WMarket Documentation

Release develop

March 03, 2016

WMarket is the reference implementation of the FIWARE Marketplace Generic Enabler. The Marketplace provides functionality necessary for bringing together offering and demand for making business. These functions include basic services for registering business entities, publishing and retrieving offerings and demands, search and discover offerings according to specific consumer requirements as well as lateral functions like review, rating and recommendation. Besides the core functions, the Marketplace may offer value because of its "knowledge" about the market in terms of market intelligence services, pricing support, advertising, information subscription and more.

This project is part of FIWARE. Check it out in the Catalogue!

CHAPTER 1

Index

Installation Guide The guide for WMarket maintainers that explains how to install it.

Administration Guide The guide for adminstrators where that explains how a user can become an offering provider.

Building from Sources The guide for developers that explains how to create your own WAR based on the source code. This section also explains how to you can execute the existing tests.

User and Programmer Guide The guide for people who will be using a WMarket site through its web interface, explains how to use the web interface and its features.

1.1 Installation Guide

The purpose of this document is to describe how to install and administrate the necessary software on a server so that it can run WMarket.

WMarket itself is a Java Web Application, packaged in a WAR file and relies on a SQL database.

1.1.1 System Requirements

This section covers the requirements needed to install and use WMarket.

Hardware Requirements

The following table contains the minimum resource requirements for running WMarket:

Resource	Requirement
CPU	1-2 cores with at leas 2.4 GHz
Physical RAM	2GB-4GB
Disk Space	10GB The actual disk space will depend on the amount of data being stored within WMarket

Operating System Support

WMarket has been tested against the following Operating Systems:

- Ubuntu 14.04 LTS
- CentOS 7

However, you can install WMarket in any machine where you can install an Application Server (such as Tomcat) and a MySQL server.

Note: This Installation Guide describes the installation process on a Linux based System.

Software Requirements

In order to have WMarket running, the following software is needed:

- MySQL Server 5.5
- Java 1.7.x
- Application Server, Apache Tomcat 7.x
- WMarket itself

1.1.2 Software Installation

Before you can use WMarket on your own machine(s), you need to install it. There are three ways to install WMarket:

1. *With the Installation Script*
2. *With a Docker Image*
3. *Manual*

Installing WMarket using scripts

In order to ease the WMarket installation, the script `install.sh` can be used. This script downloads the latest stable version of WMarket (from GitHub) if the file *WMarket.war* does not exist, install all the needed dependencies, configures the instance (database and preferences) and deploys it in the Tomcat Server. You can find this script here.

Once that you have downloaded the script, you can easily run it by executing the following command:

```
$ sudo ./install.sh
```

Note: If you use this script you can skip the configuration section; however, it is highly recommended to read it in order to understand the different configuration parameters.

During the execution of the script you will be prompted several times in order to retrieve some information. Specifically:

- **Database Installation**: in Debian/Ubuntu systems the MySQL installer automatically ask a password to create the root user. In CentOS systems the the script calls the MySQL configuration command in order to create this password so a message similar to the following one will be shown:

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current
password for the root user.  If you've just installed MySQL, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.
```

```
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

Set root password? [Y/n] y
New password: *******
Re-enter new password: *******
Password updated successfully!
Reloading privilege tables..
 ... Success!


By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MySQL comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't exist
 ... Failed!  Not critical, keep moving...
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!




All done!  If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!


Cleaning up...
```

- **Database Configuration**: the script creates a database called `marketplace`. To perform this action, MySQL credentials with administrative permission are required. For this reason, you will see the following lines:

```
> About to create 'marketplace' database. Please, provide MySQL credentials with administrative right
>> User: root
>> Password: *******
```

- **Index path**: WMarket uses Lucene indexes to provide better search results. These indexes have to be stored in files so the script will ask you for a folder to store them.

```
> Path to store indexes: [FOLDER_TO_STORE_INDEXES]
```

- **Media files**: WMarket users will be able to attach images to stores in order to improve the navigation. These images have to be saved in a folder of the file system used by the WMarket instance. For this reason, the script will ask you for a path to store these images and their maximum size:

```
> Path to store media files: [FOLDER_TO_STORE_MEDIA]
> Max size for media files (in bytes): [MAX_FILE_SIZE]
```

- **Description Autoupdate**: uploaded descriptions are regularly checked to verify if new offerings have been included. The script will ask you the period that have to be used to check these descriptions.

```
> Period to update descriptions (in seconds): [PERIOD_TO_UPDATE_DESCRIPTIONS]
```

- **Authentication Configuration**: WMarket can work with a local authentication system or using an external IdM. The script will ask you what authentication system you prefer. If you opt for an external IdM, some configuration details will be required as can be seen in the following example:

```
> Do you want to use OAuth2 to authenticate users? (Y/n): Y
>> OAuth2 Server: [FIWARE_IDM_URL]
>> OAuth2 Key: [OAUTH2_KEY]
>> OAuth2 Secret: [OAUTH2_SECRET]
>> Provider Role: [OAUTH2_PROVIDER_ROLE]
>> WMarket External IP: [MACHINE_IP]
```

This script will directly configure some preferences with default values (see the *Configuration* section for more info of the described preferences). Concretely:

- A database called `marketplace` is created.
- Database connection is configured with the administrative credentials you have provided.
- Two new folders are created: one for index files and another for media files. The permissions of these folders are changed so Tomcat can access them.
- Authentication system is configured according to your preferences.
- The Marketplace is deployed in Tomcat as `WMarket`.

Installing WMarket using Docker

Stating on version 4.3.3, you are able to run WMarket with Docker. As you may know, WMarket needs a MySQL database to store some information. For this reason, you must create an additional container to run the database. You can do it automatically with `docker-compose` or manually by following the given steps.

The WMarket image is available on Docker Hub.

Using docker-compose

You can install WMarket automatically if you have `docker-compose` installed in your machine. To do so, you must create a folder to place a new file called `docker-compose.yml` that should include the following content:

```
wmarket_db:
    image: mysql:latest
    volumes:
         - /var/lib/mysql
    environment:
        - MYSQL_ROOT_PASSWORD=my-secret-pw
        - MYSQL_DATABASE=marketplace

wmarket:
    image: conwetlab/wmarket
    volumes:
        - /WMarket
    ports:
        - "80:8080"
    links:
        - wmarket_db
    command: bash -c 'sleep 15 && catalina.sh run'
```

Once that you have created the file, run the following command:

```
docker-compose up
```

Then, WMarket should be up and running in `http://YOUR_HOST:80/WMarket` replacing `YOUR_HOST` by the host of your machine.

Without docker-compose

1) Creating a Container to host the Database The first thing that you have to do is to create a docker container that will host the database used by WMarket. To do so, you can execute the following command:

```
docker run --name wmarket_db -e MYSQL_ROOT_PASSWORD=my-secret-pw -e MYSQL_DATABASE=marketplace -v /va
```

- As can be seen, some environment variables are set in this command to set up the data base. You must **not** change these variables, since their values are the ones expected by the WMarket image.

2) Deploying the WMarket Image Once that the database is configured, you can deploy the image by running the following command (*replace* `PORT` *by the port of your local machine that will be used to access the service*):

```
docker run --name wmarket -v /WMarket -p PORT:8080 --link wmarket_db -d conwetlab/wmarket
```

Once that you have run these commands, WMarket should be up and running in `http://YOUR_HOST:PORT/WMarket` replacing `YOUR_HOST` by the host of your machine and `PORT` by the port selected in the previous step.

Manually installing WMarket

All the mandatory dependencies can be easily installed on a Debian based Linux distribution using `apt-get`:

Ubuntu/Debian:

```
sudo apt-get install mysql-server mysql-client
sudo apt-get install openjdk-7-jdk
sudo apt-get install tomcat7 tomcat7-docs tomcat7-admin
```

CentOS/RedHat:

In a CentOS/RedHat system, all the required dependencies can be installed using `yum`:

```
# Install MySQL
sudo rpm -Uvh http://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm
sudo yum -y install mysql-community-server

# Install Java
sudo yum -y install java-1.7.0-openjdk-devel

# Install Tomcat 7
sudo yum -y install tomcat tomcat-webapps tomcat-admin-webapps
```

Configuration

This section explains how to create WMarket database and how to configure the different preferences. Note that if you have used the provided script you can just skip this section. However, it is highly recommended to read it in order to understand the different settings.

Database Configuration WMarket uses an internal database named `marketplace` that need to be created in MySQL. To create it you need to have administrator permissions in MySQL. This usually means that you have to use the MySQL root user with the password you chose during the installation process.

Ubuntu/Debian

```
sudo service mysql start
mysqladmin -u root -p[MYSQL_ROOT_PWD] create marketplace
```

CentOS/RedHat:

```
sudo systemctl start mysqld
mysqladmin -u root -p[MYSQL_ROOT_PWD] create marketplace
```

Note: In CentOS, MySQL is installed without requiring any password for the root user. The default password is empty. If you want to set up a password for the root user, you can run the `/usr/bin/mysql_secure_installation` script.

WMarket Configuration Before deploying the provider JAR into your Application Server, you must configure some parameters.

Database Before you deploy WMarket, you have to set up the database. To do so, you have to edit the file `WMarket.war/WEB-INF/classes/properties/database.properties` and complete the fields as follows:

```
jdbc.driverClassName=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://[MYSQL_SERVER]:[MYSQL_PORT]/marketplace
jdbc.username=[YOUR_DB_USER]
jdbc.password=[YOUR_DB_PASSWORD]
```

Index WMarket uses index files to provide better search results. These files must be stored in some folder of your file system. You can specify this folder by editing the property `lucene.IndexPath` included in the file `WMarket.war/WEB-INF/classes/properties/marketplace.properties`.

```
# Lucene
lucene.IndexPath=[FOLDER_TO_STORE_INDEX]
```

Note: Ensure that tomcat user can read and write new files in this directory.

Media Files Under certain circumstances, users are allowed to upload media files (images) in order to ease the process of identifying assets. You can edit the file `WMarket.war/WEB-INF/classes/properties/marketplace.properties` to set where these files are stored and their maximum size:

```
# Static files
media.folder=[FOLDER_TO_STORE_MEDIA_FILES]
media.maxSize=[MAX_FILE_SIZE_IN_BYTES]
```

Note: Ensure that tomcat user can read and write new files in this directory.

Descriptions Autoupdate Descriptions are the files used to define the offerings that will be available in WMarket. These files are parsed when they are uploaded to WMarket but when they are updated, there is no way to automatically reflect these changes. For this reason, you can set an interval to update all the descriptions and the offerings that they contain. To do it, set the preference `descriptions.updatePeriod` (included in the file `WMarket.war/WEB-INF/classes/properties/marketplace.properties`) with the period that should be used to update the descriptions. The value must be written in seconds.

```
# Update rate
descriptions.updatePeriod=[PERIOD_TO_UPDATE_DESCRIPTIONS_IN_SECONDS]
```

OAuth2 WMarket uses a local authentication system by default. However, the software is ready to work with the FIWARE Identity Manager.

The first thing that you have to do is to create a new application in the FIWARE IdM. To do so, go to https://account.lab.fiware.org/idm/myApplications/create/ and complete all the required fields:

- **Name**: *You can choose any name*
- **Description**: *You can write any description*
- **URL**: `http://[WMARKET_HOST]:[WMARKET_PORT]`
- **Callback URL**: `http://[WMARKET_HOST]:[WMARKET_PORT]/WMarket/callback?client_name=FIWARECli`
- **Roles**: In order to allow some users to create stores and descriptions with the user interface, you must create a new role. You can choose any name for this role.

Once that you have created the application in the FIWARE IdM, you must edit the following configuration files:

1. `WMarket.war/WEB-INF/classes/spring/config/BeanLocations.xml`: replace `<import resource="security.xml" />` by `<import resource="securityOAuth2.xml" />`.
2. `WMarket.war/WEB-INF/classes/properties/marketplace.properties`: set up your OAuth2 following the next template:

```
# OAuth2
oauth2.server=[OAUTH2_SERVER]
oauth2.key=[OAUTH2_KEY]
oauth2.secret=[OAUTH2_SECRET]
oauth2.callbackURL=http://[WMARKET_SERVER]:[WMARKET_PORT]/WMarket/callback
oauth2.signOutPath=auth/logout
oauth2.providerRole=[OFFERING_PROVIDER_ROLE]
```

WMarket Deployment

WMarket can now be installed by copying the WAR file into the `webapps` folder of Apache Tomcat. If you have installed Tomcat using the package manager, the `webapps` folder should be located at:

- In Ubuntu/Debian: `/var/lib/tomcat7/webapps`
- In CentOS/Redhat: `/usr/share/tomcat/webapps`

To install WMarket on other Java Application Servers (e.g. JBoss), please refer to their specific application server guidelines.

Once that you have copied the WAR file into the `webapps` folder, you can start Tomcat. The way to do it depends on your operating system.

Ubuntu/Debian:

```
sudo service tomcat7 start
```

CentOS/Redhat:

```
sudo systemctl start tomcat
```

1.1.3 Sanity check procedures

The Sanity Check Procedures are those activities that a System Administrator has to perform to verify that an installation is ready to be tested. Therefore there is a preliminary set of tests to ensure that obvious or basic malfunctioning is fixed before proceeding to unit tests, integration tests and user validation.

End to End testing

Although one End to End testing must be associated to the Integration Test, we can show here a quick testing to check that everything is up and running. The first test step involves registering a new user. The second test step tests if it is possible to authenticate against the WMarket.

Step 1: Registering a new user

Note: If you have chosen an external authentication system (i.e. FIWARE IdM), you can avoid this step.

Go to `http://[WMARKET_HOST]:[WMARKET_PORT]/WMarket/register` and complete all the fields appropriately. After clicking “Create”, a confirmation message should inform you that the user has been created correctly. If you receive an error check that you have complete all the fields in an appropriate way and that you have follow all the instructions given in this guide.

Step 2: Authenticating your user and becoming a provider

Go to http://[WMARKET_HOST]:[WMARKET_PORT]/WMarket and introduce the authentication details according to the user that you have created in the previous step. Then, become a provider as explained in the Administration Guide.

Step 3: Creating a Store

Once that you are provider, you should check that you are able to create a new Store. To do so, follow the steps provided *here*. You will be required to introduce the Store details such as the name and the URL. You can introduce any name and any valid URL.

Step 4: Registering a Description

Finally, you should check that you are able to register a description. Just follow the steps provided *here*. You will be asked for a name and an URL. Please, note that the URL must be a link to a valid USDL. If you do not have a valid USDL hosted in a HTTP Server, you can use the following URL:

```
http://wmarket.readthedocs.org/en/latest/_downloads/example.rdf
```

If you don't obtain any error, the WMarket is correctly deployed. Congratulations!!

List of Running Processes

You can execute the command ps -ax | grep 'tomcat\|mysql' to check that the Tomcat web server and the MySQL database are running. It should show a message text similar to the following:

```
846 ?        Sl     60:40 /usr/bin/java -Djava.util.logging.config.file=/root/tomcat8/conf/logging.pro
911 ?        Ssl    17:24 /usr/sbin/mysqld
```

Network interfaces Up & Open

To check whether the ports in use are listening, execute the command netstat -ntpl. The expected results must be somehow similar to the following:

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      911/mysqld
tcp6       0      0 :::443                  :::*                    LISTEN      846/java
tcp6       0      0 :::8009                 :::*                    LISTEN      846/java
tcp6       0      0 :::80                   :::*                    LISTEN      846/java
```

Databases

The last step in the sanity check (once that we have identified the processes and ports) is to check the database that has to be up and accept queries. For that, we execute the following commands:

```
$ mysql -u [DB_USER] -p[DB_PASS] marketplace
> show tables;
```

It should show a message text similar to the following:

```
+------------------------+
| Tables_in_marketplace |
+------------------------+
| bookmarks              |
| categories             |
| descriptions           |
| last_viewed            |
| offerings              |
| offerings_categories  |
| offerings_services    |
| price_components      |
| price_plans           |
| reviewable_entity     |
| reviews               |
| services              |
| services_categories   |
| stores                |
| users                 |
+-----------------------+
15 rows in set (0.00 sec)
```

1.1.4 Diagnosis Procedures

The Diagnosis Procedures are the first steps that a System Administrator has to take to locate the source of an error in a GE. Once the nature of the error is identified by these tests, the system admin can resort to more concrete and specific testing to pinpoint the exact point of error and a possible solution.

Resource availability

The resource load of the WMarket strongly depends on the number of concurrent requests received as well as on the free main memory and disk space:

- Mimimum available main memory: 256 MB
- Mimimum available hard disk space: 2 GB

Resource consumption

Resource consumption strongly depends on the load, especially on the number of concurrent requests.

- The main memory consumption of the Tomcat application server should be between 48MB and 1024MB. These numbers can vary significantly if you use a different application server.

I/O flows

The only expected I/O flow is of type HTTP or HTTPS, on ports defined in Apache Tomcat configuration files, inbound and outbound. Requests interactivity should be low.

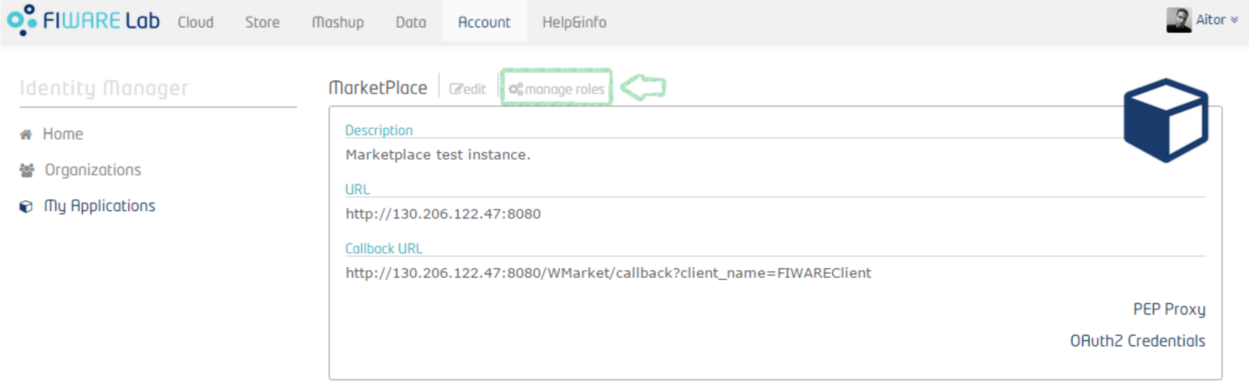
1.2 Administration Guide

The WMarket Graphical User Interface (GUI) can be very complex for consumers (those users who do not create stores or description/offerings). For this reason, options to create stores and descriptions are hidden by default.

