 1000
       ...
       Provider-org-nameN : 1
    }
}
```

Juan García Fernández

If there is any error in the service, this is the error response :

```
{
     “error”: "We are working to solve an internal issue in our service. Please,
try later."
}
```

If there is no country specified in the request:

```
{
     “error”: "You must specify a country code. For instance: 'SD'."
}
```

Response details:

- The years and the figures of the provider organizations are ordered in descending mode.
- The figures of the provider organizations in the IATI Registry could be in different currencies. The Tabletech service converts all the figures to *USD* in the served response.

Response HTTP codes

HTTP Code	Description
200	The request is OK.
400	There is no country code.
500	There is an error/issue in the service.

Design Considerations

Assumptions and Dependencies

- The service can be used from an internet browser or used in a development environment.
- The service must work in any internet browser.

General Constraints

- The service must be hosted in a server with PHP 5.3 or higher version.
- The service must throw controlled errors.
- The service must respond with defined HTTP codes in aim to identify any issue.
- The service has to return JSON formatted responses.
- The service has to cache the different requests to improve the performance.

Goals and Guidelines

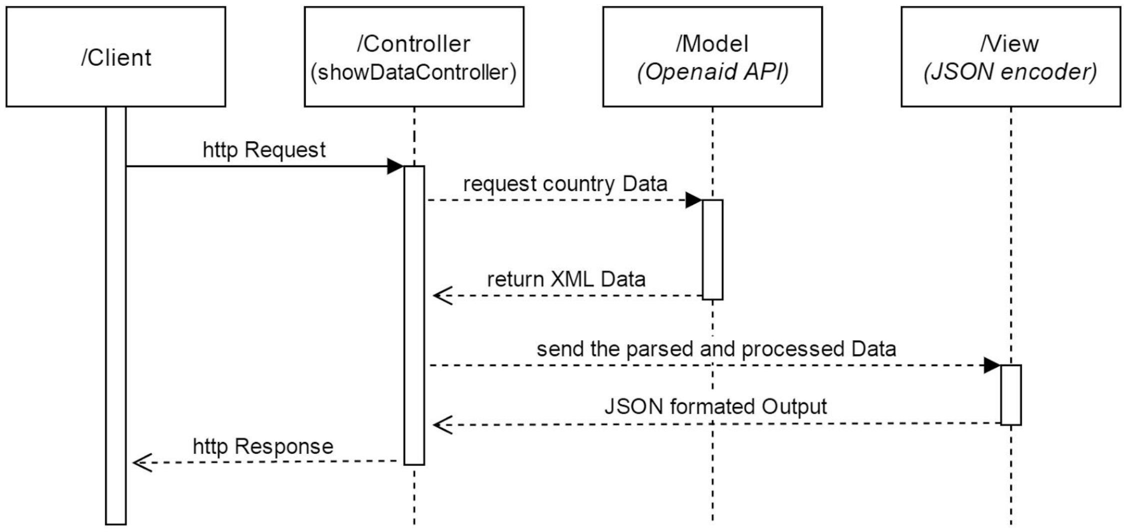
The service follows the *KISS* principle to improve performance. The service avoids using any PHP framework workflow.

Given the IATI Registry API XML response is not fast, the first goal of the service is to process this response faster. It uses Xqueries against the XML response.

Development Methods

The service development is based in the Model-view-controller (*MVC*) architectural pattern:

- *Model*: The IATI Registry API and cache system.
- *View*: We can consider the JSON encoder as a view.
- *Controller*: The service has one controller called “showData” that interacts with the model and manages the request and the response for the user.



Architectural Strategies

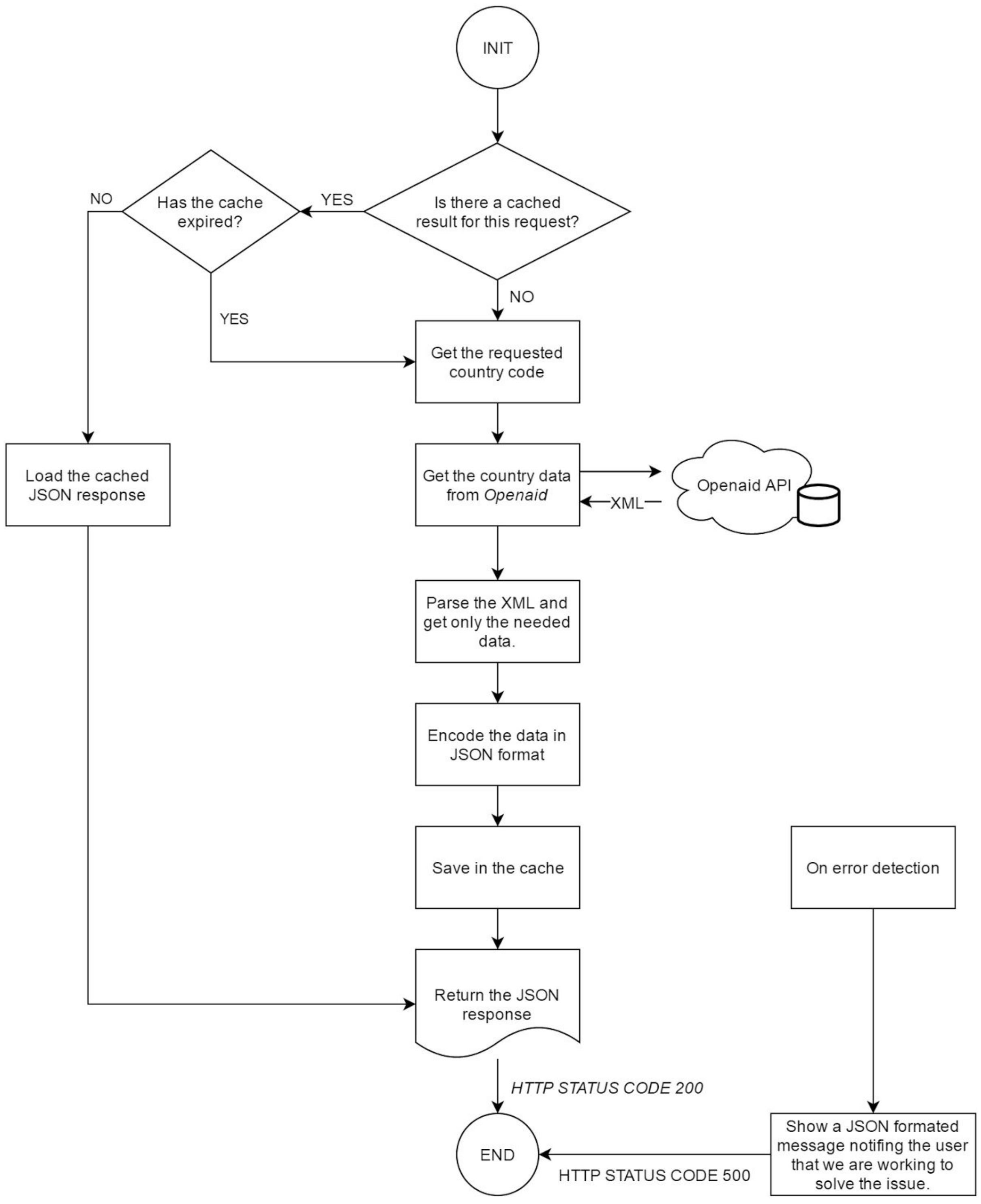
