

Date January2013

* Version 3.4

Title UI for API call  
Project 

3.4 Release

**This report intends to describe the way the APIs has been visualized in order to handle more accurately, easier and faster the url API hits. This implementation makes reference to the following doc: API\_Doc\_3\_4\_v05.docx for mGage LoyalMe+**

Contents

[Business Requirement 4](#_Toc344913126)

[UI 4](#_Toc344913127)

[UI Structure 5](#_Toc344913128)

[Implementation 5](#_Toc344913129)

# Business Requirement

*This reports as well as the implementation is for inside use in QA team.*

Calling APIs manually, is a time consuming and boring action. The tester often cannot focus on the input parameters and the response results easily in order to cover all possible test cases. Additionally the user needs to get the response and a quick link to a JavaScript Object Notation viewer.

A user starts the processing by entering the application. User has to put in the parameter blanks the values in accordance to the test cases, as these are discussed in the test plan in order to cover all possible cases. There is no limitation in the blank fields and the type of value the user can insert. This is the actual scope in order to investigate any type of error or exception in log file.

After inserting the appropriate values in the parameters’ fields (API parameters and IP, port), the user can display URL in order to check any individual part of the URL. The user can also submit these parameters as if submitting the respective URL and this leads him automatically in the response tab.

The user is able to reset the values with reset button or keep some of the values such as *msisdn* and *change code/password*. Additionally with the LoadJSon button the user can acquire with copy-paste action, the response content and use the online JSon viewer.

## UI

The UI will be used by QA Loyalty team. In this team, the members who participate may be

* The ones who in the past have used just browsers and URLs in order to test the correctness of API calls.
  1. Notify the bug and suggest corrections in the url structure
  2. Compare the expected response with the one displayed un the tool
  3. Show URL automatically and use it for reporting reasons
  4. Use this URL probably in other testing tools
* The ones who have just entered the team and they do not have the appropriate technical background in order to understand the structure of the url. This means that with a brief description of the functionality of the present tool they will be able to
  1. Notify the bug
  2. Compare the expected response with the one displayed in the tool
  3. Show the URL automatically

As a result, taking into consideration the scope of this tool, 3 main parts are identified in the tool.

1. The part which the user-tester inserts the obligatory and optional parameters
2. The part in which any relevant reference to the respective URL is made
3. The part in which the relevant response is displayed

Considering the URL structure and the QA needs there is the following template

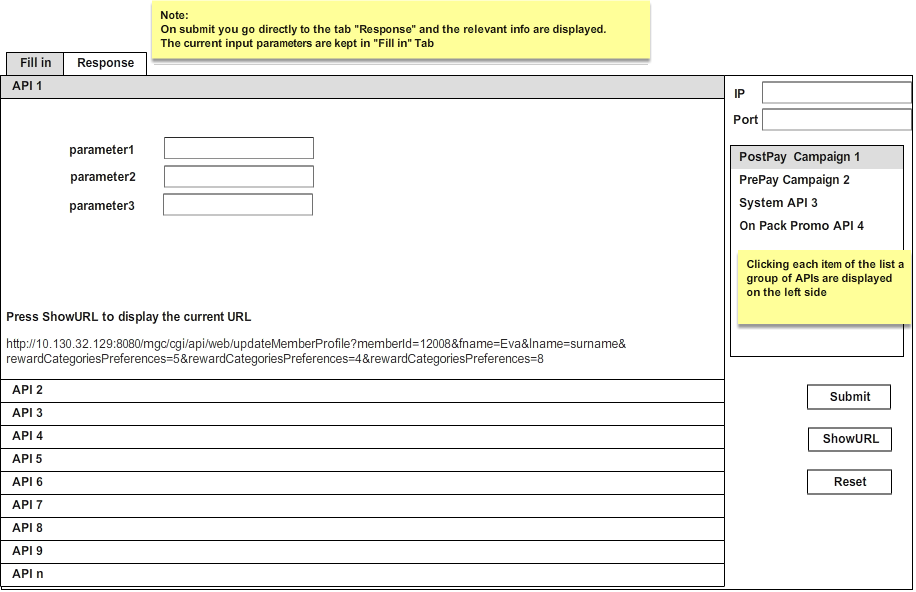
http://IP:Port/APIcall?parameter1=&parameter2=&parameter3=&parameter4=&parameter=

Depending on the variables filled in for each API in the specific campaign the mentioned above URL is the one displayed in the field where the URL that is about to submit is displayed.

This URL can be submitted, displayed or reset, respectively to each field of the form changed each time. When the URL is submitted the response tab is displaying the response of the API call automatically.

The following section describes the UI and the functionality of each button.