The process to become a provider is very easy and depends on the selected authentication system.

1.2.1 External Authentication (i.e. IdM)

If your WMarket instance is using an IdM (i.e. FIWARE) to authenticate users, you have to create a role and assign this role to those users who are interested in publishing descriptions/offerings. To do so, the first thing that you have to do is to access the IdM. Then, identify the WMarket application among all the available applications. You will be redirected to the application management page where you have to click the **Manage Roles** button.

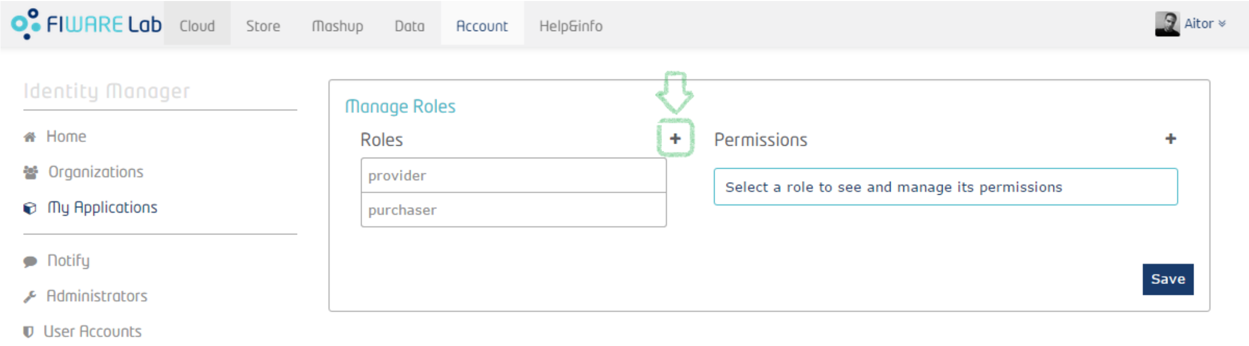


Description
Marketplace test instance.

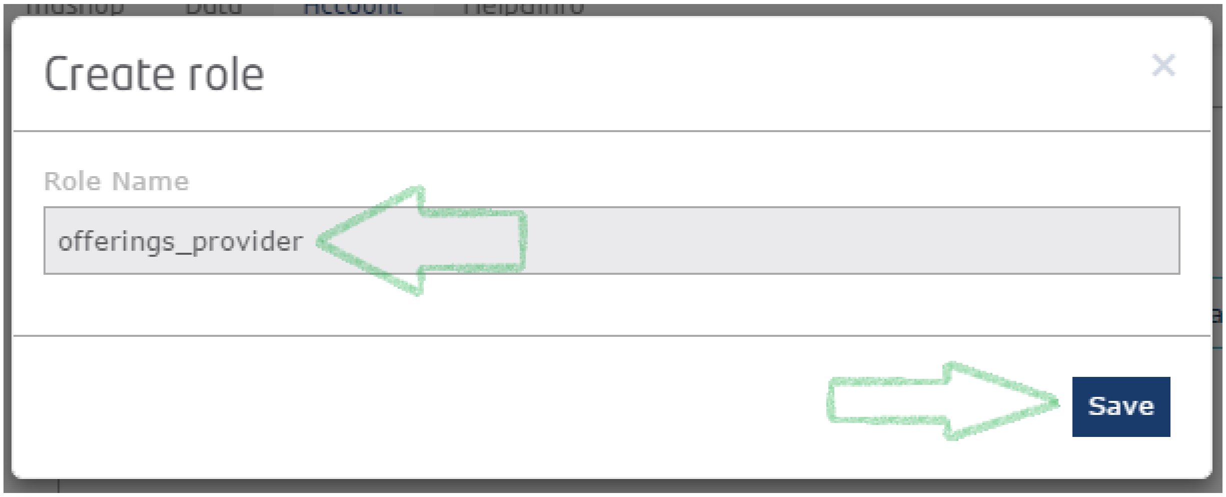
URL
http://130.206.122.47:8080

Callback URL
http://130.206.122.47:8080/WMarket/callback?client_name=FIWAREClient

Once that you are in the page to manage roles, you have to create a new one. To do so, click the **+** button as shown in the following image.



Then, type the name of the new role. You can choose any name.

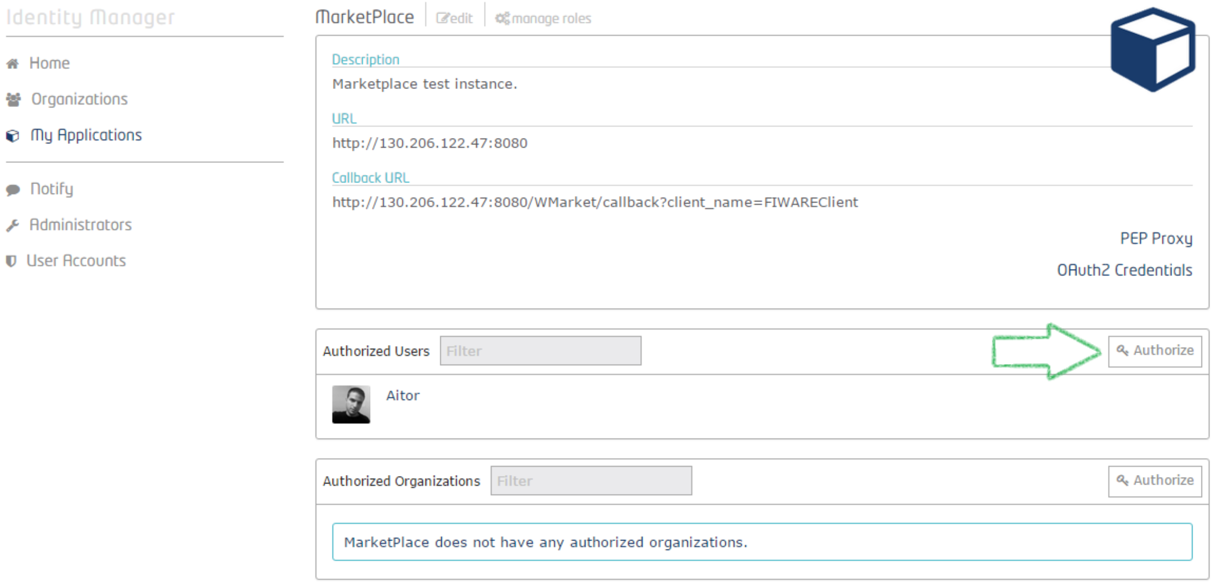


Role Name
offerings_provider

When the new role is created, you have to modify the WMarket preferences so it can understand which users are providers. To do so, you have to edit the file

`WMarket.war/WEB-INF/classes/properties/marketplace.properties` and modify the `oauth2.providerRole` preference, where you have to set the name of the role that you have created in the last step.

Finally, you have to assign the role to those users who want to become providers. To achieve this, access the IdM and click on the WMarket application. Then, click the **Authorize** button as shown in the following image.



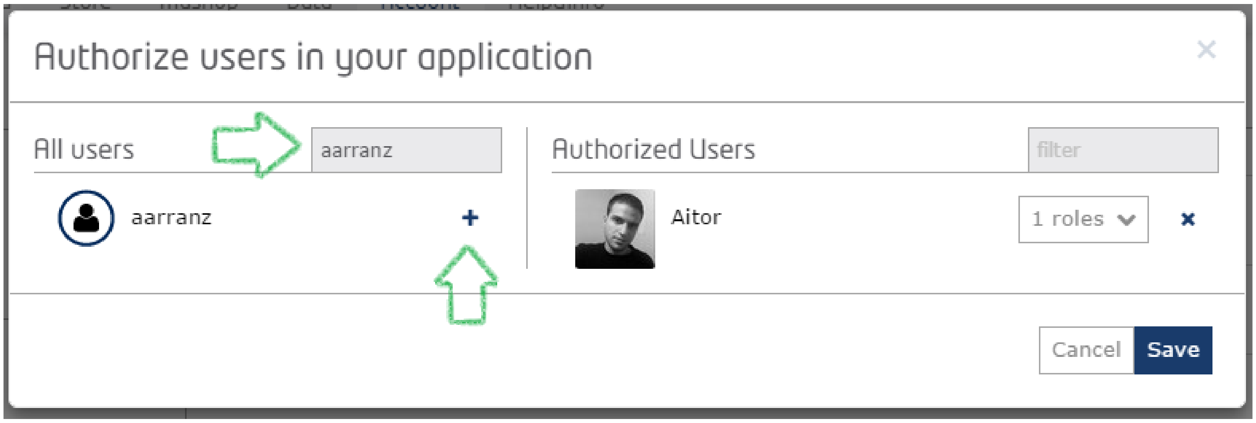
Description
Marketplace test instance.

URL
http://130.206.122.47:8080

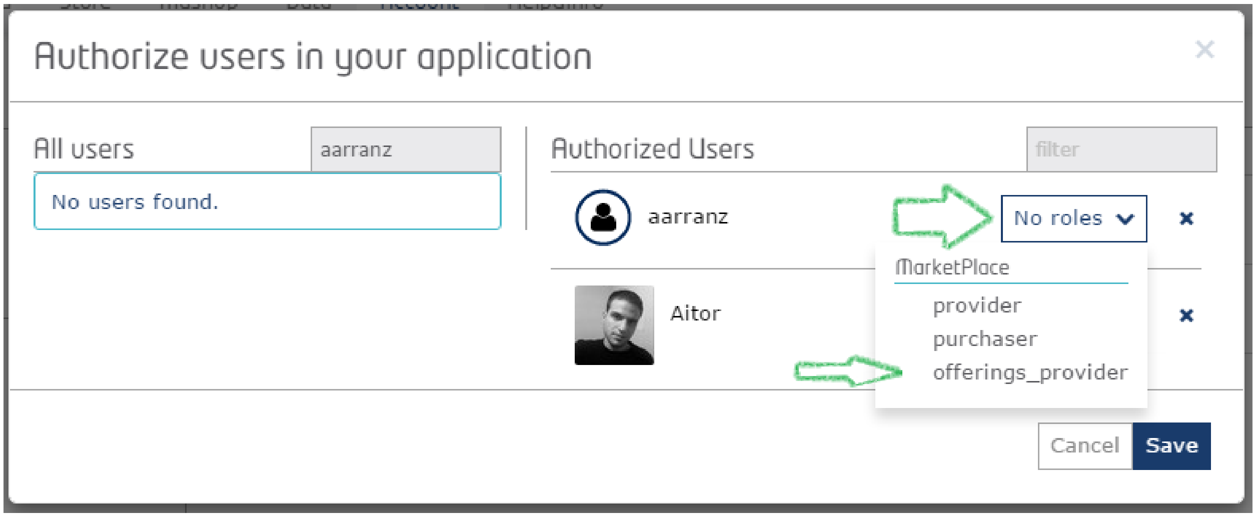
Callback URL
http://130.206.122.47:8080/WMarket/callback?client_name=FIWAREClient

MarketPlace does not have any authorized organizations.

A new dialog will be opened. Just write the name of the user to which you want to assign the provider role. Then, click the **+** button.



The user will be moved to the **Authorized users** section. Finally, click on the **No Roles** button and assign the provider role as displayed in this image.

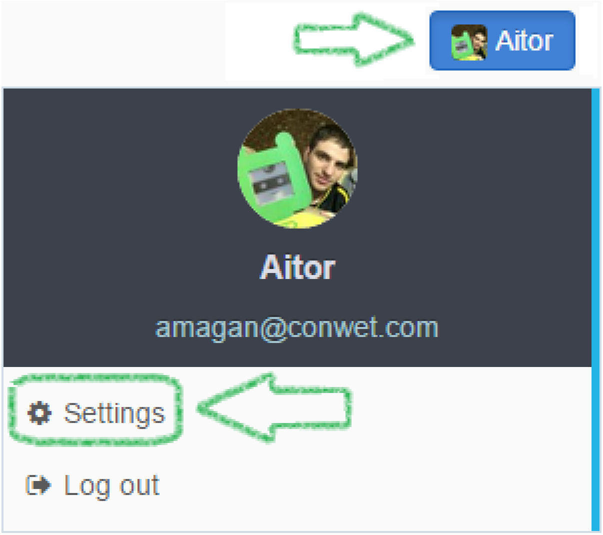


Then click **Save**.

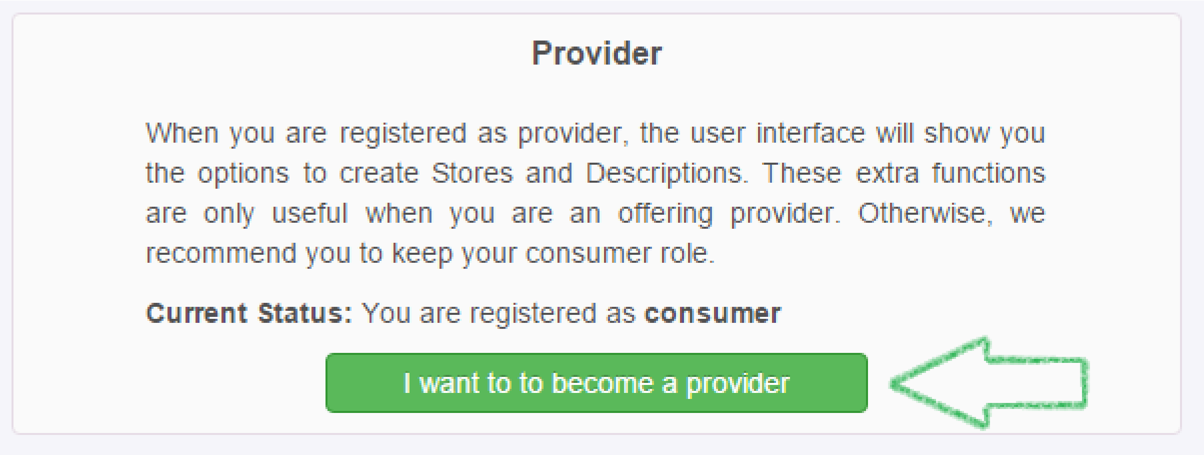
Note: These changes will only take effect when the user logins in again.

1.2.2 Local Authentication

If the WMarket instance you are administering is based on a local authentication system, you do not have to worry about upgrading users to providers since they can do it by themselves in a very easy way by editing their preferences. To do so, users should click on their name and then click the **Settings** button.



After this they will be redirected to the user settings management page where a button to become provider can be found.



Provider

When you are registered as provider, the user interface will show you the options to create Stores and Descriptions. These extra functions are only useful when you are an offering provider. Otherwise, we recommend you to keep your consumer role.

Current Status: You are registered as **consumer**

1.3 Building from Sources

This document explains how a user can obtain the WMarket sources to compile it and generate a customized WAR that can be run in an Application Server (i.e. Tomcat...)

Note: This document is intended for Java developers with experience in developing their own Web Applications

1.3.1 System Requirements

To compile WMarket you are required to install the following software:

- Git: to obtain the WMarket Source Code
- Maven 3: to compile the source code

You can easily install these programs by using the package manager software offered by your operating system.

1.3.2 Obtaining the Source Code

Once that you have installed the minimum requirements, you are able to obtain the source code. To do it, execute the following command:

```
$ git clone git@github.com:conwetlab/WMarket.git
```

1.3.3 Configuring WMarket

Before compiling the code, you can be interested in editing some preferences to customize your WMarket instance. In this way, you will be able to:

- Modify the database used to store the information
- Modify the folder where the indexes are stored
- Select the authorization system (local or external (i.e. IdM))
- Modify the folder used for media files and their maximum size

- Modify the period used to update descriptions

You can find a full explanation of all these configurations at *WMarket Configuration*

1.3.4 Compiling the Source Code

To compile the code, you can execute the following command:

```
$ mvn install
```

Note: You can execute the command with the argument `-DskipTests` so the tests will not be executed and the WAR will be generated faster.

This will generate a new folder called `target` that will contain the WAR that you can deploy on your Application Server. This WAR is called `WMarket.war`

1.3.5 Executing the tests

WMarket is provided with unit tests and integration tests. You can run the unit tests by executing this command:

```
$ mvn test
```

If you want to run all the tests (unit and integration) you must run the following command:

```
$ mvn integration-test
```

1.4 User and Programmer Guide

This guide involves all the processes that can be performed by final users and programmers.

This guide is related to the version 4.4.3 of WMarket, released in September 2015.

1.4.1 User Guide

WMarket is a software that eases the users the process of looking for the best offering in different Stores since it aggregates them and all their offerings so users can find what they need more easily.

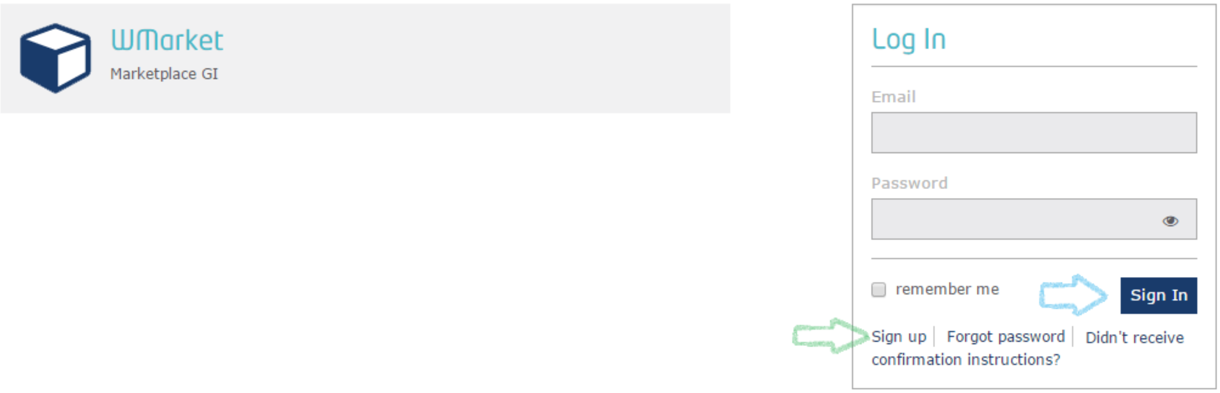
The user guide is divided into two parts: one related to consumers (users interested in acquiring offerings) and another one related to providers (users interested in acquiring and publishing offerings).

