**Installation Guide**

**CITADEL…on the move**

**Mobile Application Templates: ‘Environmental Data’**

**Project co-funded by the European Commission within the ICT Policy Support Programme**

Table of Contents

[CITADEL…on the move 1](#_Toc358039408)

[1. CITADEL Mobile Application template 3](#_Toc358039409)

[1.1 Structure of a template 3](#_Toc358039410)

[1.2 Prerequisites 4](#_Toc358039411)

[1.3 Deploying a template in 6 steps 5](#_Toc358039412)

[1.4 Settings of the application 6](#_Toc358039413)

[1.5 Changing the default sensor feeds coming via the Xively API 7](#_Toc358039414)

# CITADEL Mobile Application template

## Structure of a template

A template in the context of the CITADEL project is actually a package containing all the necessary files that will allow a citizen developer to easily deploy it and afterwards extend it. The technology stack of a template consists of the following widely used technologies:

* HTML5
* Javascript
* PHP
* MySql
* JSON

It should be also noted that the jQuery Mobile javascript framework is also used in the templates. There has been an effort to limit the use of the framework for visualization purposes and for the navigation between the pages only. Data retrieval and manipulation have been implemented using simple javascript functions.

Inside the template all files are organised under the following folder structure:

1. js – contains all javascript files
2. css – contains all css files
3. images – contains the images used
4. doc – contains the installation and programming guides

The file serving as the home page of the application is “index.php” and is located outside of all the above folders and inside the root folder of the template. File “Config.php” includes the settings for the application as presented in Section 1.5. The “dataset.php” file contains the code responsible to load the dataset from the .json file or the database.

The folder contents are depicted in Figure 1.

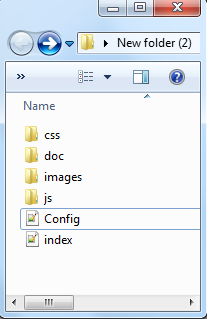


Figure 1

## Prerequisites

Running the mobile application templates requires a minimum of software elements installed to your machine. A web server and PHP support are needed in order to be able to deploy the template using the 6 steps described in Section 1.3.

1. **Web server**  
   In order to serve the generated pages to your browser, the template requires some web server software. The most common choice would be the [Apache](http://httpd.apache.org/) (<http://httpd.apache.org/>) web server, available at [the official download page](http://httpd.apache.org/download.cgi) (<http://httpd.apache.org/download.cgi>).
2. **PHP support**  
   [PHP](http://www.php.net/) (<http://www.php.net/>) is the programming language in which the template is written and is required to run the template application. It can be downloaded at [the official download page](http://www.php.net/downloads.php) (<http://www.php.net/downloads.php>). You should choose the download that is appropriate for the operating system of your machine, e.g. for windows the easiest way is to download the installer.

The above technology stack is also called a WAMP (**W**indows **A**pache **M**ySQL **P**HP) environment (LAMP in linux). There are a number of software packages that perform the installation and configuration of this environment in one place, without having to download and set up the different elements separately. Two of the most popular are [XAMP](http://www.apachefriends.org/en/xampp.html) (<http://www.apachefriends.org/en/xampp.html>) and [WAMPServer](http://www.wampserver.com/en/) (<http://www.wampserver.com/en/>).

## Deploying a template in 6 steps

The fastest way to deploy a template and view it in the web browser of a desktop or a mobile device is described in the following steps:

|  |
| --- |
| Deployment in 6 STEPS |
| 1. Download the template **citadel-environmental-template.zip** |
| 1. Unzip the contents in a file named **citadel- environmental -template.** This file should contain all the folder structure described in the previous paragraph |
| 1. Put the **citadel- environmental -template** folder inside the web directory of your local webserver. For instance, if you are using wamp server, go to www directory (see figure 2 below) and save the folder there. |
| 1. Open the **Config.php** file and fill in your root web directory. For instance, if you used the wampServer default setup you should have:   define(“HTDOCS\_ROOT”, “C:/wamp/www/”) |
| 1. Open a browser and point it at:   [http://localhost/**citadel -** **environmental** **-template**/index.php](http://localhost/citadel%20-crowd-sourcing%20-template/index.php) |
| 1. The template application is up and running with the default sensor feeds coming with ajax rest calls via the Xively Api. |