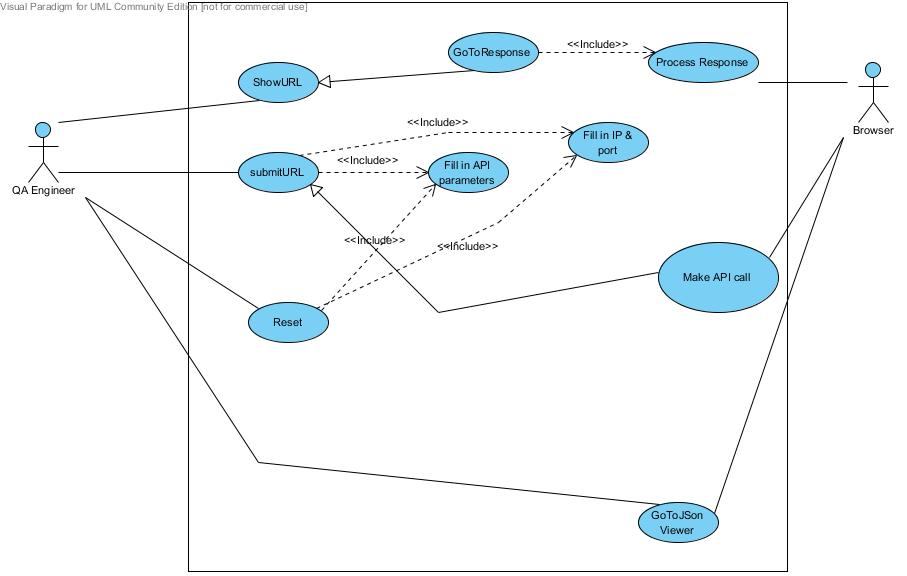
### UI Structure

The following picture displays a wireframe of the UI

Wireframe Figure

## Implementation

### UML Design



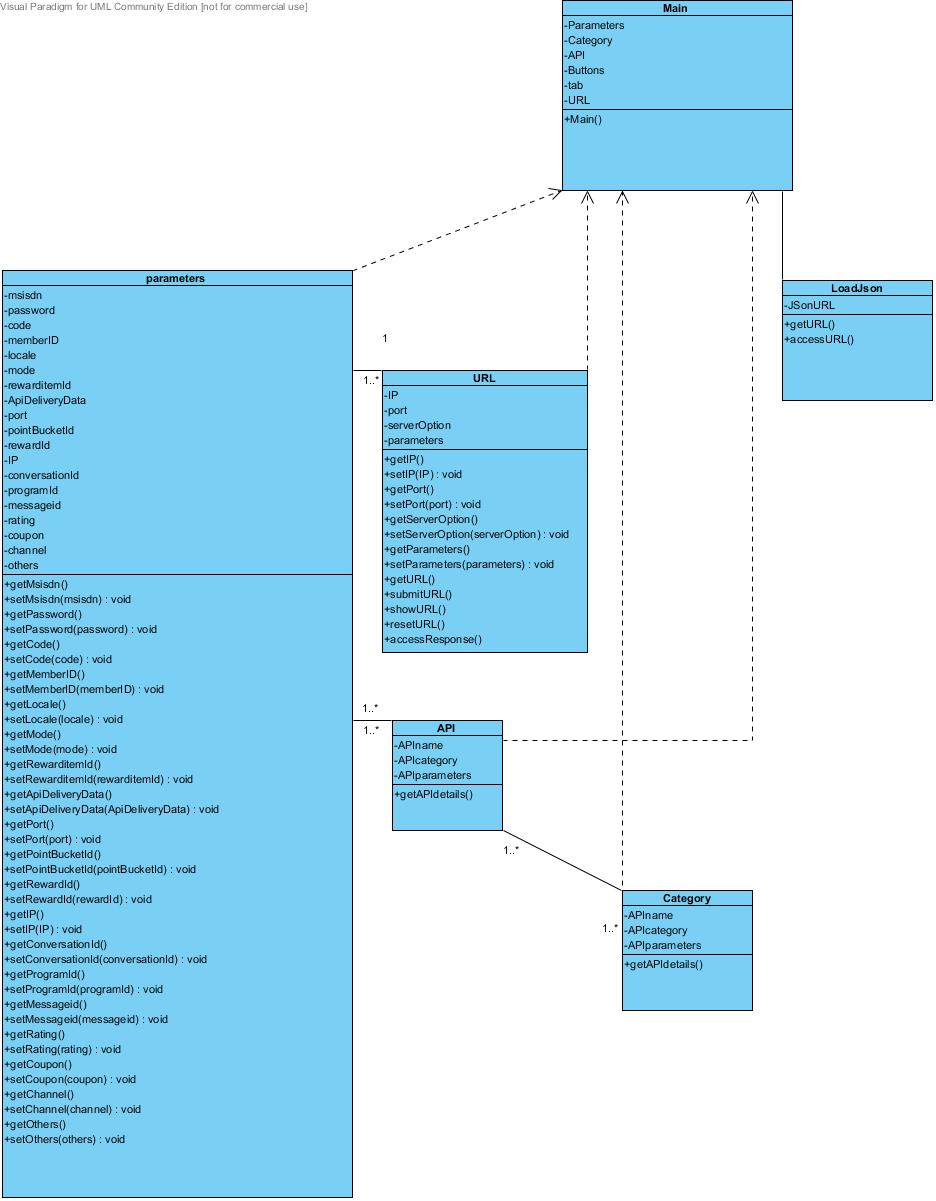
The Use case diagram of the online Easy API processing system contains the scope of the system, the list of system functionalities and the total vision of the possible interactions between the system and the actors of this system. The main actors identifying here are:

* *QA Engineer:* The main user of the EasyAPI, primary actor of the system.
* *Browser:* The Browser used in order to run the tests via the EasyAPI

Based on the above use case diagram we can identify the following groups of main types of activities:

* QA Eng. starts inserting the values for each parameter. The system asks for IP and port, 2 values QA Eng. should have in mind before using EasyAPI.
* QA Eng. displays the URL and checks if the parameters are as expected
* QA Eng. after placing the right parameters submits the data and automatically goes to response tab.
* QA Eng. copies the content of the response tab and pastes it to the online JSon viewer.
* The Browser processes the respective responses in each action calling the relevant APIs

### UML classes



### Environment

IDE and tools

**Eclipse and Aptana:** Eclipse for Testers Version: Juno Service Release 1 as an Integrated Development Environment (IDE) to develop the application & Aptana Studio 3 for Web Project.

**Visual Paradigm for UML Version 10**: Visual Paradigm has been used for the design and UML10.0 notation.

## Prototypes of the System (Graphical User Interface)