Consumers

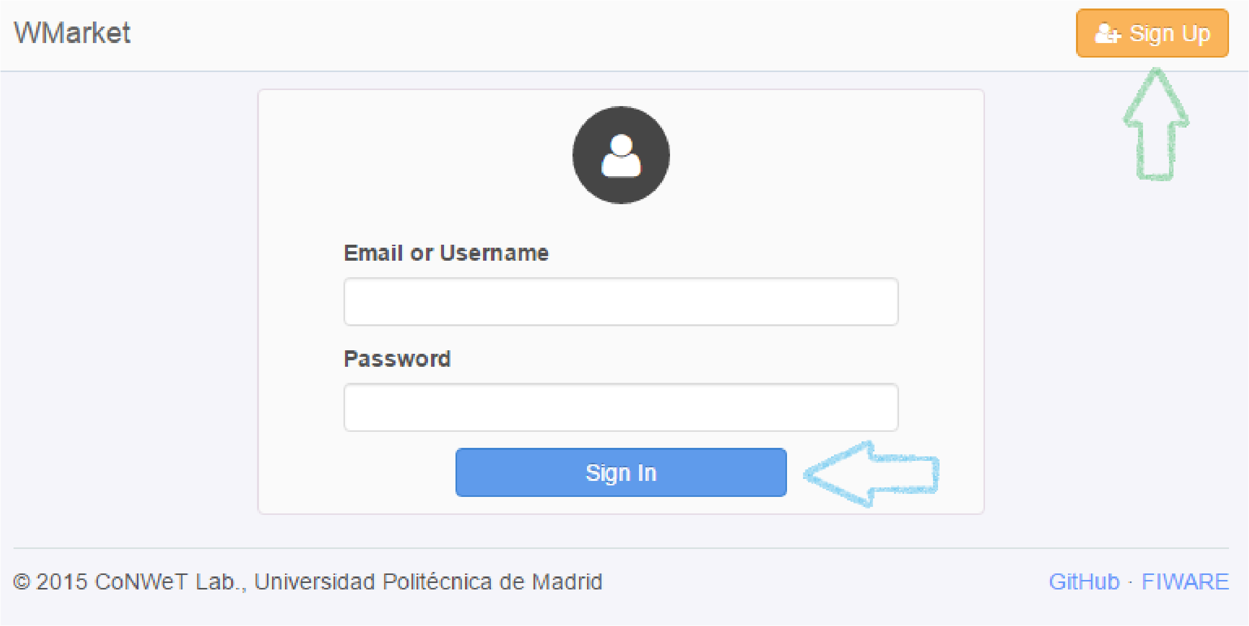
Users Management

Registering and Logging In The first step to access WMarket is to create a new user account. This process may vary depending on the authentication system selected by the WMarket admin.

If the WMarket instance you are accessing uses an external authentication system (i.e. FIWARE IdM), you are not required to create a new user account: you can use the one existing in the IdM. The first time that you access the instance, you will be redirected to the IdM where you have to introduce your credentials. Once that you have introduced your credentials, click **Sign in** (the button indicated by a blue arrow). If you do not have a FIWARE account yet, you can create a new one by clicking the **Sign up** button (the one indicated the by the green arrow).



On the other hand, if you are using the local authentication system, when you access the instance and you are not logged in, a log in dialog will be displayed.



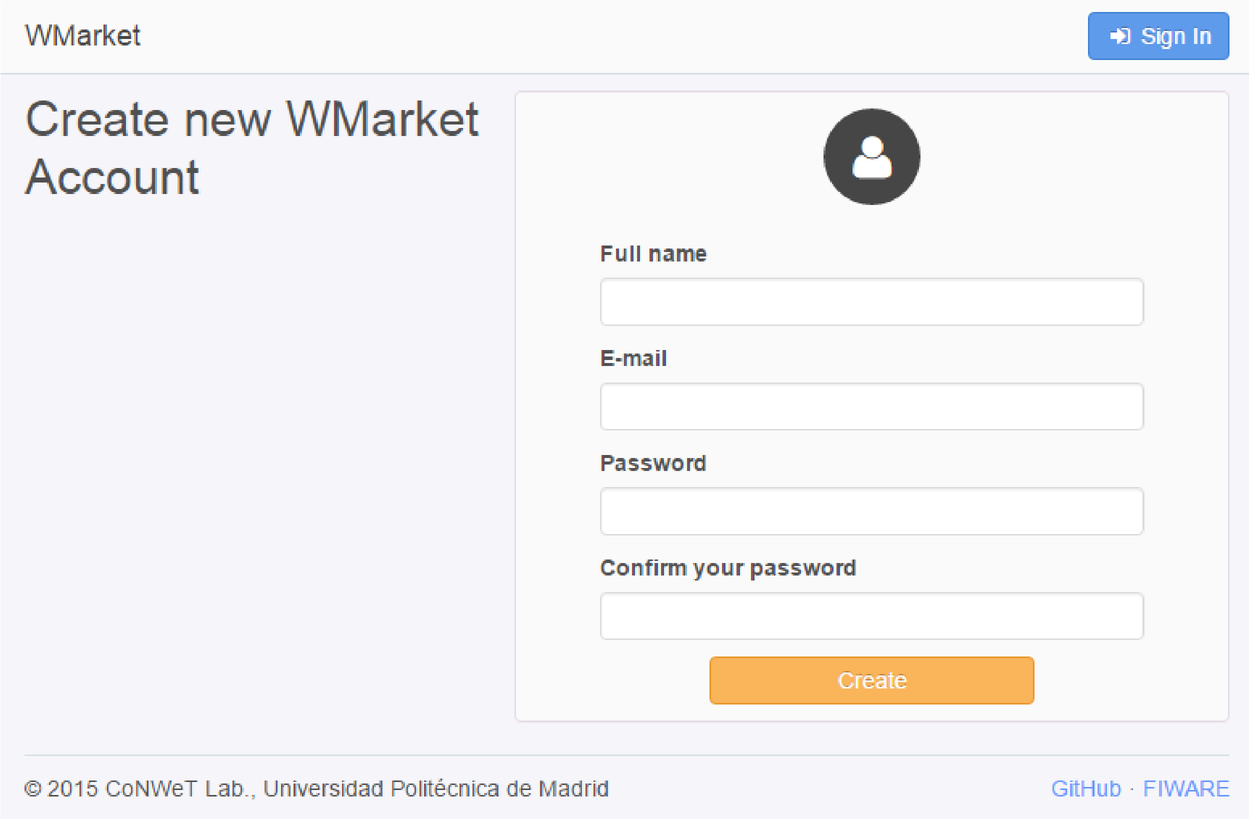
If it is your first time in this instance, you have to create a new account. To do so, just click the **Sign up** button (the once indicated by the green arrow). Otherwise, introduce your credentials (your email and your password) and click the **Sign in** button (the one indicated by the blue arrow).

To create a new account, you will be required to introduce some details:

- **Full Name:** your complete full name. It must contain between 3 and 30 characters. Only letters and spaces are accepted.
- **Email:** your email address. It must be a valid email address according to the email pattern. Two users cannot use the same email address.
- **Password:** password to access your account. It must contain between 8 and 30 characters. It must contain one number, one letter and one special character (`!`, `#`, `$`, `%`, `&` and `?`).

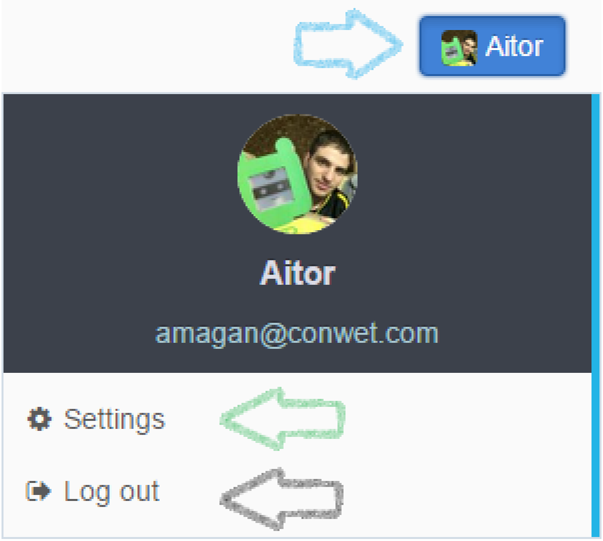
Note: If the user cannot be created because one or more fields are invalid, these fields will be highlighted in red colour and a message will be attached to every invalid field so the user can understand why these fields are invalid.

Once that you have introduced these details, just click **Create**.


WMarket
Sign In
Create new WMarket Account
Full name
E-mail
Password
Confirm your password
Create
© 2015 CoNWeT Lab., Universidad Politécnica de Madrid
GitHub · FIWARE

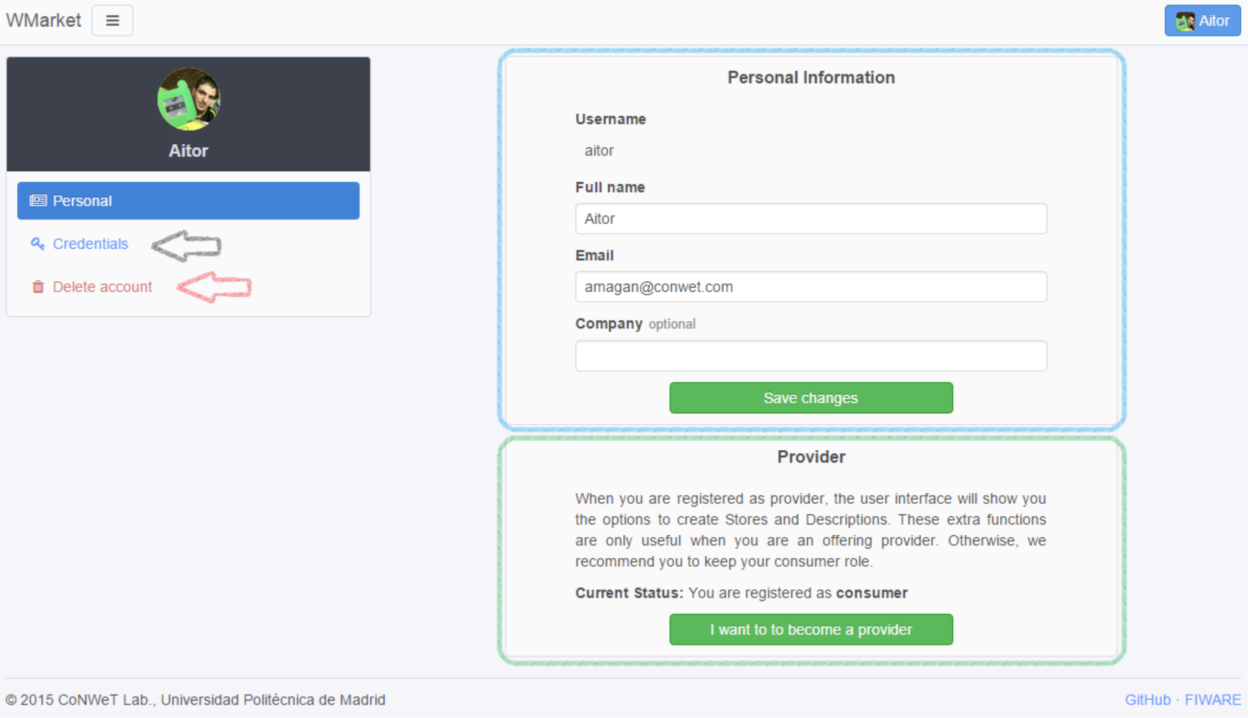
Note: When using the local authentication system, you can log in using your email or your user name. This user name is based on your full name and you can get it in the profile management page as will be explained below.

Managing User Profile Once that you are logged in, you will be redirected to the WMarket main page where you will be able to find all the provided offerings. To manage your profile, just click in your user name on the upper right corner (the button indicated by the blue arrow) and then click the **Settings** button (the one indicated by the green arrow).



Note: As can be seen, you can log out by clicking **Log out** button (the one indicated by the black arrow).

When you click on **Settings**, you are redirected to the profile management page.



Personal Information

Username

aitor

Full name

Aitor

Email

amagan@conwet.com

Company optional

Provider

When you are registered as provider, the user interface will show you the options to create Stores and Descriptions. These extra functions are only useful when you are an offering provider. Otherwise, we recommend you to keep your consumer role.

Current Status: You are registered as **consumer**

© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

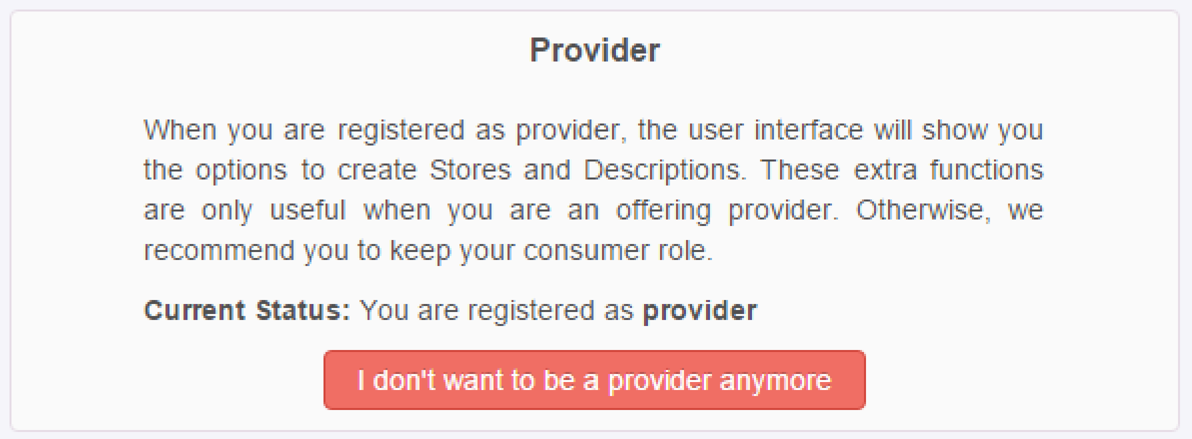
If the instance is using the local authentication system, this page will allow you to:

- Change your personal information
- Become a provider
- Change your password
- Delete your account

If you want to change your personal information, just type your new details into the **Personal Information** box (the one rounded with a blue rectangle) and click the **Save Changes** button.

Note: As can be seen, the **Personal Information** box contains your user name. As stated before, you can also use this user name to log in.

You will also be able to become a provider in order to enable new options to create stores and publish offerings. To do so, just click the **I want to become a provider** button located within the green box. If you are already a provider and you just want to be a simple consumer, just click the **I don't want to be a provider anymore** button.

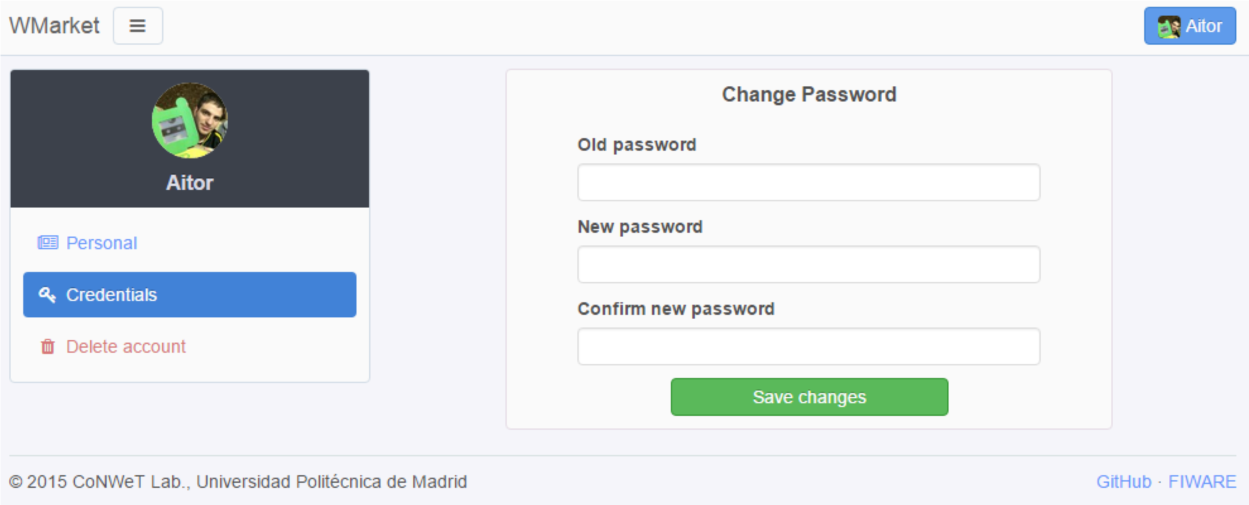


Provider

When you are registered as provider, the user interface will show you the options to create Stores and Descriptions. These extra functions are only useful when you are an offering provider. Otherwise, we recommend you to keep your consumer role.

Current Status: You are registered as **provider**

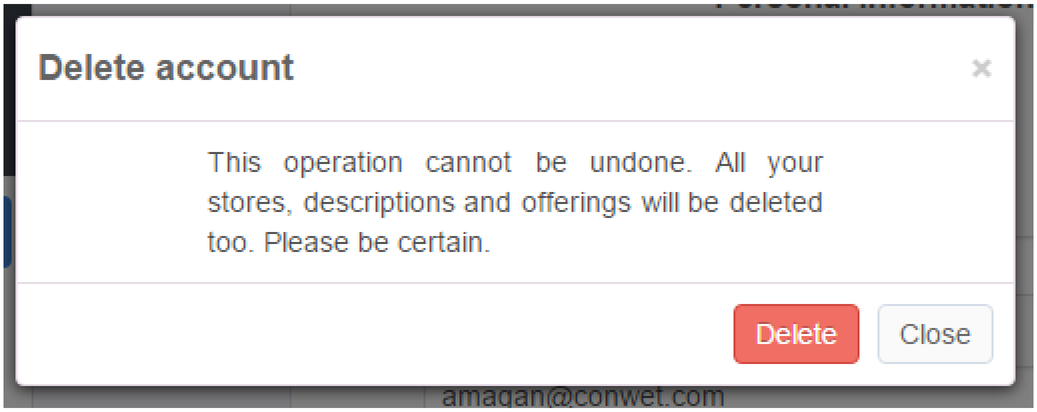
To change your password, click the **Credentials** button (the one indicated by the black arrow). Then, you will be redirected to a new page that will allow you to change your password. In this page, just write your old password and type twice your new password. If your old password is correct and the new password fits the password requirements, your password will be changed.



© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

Note: When you change your password, the system will close your session so you will be required to log in again.

Finally, you can also deleted your account. To do so, just click the **Delete account** button (the one indicated by the red button). You will be required to confirm the operation as can be seen in the following image.