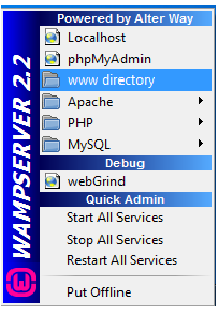


Figure 2

**Things to notice:**

* You should not change the folder names or the folder structure described in the previous steps in order to successfully deploy the application.
* If you want to also **access the application from a mobile device** connected to the same network as the desktop machine, then you should modify the url given in step 5 as follows:
  + http://your-current-IP[[1]](#footnote-1)/**citadel-** **environmental -template**/index.php.

This change requires an update of the parameter SERVERNAME in the Config.php file.

* There are a number of browser plugins allowing you to see how the application looks when accessed by different-sized devices (e.g. Mobile Resizer for Chrome). Note that the looks might eventually be slightly different in real mobile devices.

## Settings of the application

The list of settings that can be found in the “Config.php” file and their meaning are presented in the table below:

Table 1: Mobile Application Settings

|  |  |  |
| --- | --- | --- |
| setting | Default value | meaning |
| SERVERNAME | localhost | The name of the local web server instance. It should be changed to the IP address in order to be able to access the template from another device of the same network. |
| HTDOCS\_ROOT |  | The root directory of the web server. There is no standard path for this so it should be changed to match the current root directory. |
| BASE\_DIR | citadel-environmental-template/ | This is the root folder of the application. This should not be changed in no backend is used. |
| CLASSES\_DIR | php | This is the folder containing the php files. This should be not changed if the default folder structure is followed. |
| MAP\_CENTER\_LATITUDE | 53.483526 | The latitude of the centre of Manchester. Manchester is one of the two pilot cities that provide environmental sensor data. |
| MAP\_CENTER\_LONGITUDE | -2.23065399999996 | The longitude of the centre of Manchester. Manchester is one of the two pilot cities that provide environmental sensor data. |
| MAP\_ZOOM | 6 | The initial zoom level of the google map. |

## Setting the sensor feeds

The environmental mobile application template displays data coming from Xively feeds. Xively ([www.xively.com](http://www.xively.com)). Xively is a Platform as a Service (PaaS) for the Internet of Things which simplifies the interconnection of devices, data, people and places.

The Xively API[[2]](#footnote-2) presents a collection of Restful resources. The Xively API makes it easy for devices, applications and services to read and write data to Xively and through Xively to each other. Xively supports reading and writing data via three resources: **Feeds**, **Datastreams** and **Datapoints**. The template mainly uses the first two, Feeds and Datastreams. A feed is a data representation of an environment and its datastreams. Metadata associated with feeds allow for the specification of whether a feed is fixed or mobile, indoor or outdoor, etc. Datastreams are required to have unique alphanumeric IDs within the environment. Datastreams can also specify units (celsius for temperature, for example) as well as tags.

The API call https://api.xively.com/v2/feeds/ returns all the public sensor feeds, containing the current state of all datastreams and metadata. This way, you can find all feeds you may be interested in (see the example below). Every feed is represented by a feed id (e.g. **"id":** 336217921**,**). The template loads the data coming from the feeds configured in an array with feed ids inside the **“index.php”**. If you add an extra feed id e.g. “336217921” into this array, you can watch the application running with an extra sensor displayed on the map.

The following code snippet from index.php show the array of feeds that has to be updated with the desired feed ids.

/\* List of feeds read from json object \*/

var feeds = new Array("36075", "125493", "125492", "125488", "125490");

1. In Windows you can find the current IP by opening a command window (start-> type ‘cmd’ in Search programs) and typing ‘**ipconfig**’ [↑](#footnote-ref-1)
2. <https://xively.com/dev/docs> [↑](#footnote-ref-2)