- Usage of PHP programming language under Apache or Nginx server.
- Usage of PHPUnit to implement unitary testing.
- Store the cache as files in the server.
- Use the IATI Registry API as datasource.

The service is a scalable development that allows the addition of new features easily. For instance:

- Allow the user to specify in what currency wants the response.
- Allow the user to specify the output response format. (XML, JSON, CSV).
- Allow the user to paginate the response.

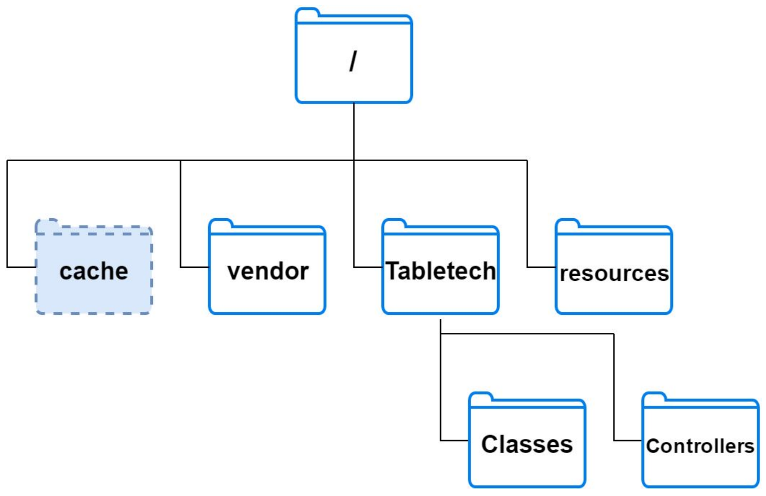
System Architecture

Workflow of a request:


INIT
Is there a cached result for this request?
YES
Has the cache expired?
NO
YES
NO
Get the requested country code
Load the cached JSON response
Get the country data from *Openaid*
Openaid API
XML
Parse the XML and get only the needed data.
Encode the data in JSON format
Save in the cache
On error detection
Return the JSON response
HTTP STATUS CODE 200
END
HTTP STATUS CODE 500
Show a JSON formated message notifing the user that we are working to solve the issue.

Juan García Fernández

Files structure:



Files in directories:

- root (/) : root directory
    - *app.php* : Service frontal that derives the request to the required controller.

- Tabletech : project bundle
    - Classes
        - *BaseController.php* : Base class to be extended by the Controller classes.
        - *CurrencyToUSD.php* : Class specialized in converting currencies to USD.
        - *OpenaidLoader.php* : Allows the communication with the IATI Registry API.
        - *Cache.php* : Class that allows the service to cache the response.

    - Controllers
        - *showDataController.php* : Controller class that requests the required country data and processes the response.

- resources : directory to store resources
    - Constants.php : Class with constants for static usage.

- vendor : directory for third parties code

- cache : directory is created by the cache system.

Policies and Tactics

- Every function / method must be documented in Javadoc notation.
- Git code repository. Usage of bitbucket.
- Testing the software with PHPUnit tests.
- Automatic deployment of the code using tools as Ansible.