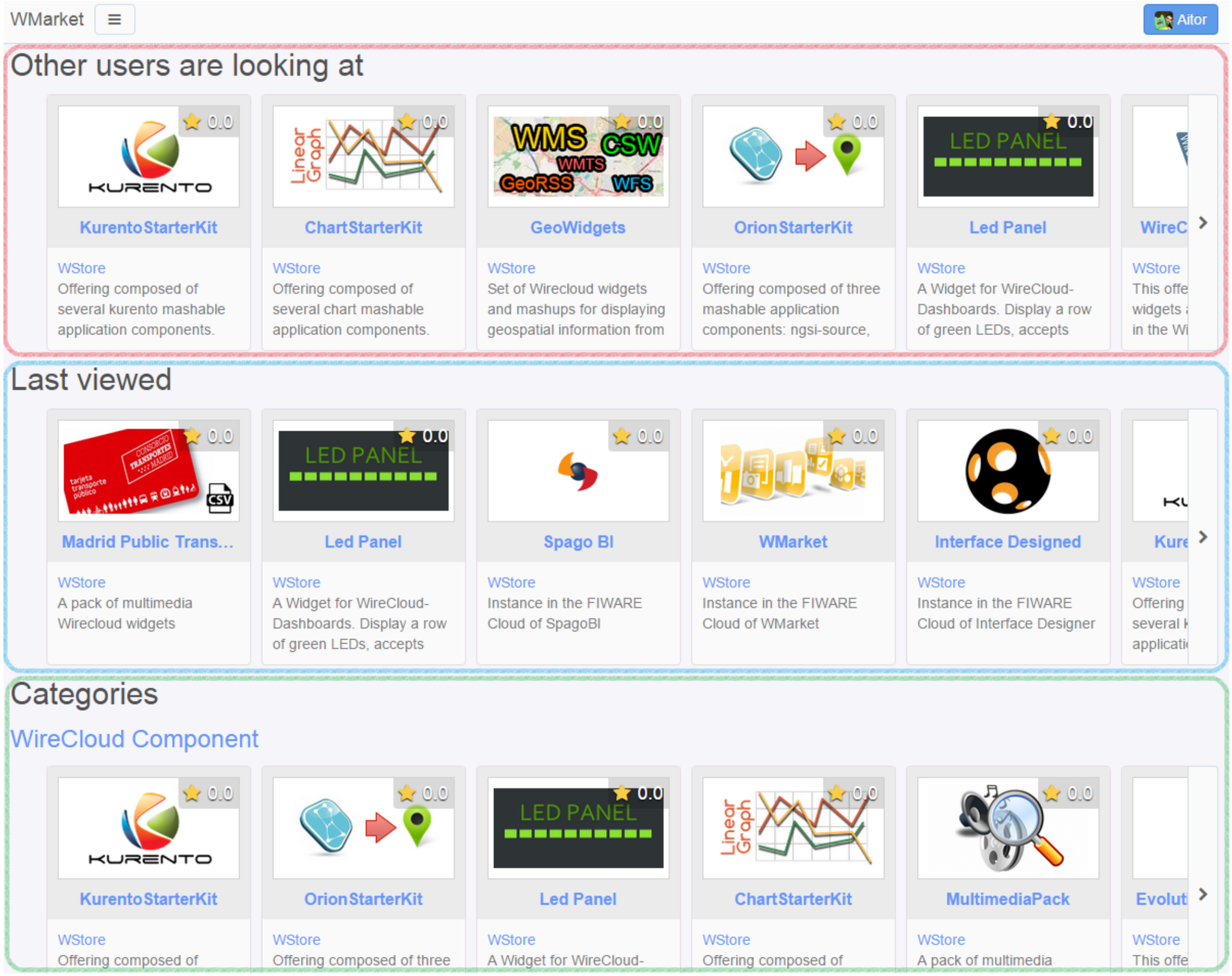
Delete account

This operation cannot be undone. All your stores, descriptions and offerings will be deleted too. Please be certain.

Note: When you delete your account, all your content (reviews, stores, offerings, descriptions...) will be deleted.

Browsing Offerings

When you access WMarket, you are redirected to the main page.



Offering composed of several kurento mashable application components.

Offering composed of several chart mashable application components.

Set of Wirecloud widgets and mashups for displaying geospatial information from

Offering composed of three mashable application components: ngsi-source,

A Widget for WireCloud-Dashboards. Display a row of green LEDs, accepts

This offe widgets a in the Wi

A pack of multimedia Wirecloud widgets

A Widget for WireCloud-Dashboards. Display a row of green LEDs, accepts

Instance in the FIWARE Cloud of SpagoBI

Instance in the FIWARE Cloud of WMarket

Instance in the FIWARE Cloud of Interface Designer

Offering several k applicati

Offering composed of

Offering composed of three

A Widget for WireCloud-

Offering composed of

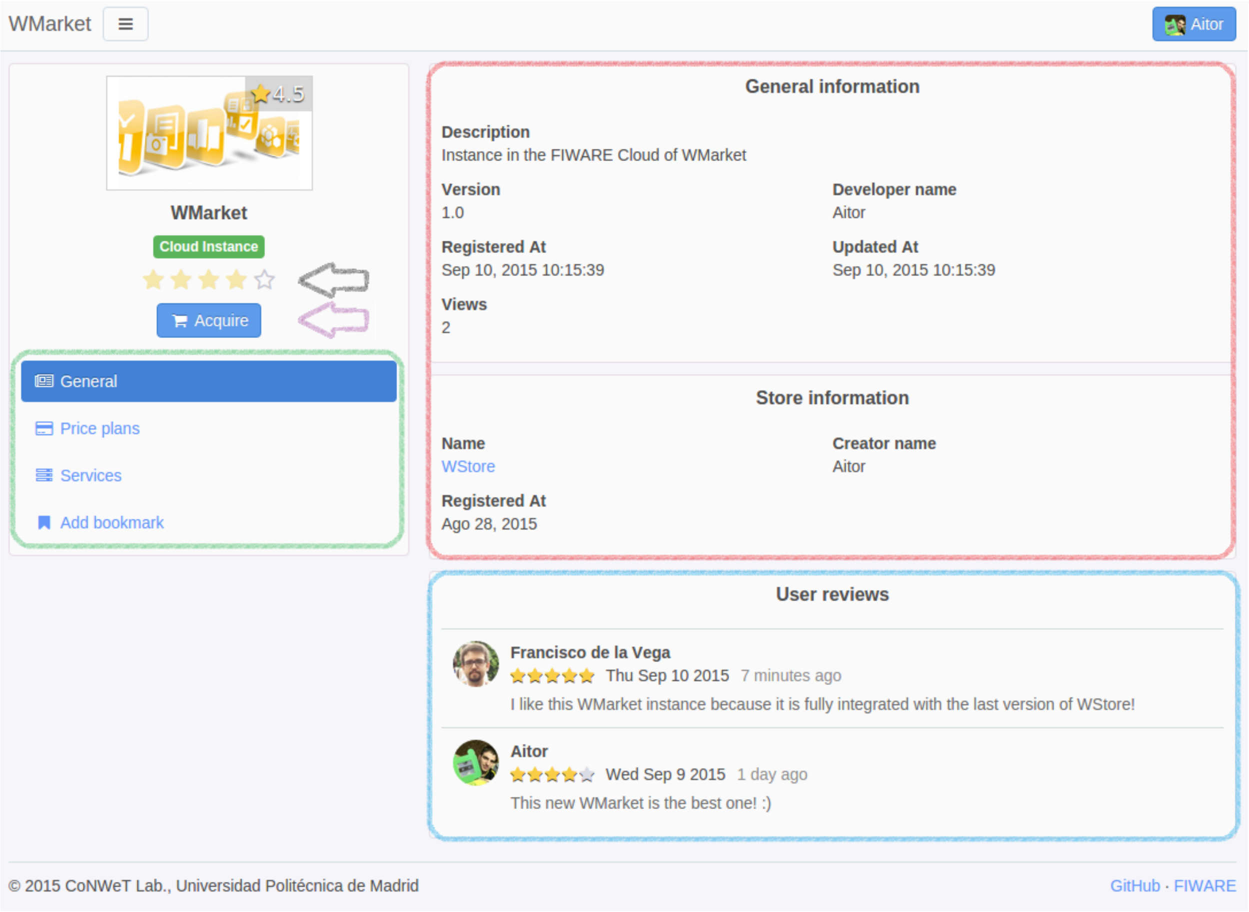
A pack of multimedia

This offe

This page is divided into three parts:

- **Other users are looking at** (red box): shows offerings viewed by other users.
- **Last viewed** (blue box): shows the last offerings viewed by you.
- **Categories** (green box): shows offerings divided by their category. You can click the name of any category to retrieve the full list of offerings contained in this category.

Offering Details You can click any offering to view its details.



General information

Description
Instance in the FIWARE Cloud of WMarket

Version
1.0

Developer name
Aitor

Registered At
Sep 10, 2015 10:15:39

Updated At
Sep 10, 2015 10:15:39

Views
2

Store information

Name
WStore

Creator name
Aitor

Registered At
Ago 28, 2015

User reviews

Francisco de la Vega
★★★★★ Thu Sep 10 2015 7 minutes ago
I like this WMarket instance because it is fully integrated with the last version of WStore!

Aitor
★★★★☆ Wed Sep 9 2015 1 day ago
This new WMarket is the best one! :)

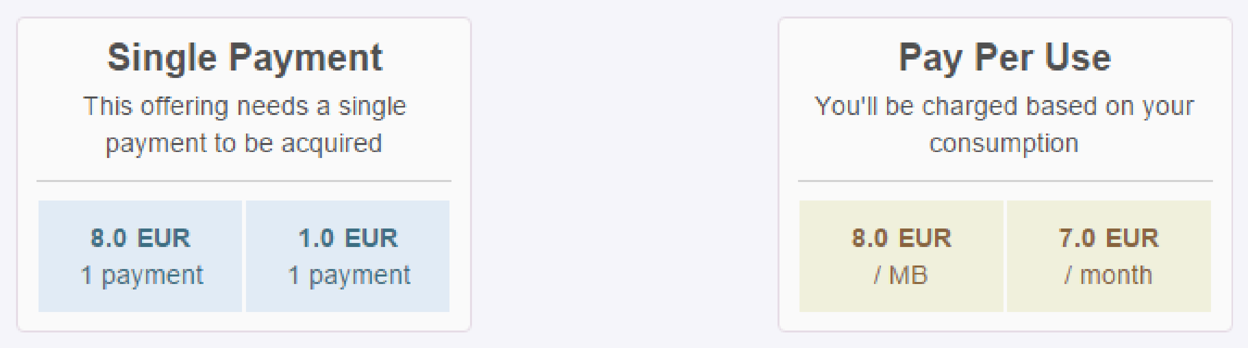
© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

GitHub · FIWARE

This page is divided in different zones:

- The **red box** contains general information about the offering: name, version, description, provider...
- The **blue box** contains the reviews mage by users.
- The **green box** contains all the actions that you can do with the offering: view its price plans, view its services or add it you to your bookmarks.
- The stars indicated by the **black arrow** can be used to review the offering.
- The button indicated by the **purple arrow** will open in a new window (or tab) the Store page where the offering can be acquired.

To view all the price plans included in an offering, just click the **Price Plans** button.



Single Payment

This offering needs a single payment to be acquired

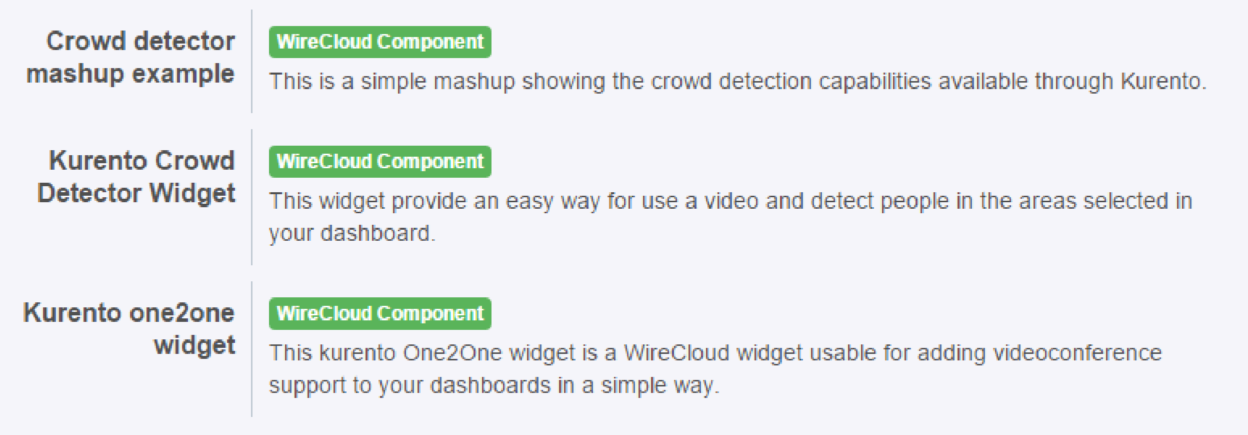
8.0 EUR 1 payment	1.0 EUR 1 payment

Pay Per Use

You'll be charged based on your consumption

8.0 EUR / MB	7.0 EUR / month

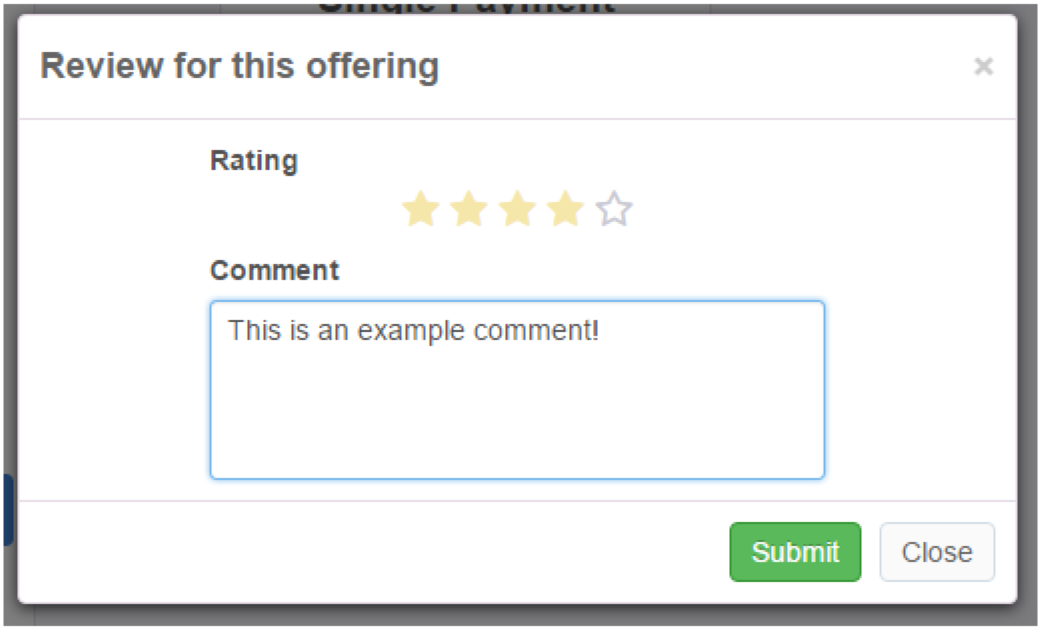
To view all the services included in an offering, just click the **Services** button.



Crowd detector mashup example	WireCloud Component This is a simple mashup showing the crowd detection capabilities available through Kurento.
Kurento Crowd Detector Widget	WireCloud Component This widget provide an easy way for use a video and detect people in the areas selected in your dashboard.
Kurento one2one widget	WireCloud Component This kurento One2One widget is a WireCloud widget usable for adding videoconference support to your dashboards in a simple way.

Reviewing Offerings If you want to review an offering, you can easily do it by clicking one of the stars contained in the black box. The first star means that you do not like the offering while the last one means that you really like the offering. You should click on the appropriate star according to your thoughts.

Once that you have clicked a star, a new dialog will be opened asking for an extended review as can be seen in the following picture.



Review for this offering

Rating

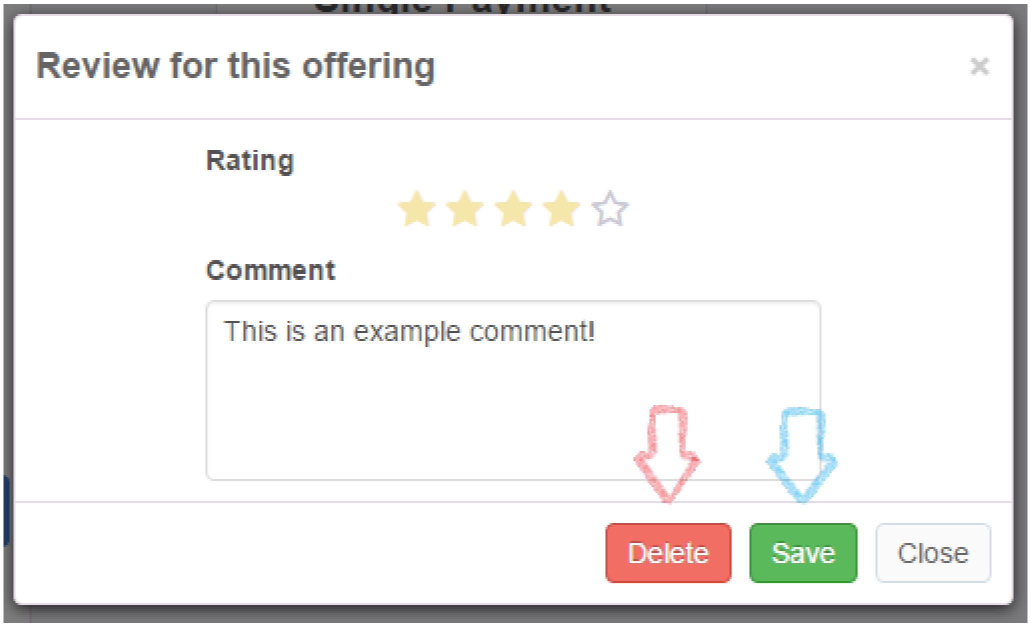
★★★★☆

Comment

This is an example comment!

In this dialog you can change the number of stars and add a brief comment to explain your decision. When you have completed the review, just click the **Submit** button.

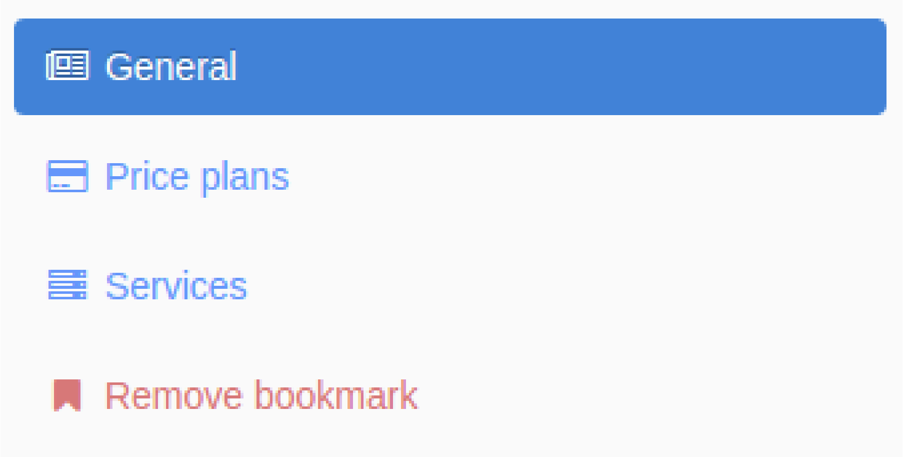
Once that you have reviewed an offering, you can edit or delete your review. To do so, just click one of the stars contained in the black box. A new dialog will be opened with your review.



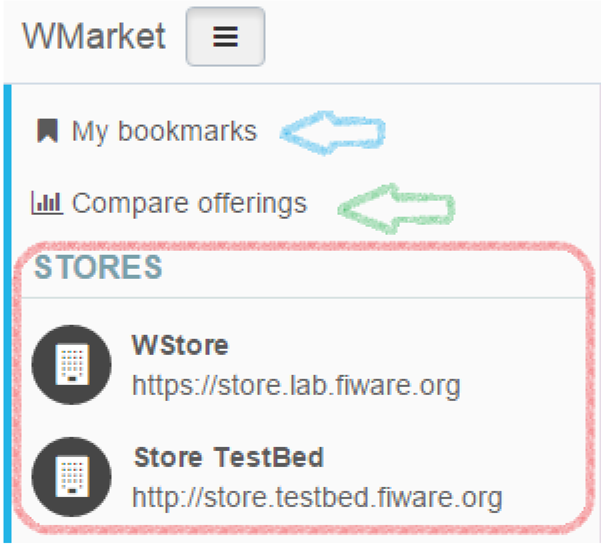
This is an example comment!

To update your previous review, just modify the number of stars and type a new comment. Then, click the **Save** button (the one indicated by the blue arrow). On the other hand, if you want to delete your review, just click the **Delete** button (the one indicated by the red arrow).

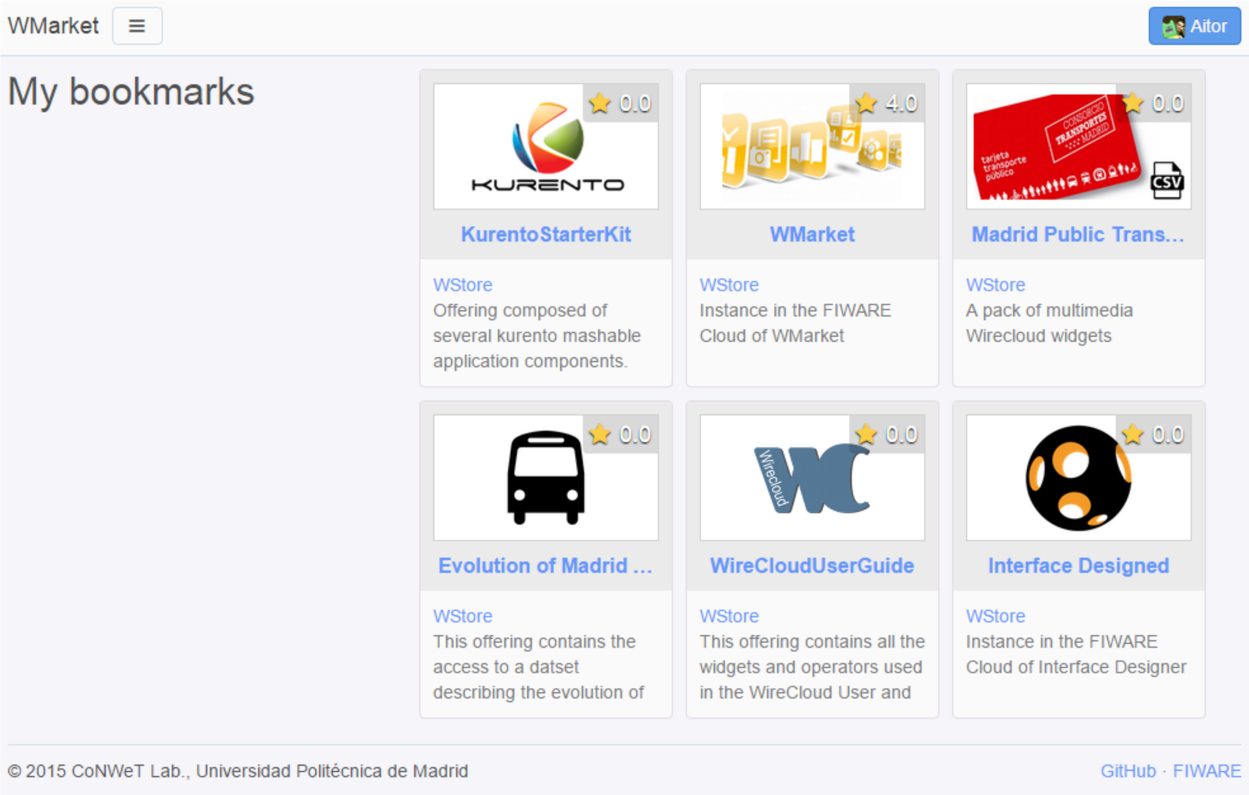
Bookmarking To add an offering to your bookmarks, just click the **Add bookmark** button (contained in the green box). The button will change to **Remove bookmark** so you will be able remove this offering from your bookmarks.



To view the list of bookmarked offerings, you can click the menu button that is next to the **WMarket** text on the left upper corner as can be seen in the following image.



Then click the **My Bookmarks** button (the one indicated by the blue arrow). A screen similar to the following one will be displayed.



WStore
Offering composed of several kurento mashable application components.

WStore
Instance in the FIWARE Cloud of WMarket

WStore
A pack of multimedia Wirecloud widgets

WStore
This offering contains the access to a datset describing the evolution of

WStore
This offering contains all the widgets and operators used in the WireCloud User and

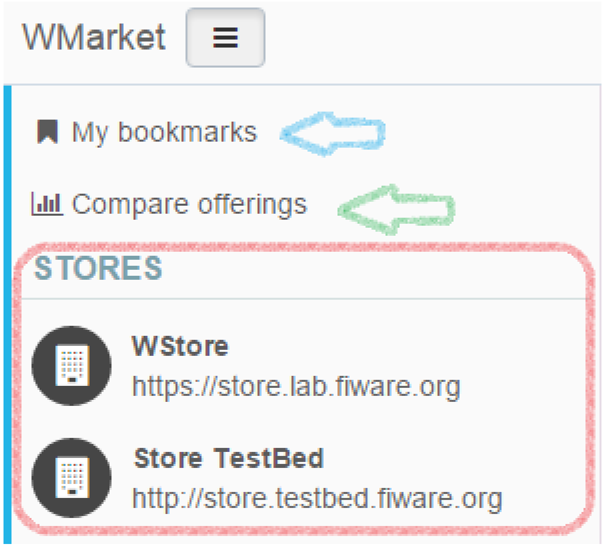
WStore
Instance in the FIWARE Cloud of Interface Designer

© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

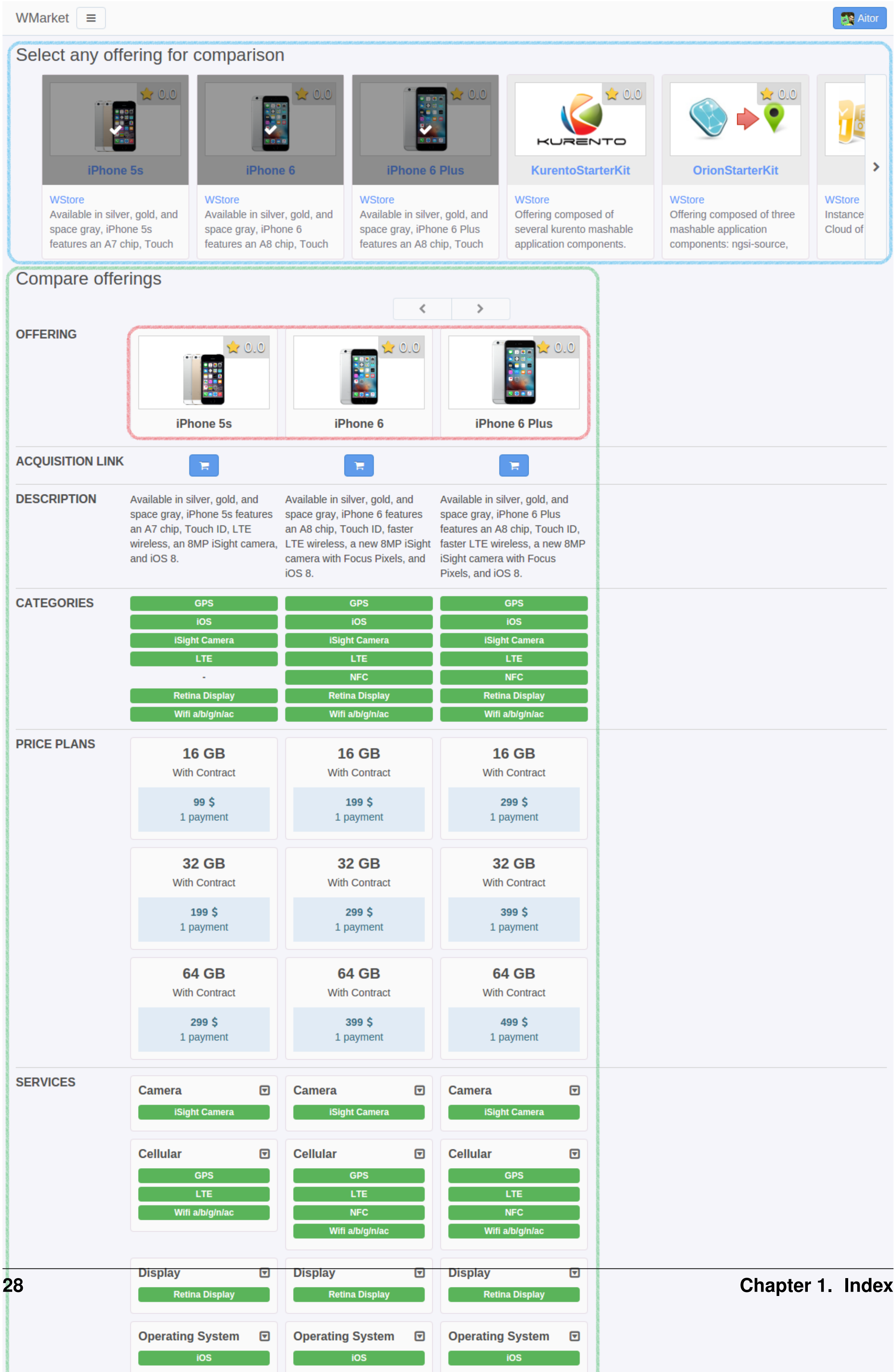
You can click any offering to obtain extended details about the offering.

Comparing Offerings

WMarket allows users to compare different offerings. To do so, click the menu button that is next to the **WMarket** text on the left upper corner (as can be seen in the following image) and click the **Compare offerings** button (the one indicated by the green arrow).



You will be redirected to a new page that will allow you to create comparisons.



Select any offering for comparison

0.0 ✓ iPhone 5s	0.0 ✓ iPhone 6	0.0 ✓ iPhone 6 Plus	0.0 KurentoStarterKit	0.0 OrionStarterKit	
WStore Available in silver, gold, and space gray, iPhone 5s features an A7 chip, Touch	WStore Available in silver, gold, and space gray, iPhone 6 features an A8 chip, Touch	WStore Available in silver, gold, and space gray, iPhone 6 Plus features an A8 chip, Touch	WStore Offering composed of several kurento mashable application components.	WStore Offering composed of three mashable application components: ngsi-source,	WStore Instance Cloud of

Compare offerings

OFFERING	0.0 iPhone 5s	0.0 iPhone 6	0.0 iPhone 6 Plus
ACQUISITION LINK	🛒	🛒	🛒
DESCRIPTION	Available in silver, gold, and space gray, iPhone 5s features an A7 chip, Touch ID, LTE wireless, an 8MP iSight camera, and iOS 8.	Available in silver, gold, and space gray, iPhone 6 features an A8 chip, Touch ID, faster LTE wireless, a new 8MP iSight camera with Focus Pixels, and iOS 8.	Available in silver, gold, and space gray, iPhone 6 Plus features an A8 chip, Touch ID, faster LTE wireless, a new 8MP iSight camera with Focus Pixels, and iOS 8.
CATEGORIES	GPS iOS iSight Camera LTE - Retina Display Wifi a/b/g/n/ac	GPS iOS iSight Camera LTE NFC Retina Display Wifi a/b/g/n/ac	GPS iOS iSight Camera LTE NFC Retina Display Wifi a/b/g/n/ac
PRICE PLANS	16 GB With Contract 99 $ 1 payment	16 GB With Contract 199 $ 1 payment	16 GB With Contract 299 $ 1 payment
	32 GB With Contract 199 $ 1 payment	32 GB With Contract 299 $ 1 payment	32 GB With Contract 399 $ 1 payment
	64 GB With Contract 299 $ 1 payment	64 GB With Contract 399 $ 1 payment	64 GB With Contract 499 $ 1 payment
SERVICES	Camera iSight Camera	Camera iSight Camera	Camera iSight Camera
	Cellular GPS LTE Wifi a/b/g/n/ac	Cellular GPS LTE NFC Wifi a/b/g/n/ac	Cellular GPS LTE NFC Wifi a/b/g/n/ac
	Display Retina Display	Display Retina Display	Display Retina Display
	Operating System iOS	Operating System iOS	Operating System iOS

To compare offerings, just look for the offerings that you want to compare in the **Select any offering for comparison** section (the one rounded with a blue rectangle) and click them. Offerings will appear in the **Compare Offerings** section (the one rounded with a green rectangle) as you click them.

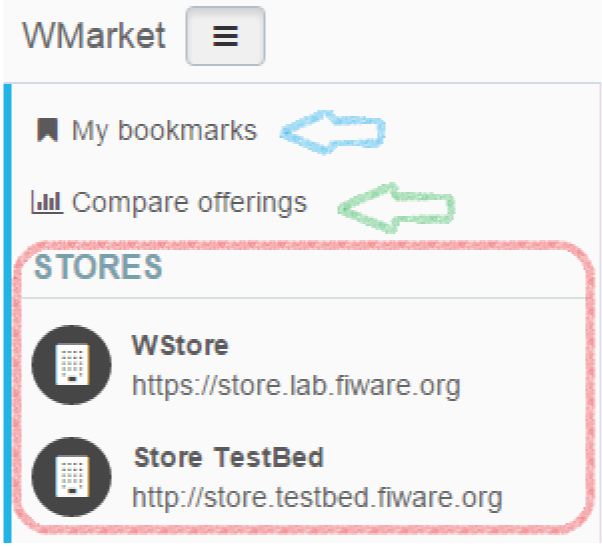
To remove any offering from the comparison, just click on its picture (the one contained in the red rectangle).

The comparison will show you:

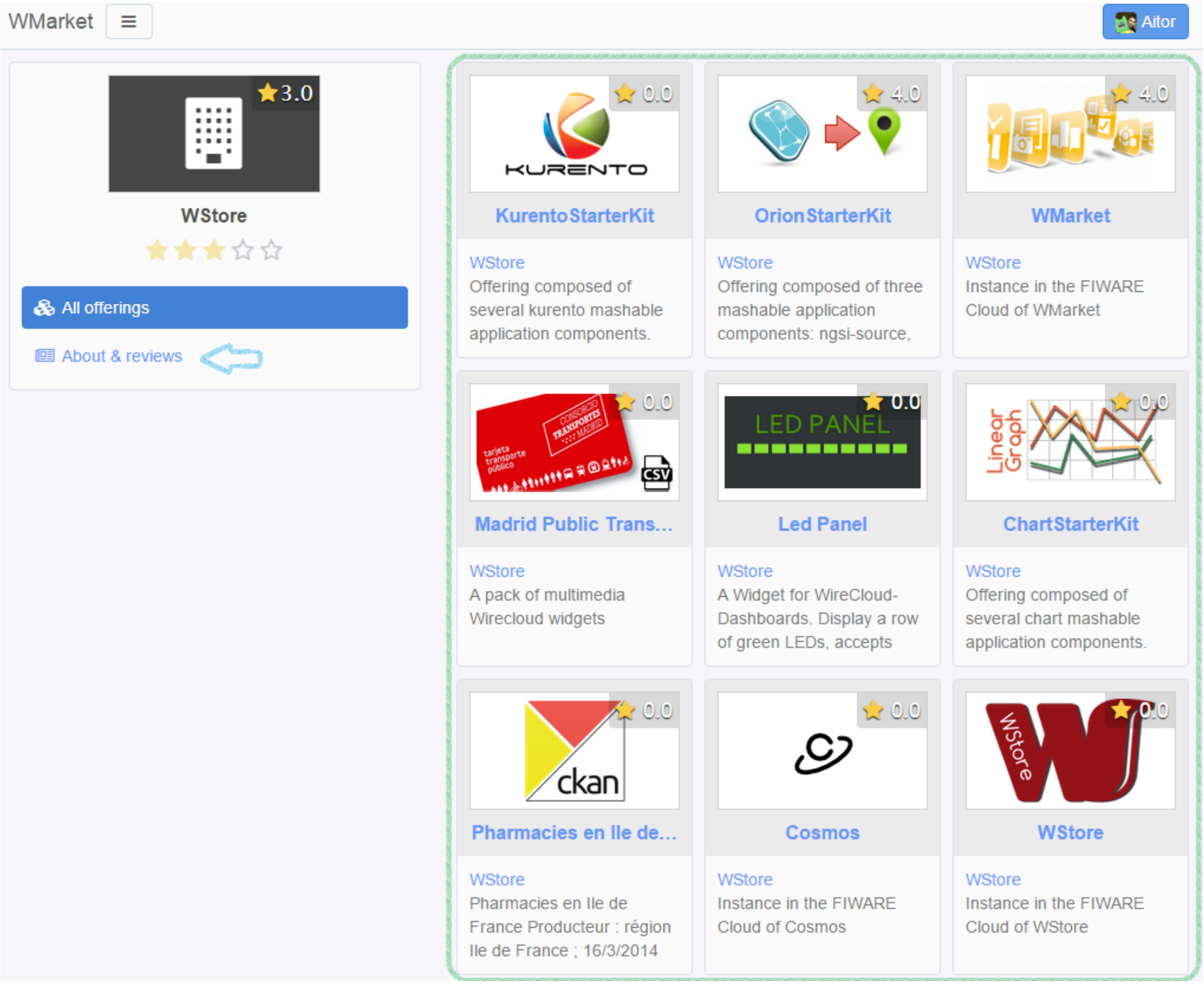
- The name and the image of each offering.
- A link to acquire each offering.
- The description of each offering.
- The categories of each offering. Categories will be ordered and aligned in order to highlight the difference of categories between the compared offerings.
- The price plans of each offering (ordered alphabetically).
- The services of each offering. Services will be ordered and aligned in order to highlight the difference of services between the different offerings.

Browsing Stores

Apart from retrieving all the registered offerings, you can also get the offerings that are included in a certain Store. To do so, click the menu button that is next to the **WMarket** text on the left upper corner (as can be seen in the following image) and click on one of the registered stores.

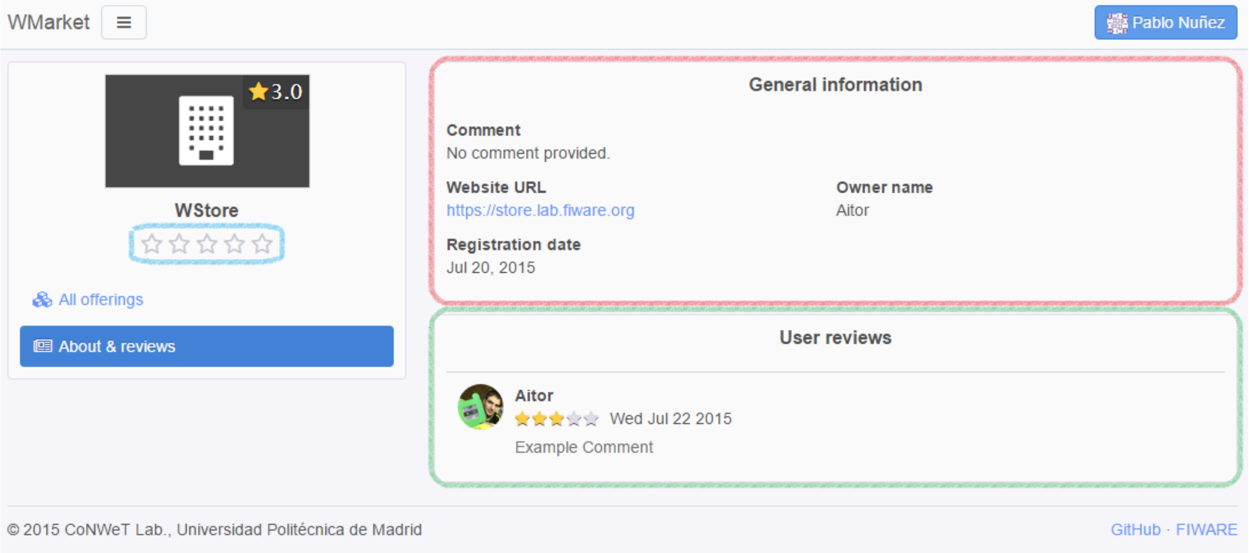


Once that you click on one store, you will see a screen similar to the following one:


WMarket
Aitor
3.0
WStore
All offerings
About & reviews
0.0
KurentoStarterKit
WStore
Offering composed of several kurento mashable application components.
4.0
OrionStarterKit
WStore
Offering composed of three mashable application components: ngsi-source,
4.0
WMarket
WStore
Instance in the FIWARE Cloud of WMarket
0.0
Madrid Public Trans...
WStore
A pack of multimedia Wirecloud widgets
0.0
Led Panel
WStore
A Widget for WireCloud-Dashboards. Display a row of green LEDs, accepts
0.0
ChartStarterKit
WStore
Offering composed of several chart mashable application components.
0.0
Pharmacies en Ile de...
WStore
Pharmacies en Ile de France Producteur : région Ile de France ; 16/3/2014
0.0
Cosmos
WStore
Instance in the FIWARE Cloud of Cosmos
0.0
WStore
WStore
Instance in the FIWARE Cloud of WStore

The area rounded by the green rectangle contains all the offering registered in this Store.

To retrieve the details about the Store and its review, just click the **About & reviews** button (the one indicated by the blue arrow). You will be redirected to the following page.


WMarket

Pablo Nuñez

3.0

WStore

All offerings

About & reviews

General information

Comment
No comment provided.

Website URL
https://store.lab.fiware.org

Owner name
Aitor

Registration date
Jul 20, 2015

User reviews

Aitor
Wed Jul 22 2015
Example Comment

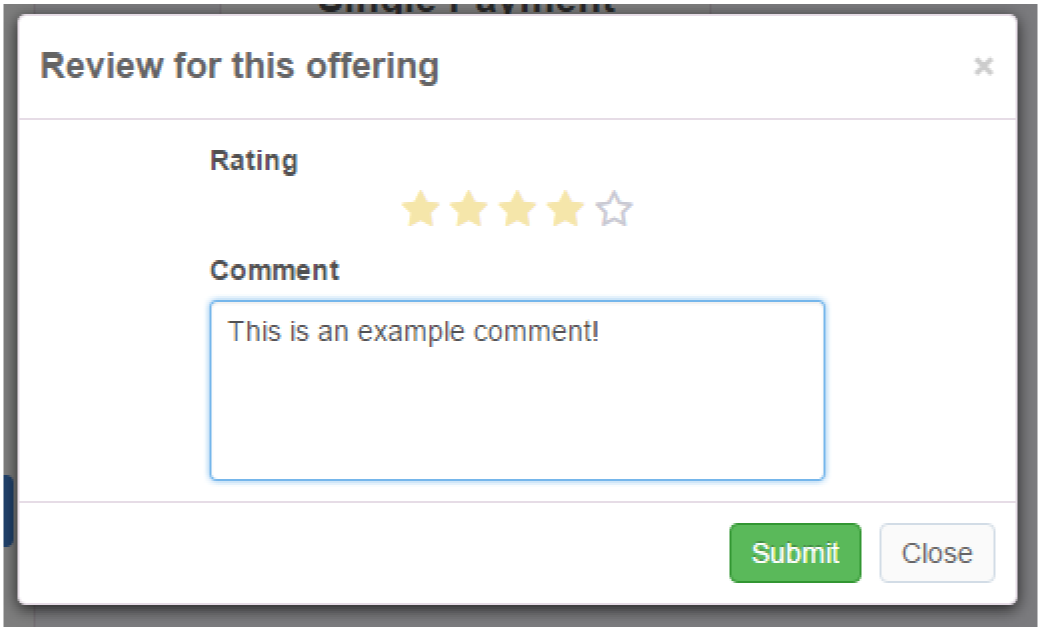
© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

GitHub · FIWARE

The **General Information** box (the one rounded with the red rectangle) contains general information about the Store (comment, web site, creator and registration date) while the **Users review** box (the one rounded with the blue rectangle) contains all the reviews made by users.

Reviewing Stores If you want to review a store, you can easily do it by clicking one of the stars contained in the blue box. The first star means that you do not like the store while the last one means that you really like the store. You should click on the appropriate star according to your thoughts.

Once that you have clicked a star, a new dialog will be opened asking for an extended review as can be seen in the following picture.


Review for this offering

Rating

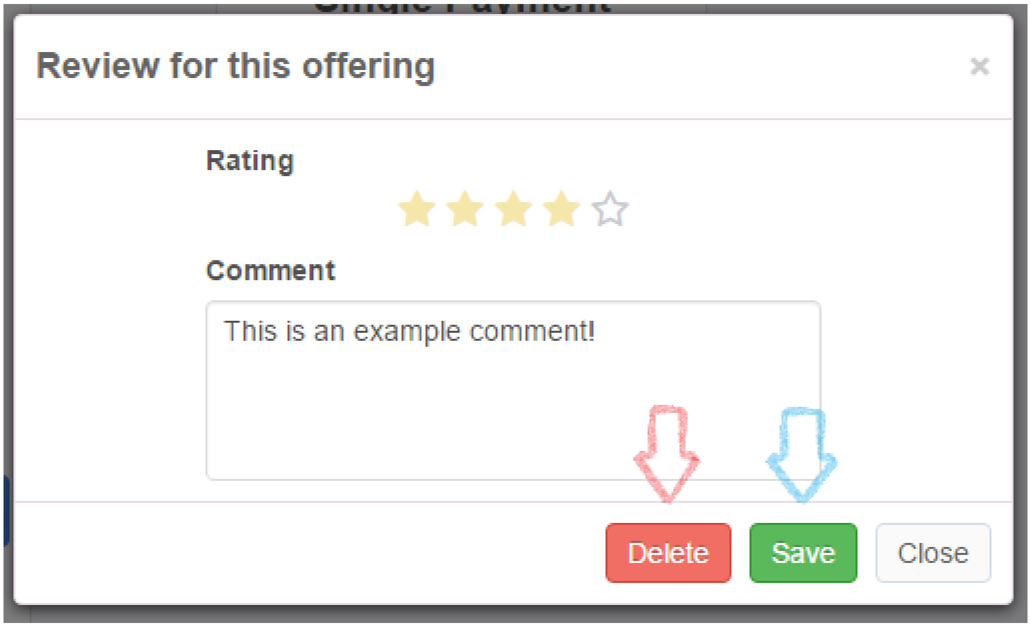
★★★★☆

Comment

This is an example comment!

In this dialog you can change the number of stars and add a brief comment to explain your decision. When you have completed the review, just click the **Submit** button.

Once that you have reviewed the store, you can edit or delete your review. To do so, just click one of the stars contained in the blue box. A new dialog will be opened with your review.



This is an example comment!

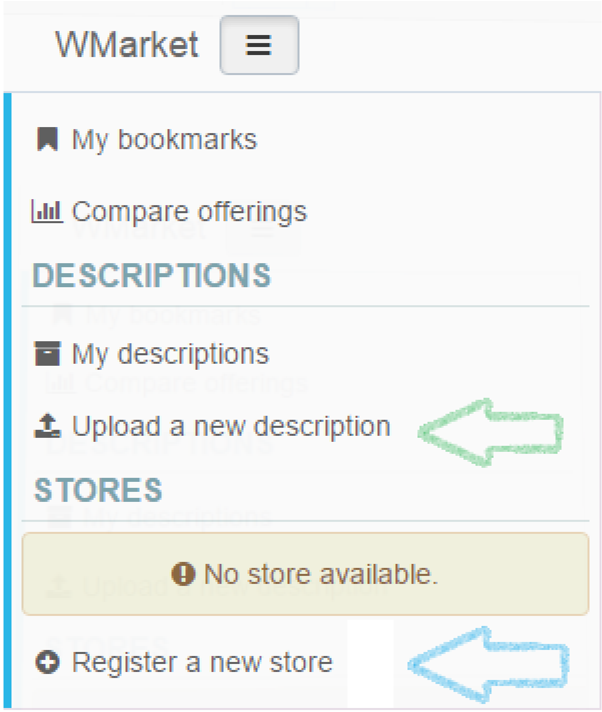
To update your previous review, just modify the number of stars and type a new comment. Then, click the **Save** button (the one indicated by the blue arrow). On the other hand, if you want to delete your review, just click the **Delete** button (the one indicated by the red arrow).

Providers

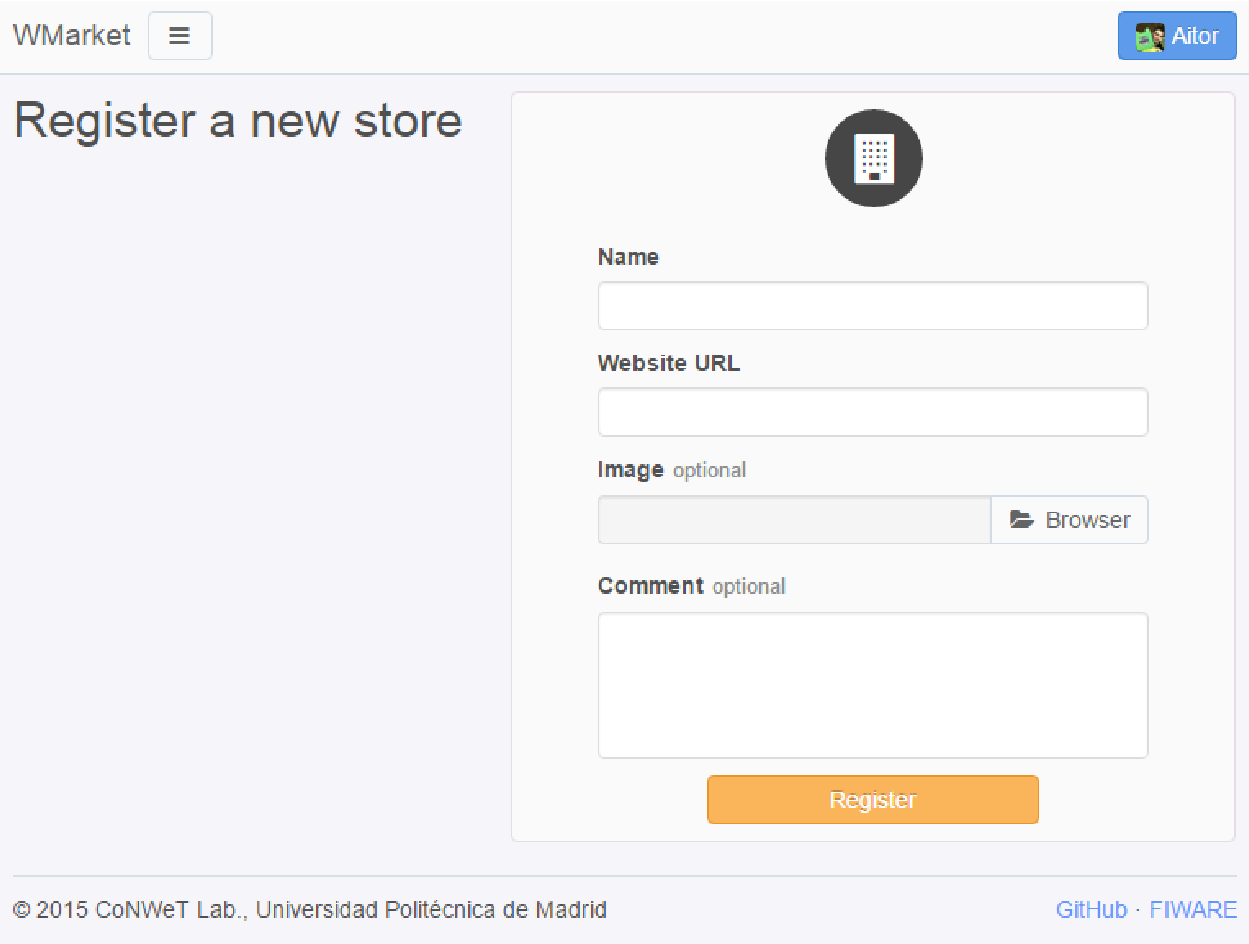
As explained before, providers are those users who want to create stores and upload offerings to WMarket. To become a provider, you can refer to the Administration Guide.

Stores Management

Creating Stores To register a new store, click in the menu button that is next to the **WMarket** text on the left upper corner (as can be seen in the following image) and click on the **Register a new store** button (the one indicated by the blue arrow).



The following screen will be opened:

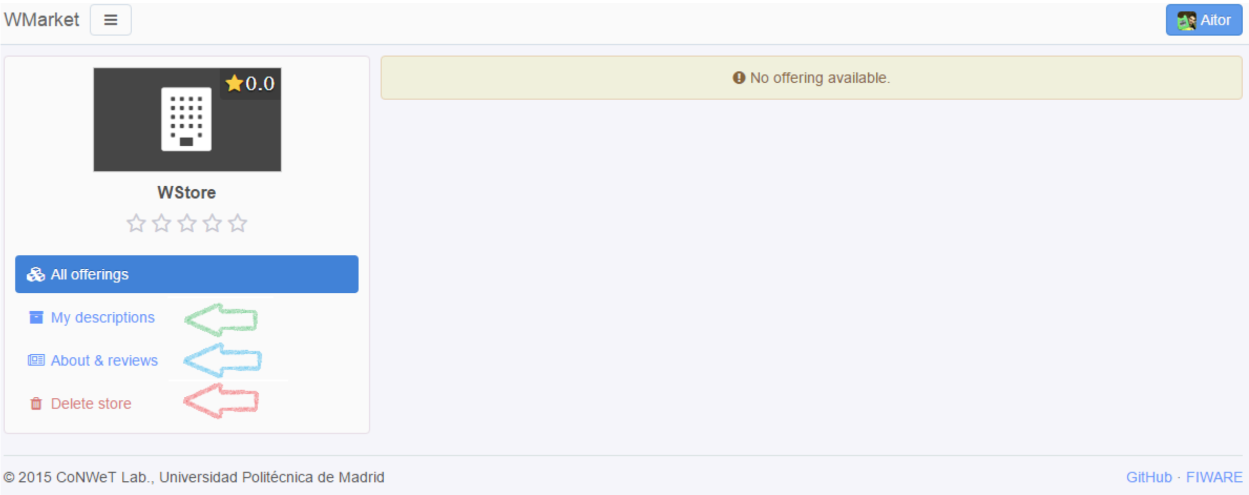


Introduce the following content:

- **Name:** The name of the Store. It must contain between 3 and 100 characters. Only letters, numbers, white spaces, dots and hyphens are accepted.
- **Website URL:** The URL where the store is hosted. The URL must be valid.
- **Image:** An image that will ease the process of identifying your store. You can upload it from your computer. The maximum size accepted (by default) is 3 MB.
- **Comment:** A brief description of your store. You can type until 200 characters.

Note: If the store cannot be created because one or more fields are invalid, these fields will be highlighted in red colour and a message will be attached to every invalid field so the user can understand why these fields are invalid.

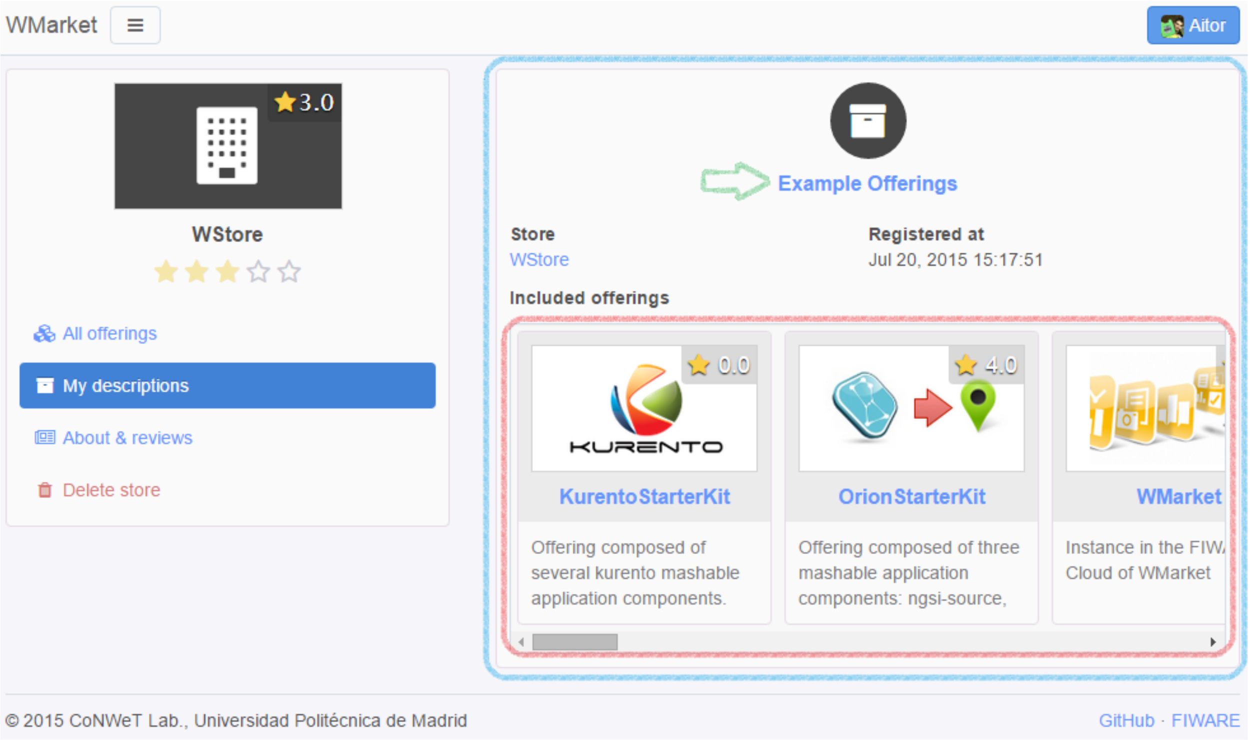
If you have introduced these fields correctly, you will be redirected to the main page of the Store, where you will be able to manage it.



No offering available.

© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

Managing Existing Stores To view all the descriptions that you have uploaded to the Store, click the **My descriptions** button (the one indicated by a green arrow). A screen similar to the following one will be displayed:



Store
WStore

Registered at
Jul 20, 2015 15:17:51

Included offerings

KurentoStarterKit: Offering composed of several kurento mashable application components.

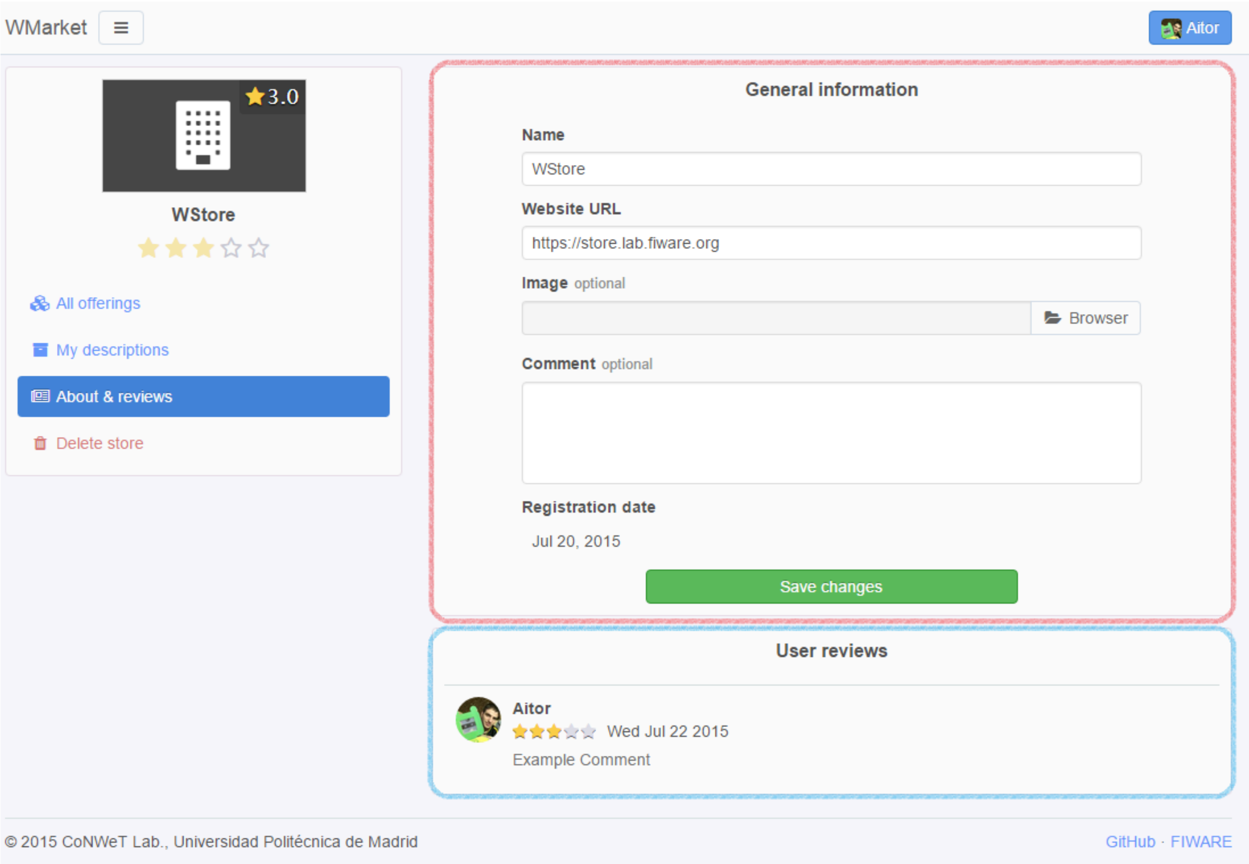
OrionStarterKit: Offering composed of three mashable application components: ngsi-source,

WMarket: Instance in the FIW/ Cloud of WMarket

© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

You can click on the name of the description to manage it as will be explained later.

To edit the details of your store, just click the **About & reviews** button (the one indicated by a blue arrow). Only store creators can edit its details. When you click this button, the following screen will be shown:



General information

Name: WStore

Website URL: https://store.lab.fiware.org

Image optional

Browser

Comment optional

Registration date: Jul 20, 2015

Save changes

User reviews

Aitor ★★★☆☆ Wed Jul 22 2015

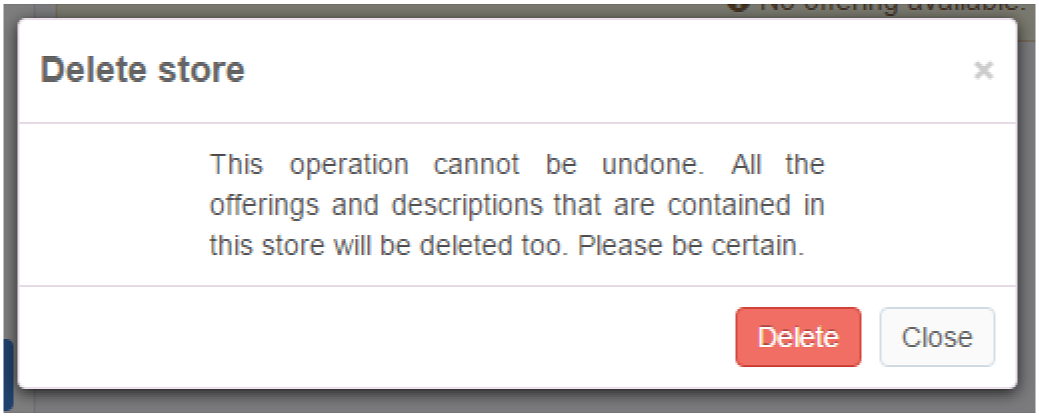
Example Comment

© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

GitHub · FIWARE

The area rounded by the red rectangle is the one that you should use to modify your store details. Introduce new details as desired and click the **Save changes** button when finished. The area rounded by the blue rectangle contains all the reviews that your store have received.

To delete your store, click the **Delete Store** button (the one indicated by the red arrow). Only the store creator can delete it. Before deleting the store, you have to confirm the operation in the following dialog:


Delete store ×

This operation cannot be undone. All the offerings and descriptions that are contained in this store will be deleted too. Please be certain.

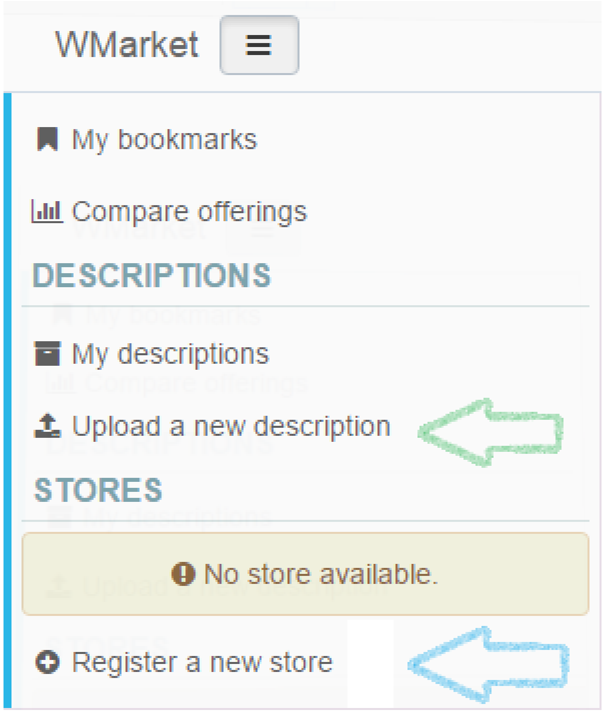
Delete | Close

Note: When you delete a store, all its descriptions, offerings and reviews will be deleted.

Descriptions Management

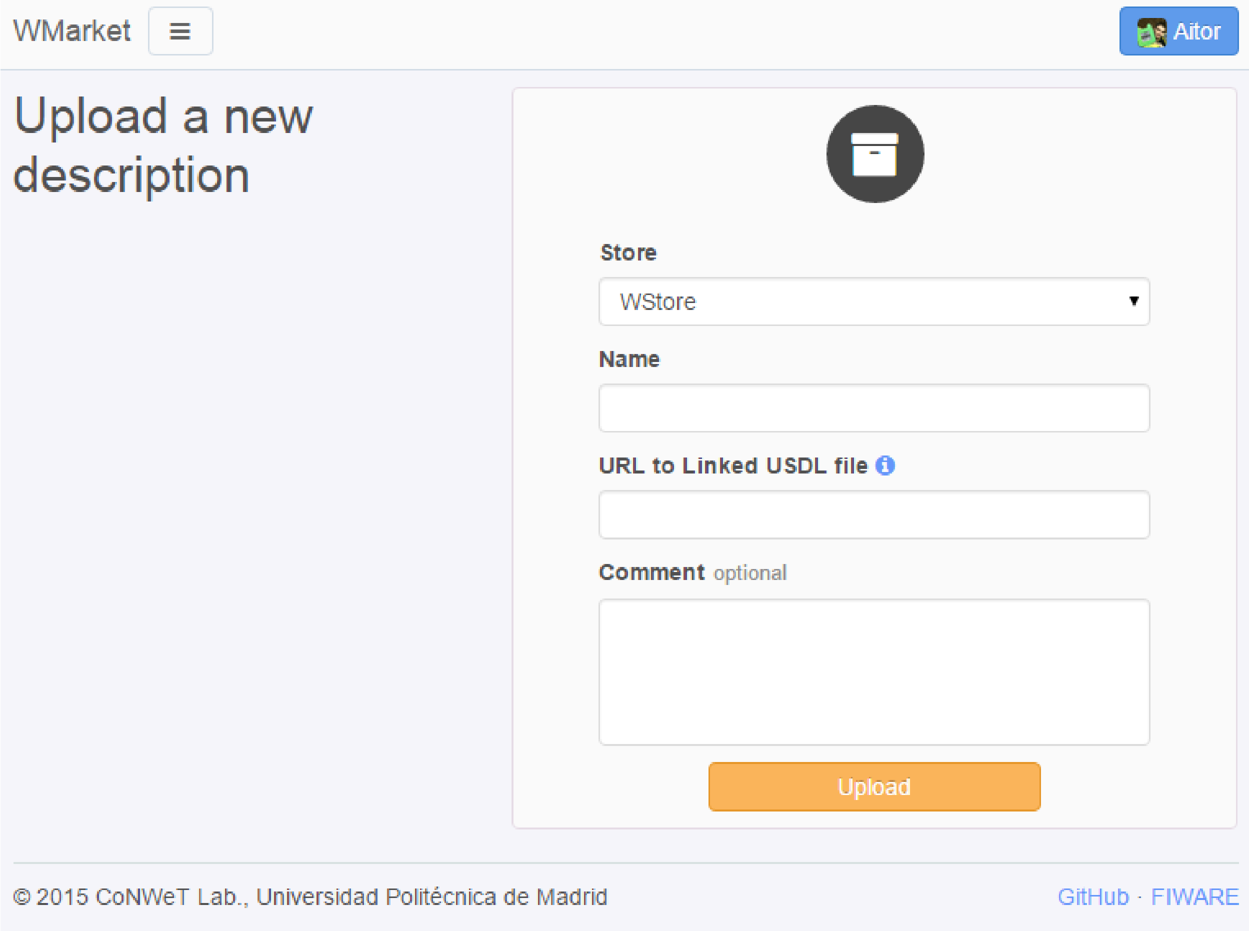
Descriptions are the definitions of your offerings. These descriptions are written in Linked USDL.

Creating Descriptions To register a new description, click in the menu button that is next to the **WMarket** text on the left upper corner (as can be seen in the following image) and click on the **Upload a new description** button (the one indicated by the green arrow).



No store available.

Once that you have clicked the button, you will be redirected to the following page:



Upload a new description

Store

WStore

Name

URL to Linked USDL file

Comment optional

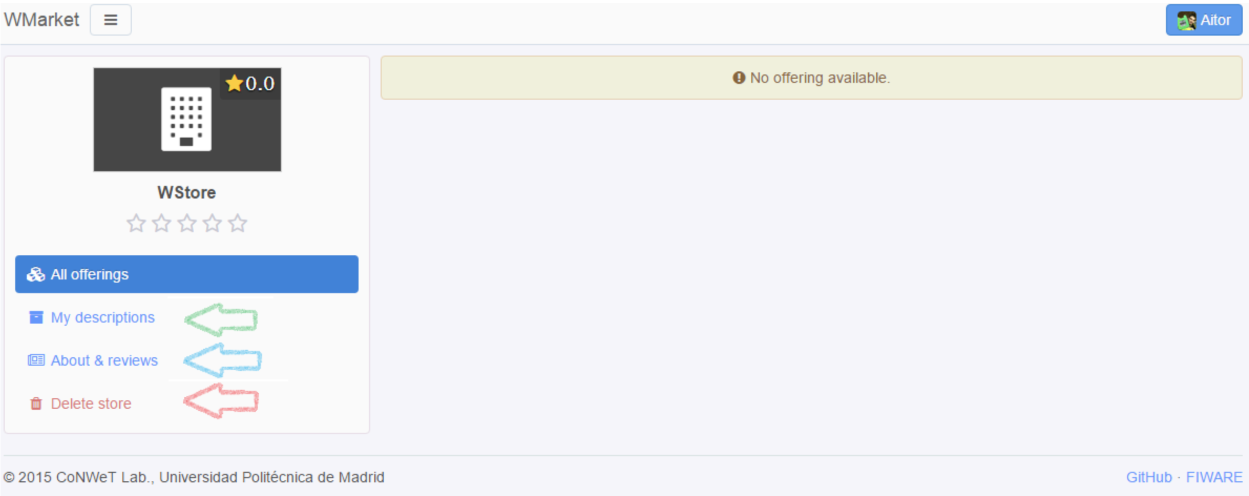
© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

To create the description, set the following details:

- **Store:** The store where you want to upload the description.
- **Name:** The name to give to your description. It must contain between 3 and 100 characters. Only letters, numbers, white spaces, dots and hyphens are accepted.
- **URL to Linked USDL file:** the HTTP URL where your description is hosted. This URL must exist and contain a valid Linked USDL file. Otherwise, the description creation will fail.
- **Comment:** A brief comment about the description you are uploading. You can type until 200 characters.

Note: If the description cannot be created because one or more fields are invalid, these fields will be highlighted in red colour and a message will be attached to every invalid field so the user can understand why these fields are invalid.

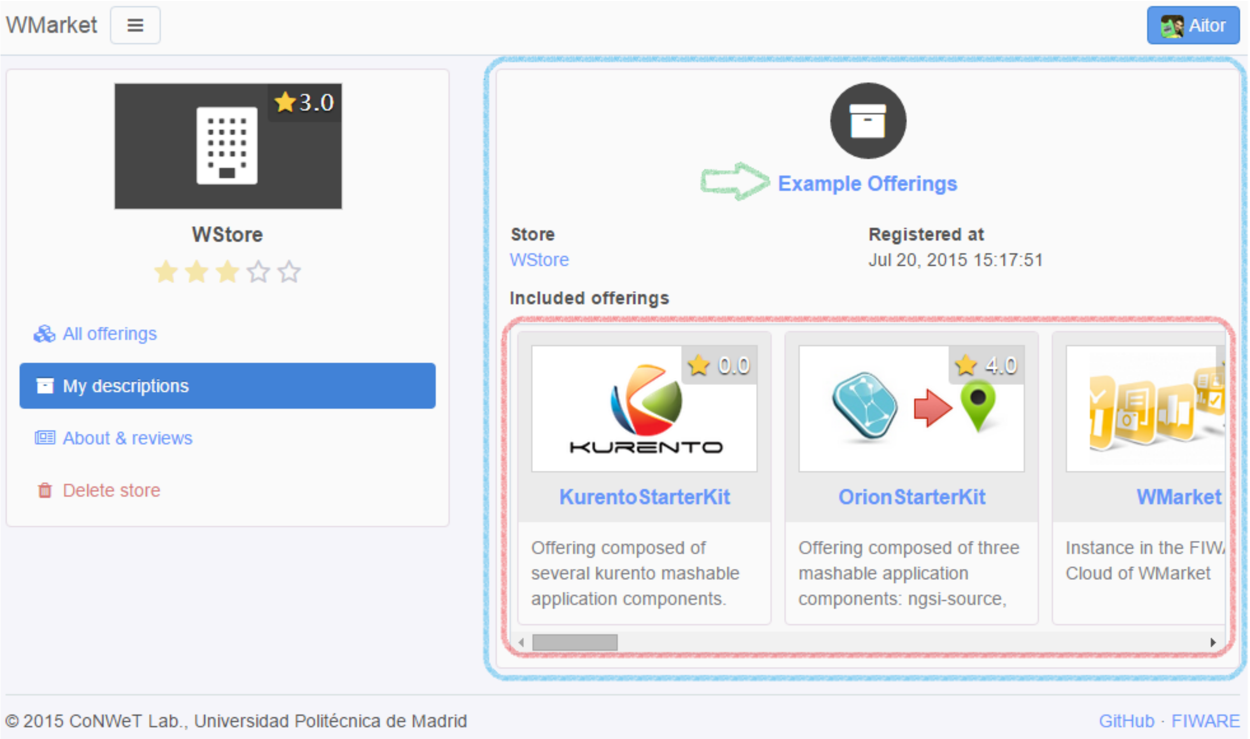
Once that you have completed the form, the description will be created and all the contained offerings will be loaded into the corresponding store and you will be redirected to the page that contains all the offerings of this store.



No offering available.

© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

Managing Existing Descriptions To manage all the descriptions that you have upload to this store, you can click the **My Descriptions** button (the one indicated by a green label). A screen similar to the following one will be shown:



Store: WStore

Registered at: Jul 20, 2015 15:17:51

Included offerings

Offering composed of several kurento mashable application components.

Offering composed of three mashable application components: ngsi-source,

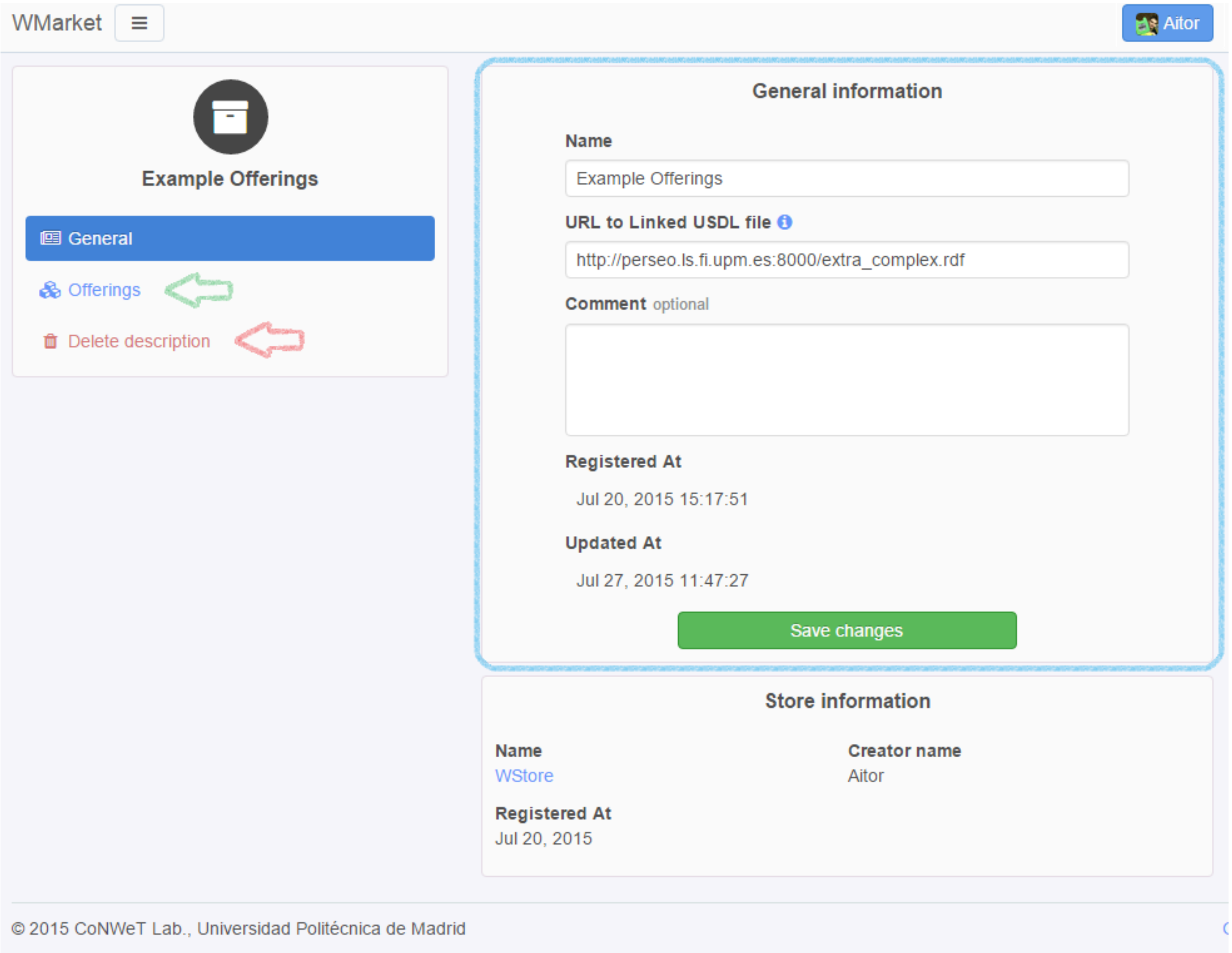
Instance in the FIW
Cloud of WMarket

© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

For each description a box similar to the blue one will appear. As you can see, the red box contains all the offerings attached to this description. To manage the description click on its name (the button indicated by the green arrow).

Note: If you have uploaded a description but you do not remember the Store that you have used to publish the offerings, you can view all the uploaded descriptions by opening the menu and clicking the **My Descriptions** button.

When you click on the name, you will see a page similar to the following one:



General information

Name

Example Offerings

URL to Linked USDL file

http://perseo.ls.fi.upm.es:8000/extra_complex.rdf

Comment optional

Registered At

Jul 20, 2015 15:17:51

Updated At

Jul 27, 2015 11:47:27

Store information

Name

WStore

Creator name

Aitor

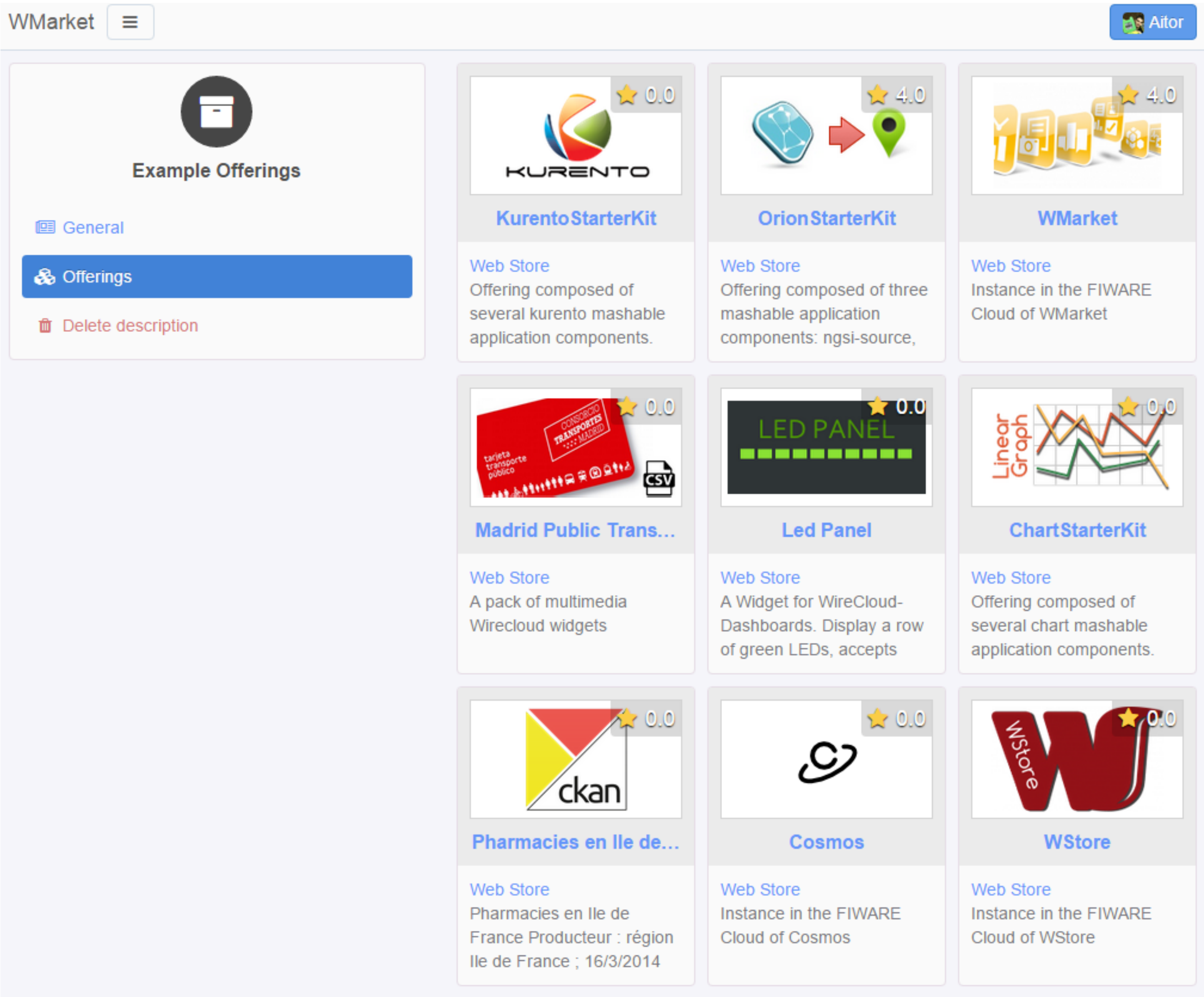
Registered At

Jul 20, 2015

© 2015 CoNWeT Lab., Universidad Politécnica de Madrid

If you want to edit the description details, you can use the form contained in the blue rectangle. Once that you have finished editing the description, click the **Save changes** button.

To view all the offerings contained in the offering, click the **Offerings** button (the one indicated by a green arrow). You will be displayed a screen similar to the following one:



Web Store
Offering composed of several kurento mashable application components.

Web Store
Offering composed of three mashable application components: ngsi-source,

Web Store
Instance in the FIWARE Cloud of WMarket

Web Store
A pack of multimedia Wirecloud widgets

Web Store
A Widget for WireCloud-Dashboards. Display a row of green LEDs, accepts

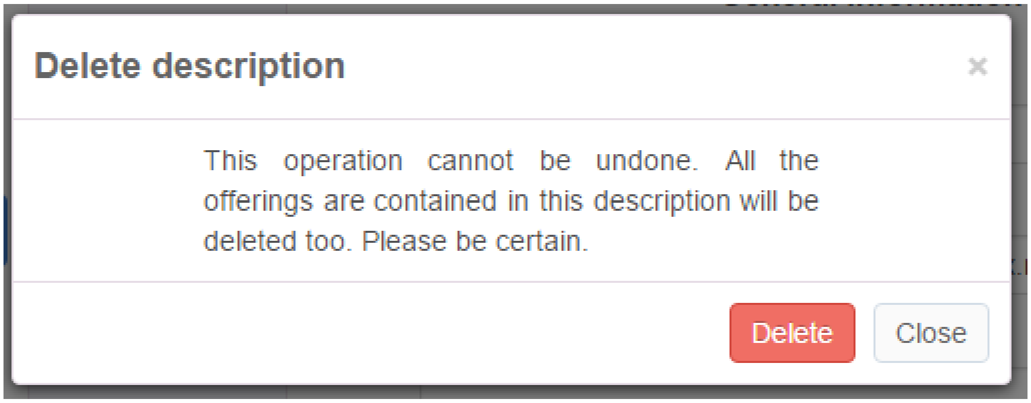
Web Store
Offering composed of several chart mashable application components.

Web Store
Pharmacies en Ile de France Producteur : région Ile de France ; 16/3/2014

Web Store
Instance in the FIWARE Cloud of Cosmos

Web Store
Instance in the FIWARE Cloud of WStore

Finally, if you want to delete a description, just click the **Delete description** button. Before deleting the description, you will be required to confirm the action with a dialog similar to the following one:



Delete description

This operation cannot be undone. All the offerings are contained in this description will be deleted too. Please be certain.

Note: By deleting a description, you will delete all the attached offerings (and its reviews or statistics).

Note: Descriptions are periodically updated to check whether new offerings have been included.

1.4.2 Programmer Guide

API

If you are not a normal user but a programmer, you may be interested in using the API to interact with the Marketplace, so you can build your own apps based on it. You can check the API Reference at:

- Apiary
- GitHub Pages

Please, note that all the requests against the Marketplace **must** be authenticated (except for the one to create users). Each instance can be configured to work with a different authentication system.

If the instance you are working with is using an external IdM, you must include the OAuth2 header to authenticate the user:

```
Authentication: Bearer OAUTH2_TOKEN
```

On the other hand, if the instance is using the local authentication system, you can use the Basic header to authenticate the user:

```
Authentication: Basic BASIC_TOKEN
```

where `BASIC_TOKEN` is a String that can be constructed as follows:

1. Username and password are combined into a string `username:password`. (ex: `aitor:1234`)
2. The resulting string is then encoded usign the RFC2045-MIME variant of Base64 (ex: `YWl0b3I6MTIzNA==`)

Offerings Descriptions (Linked USDL)

As stated before, Linked USDL files are those files used to describe the offerings published in WMarket. One Linked USDL file can contain one or more offerings. These files are written in RDF.

WMarket is fully integrated with WStore so you do not have to worry about creating Linked USDL files to upload your offerings to WMarket since WStore will create and upload them for you.

However, in some cases you can be interested in uploading your offerings to WMarket without using WStore. To do so, you have just to create the Linked USDL file that describe your offerings. This tutorial will guide you in creating these files in a very easy way.

Note: All the examples given in the following sections are written using the XML representation of RDF.

Structure

The first thing that you have to do is to create the main node of the file to include all the vocabularies that will be used to describe the different offerings. So, your header should be similar to the following one:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:usdl="http://www.linked-usdl.org/ns/usdl-core#"
    xmlns:pav="http://purl.org/pav/"
    xmlns:price="http://www.linked-usdl.org/ns/usdl-pricing#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:skos="http://www.w3.org/2004/02/skos/core#"
    xmlns:gr="http://purl.org/goodrelations/v1#">

    ...

</rdf:RDF>
```

Service Offerings Once that you have included all the required vocabularies, you are ready to create the offerings. As stated before, a Linked USDL can contain one or more offerings. Each offering is represented by a `usdl:ServiceOffering` as follows:

```
<usdl:ServiceOffering xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" rdf:about="http://repo
    <dcterms:title xml:lang="en">Input Box Widget</dcterms:title>
    <dcterms:description xml:lang="en">A simple widget with an input box.</dcterms:description>
    <pav:version>1.0</pav:version>
    <foaf:depiction rdf:resource="https://store.lab.fiware.org/media/CoNWeT__InputBoxWidget__1.0/cata
    <gr:availableDeliveryMethods rdf:resource="http://store.lab.fiware.org/offering/user/inputboxwidg
</usdl:ServiceOffering>
```

For each offering, you are required to include **at least** the following tags:

- `dcterms:title`: The title of your offering
- `dcters:description`: A brief description for your offering
- `pav:version`: The version of your offering
- `foaf:depiction`: The URL of the image that identifies your offering. This image will be displayed in the GUI
- `gr:availableDeliveryMethods`: The URL where consumer must be redirected to acquire the offering

However, this information can be not enough for WMarket consumers. For this reason you can also include extra nodes to specify the price plans and the services included in your offering.

Price Plans One Offering (`usdl:ServiceOffering`) can contain zero or more price plans. To include one price plan in your offering, just add the price plan tag (`usdl:hasPricePlan`) to your `usdl:ServiceOffering` as follow:

```
<usdl:ServiceOffering xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" rdf:about="http://repo
  ...
  <usdl:hasPricePlan rdf:nodeID="priceplan1"/>
</usdl:ServiceOffering>
```

As can be seen, the node refers to the specific price plan included in the offering. Each price plan is a node (`price:PricePlan`) that must be embedded in the same file and should look similar to the following one:

```
<price:PricePlan xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" rdf:nodeID="priceplan1">
    <dcterms:title xml:lang="en">Single Payment</dcterms:title>
    <dcterms:description xml:lang="en">This offering needs a single payment to be acquired</dcterms:
    <price:hasPriceComponent rdf:nodeID="pricecomponent1"/>
</price:PricePlan>
```

Price Plans (`price:PricePlan`) include zero (for free offerings) or more Price Components tagged as `price:hasPriceComponent`. As with Price Plans, these Price Components are nodes (`price:PriceComponent`) which must be embedded in the same file and follow this structure:

```
<price:PriceComponent rdf:nodeID="pricecomponent1">
    <rdfs:label xml:lang="en">Single payment</rdfs:label>
    <dcterms:description xml:lang="en">This component defines a single payment</dcterms:description>
    <price:hasPrice rdf:nodeID="pricespec1"/>
</price:PriceComponent>

<gr:PriceSpecification rdf:nodeID="pricespec1">
    <gr:hasCurrencyValue>1.0</gr:hasCurrencyValue>
    <gr:hasUnitOfMeasurement>single payment</gr:hasUnitOfMeasurement>
    <gr:hasCurrency>EUR</gr:hasCurrency>
</gr:PriceSpecification>
```

The name and description of the Price Component can be set in the `price:PriceComponent` node while the pricing details (currency, units and value) must be set in a separate node with type `gr:PriceSpecification` as stated in the given example. A Price Component **must** contain just one Price Specification.

Services An Offering (`usdl:ServiceOffering`) can also contain zero or more services. Services are the products (APIs, widgets, datasets, files, etc.) that you attach to the offering. To include one service in your offering, just add the service tag (`usdl:includes`) to your `usdl:ServiceOffering` as follow:

```
<usdl:ServiceOffering xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" rdf:about="http://repo
    ...
    <usdl:includes rdf:resource="http://repo.fiware/inputboxwidget#service1"/>
</usdl:ServiceOffering>
```

The node refers to the specific service included in the offering. Each service is a node (`usdl:Service`) that must be embedded in the same file. It will be similar to the following one:

```
<usdl:Service xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" rdf:about="http://repo.fiware/
    <dcterms:title xml:lang="en">Input Box Widget</dcterms:title>
    <dcterms:description xml:lang="en">A widget with a simple input box that can be use to request d
    <usdl:hasClassification rdf:nodeID="classification1" />
</usdl:Service>
```

As can be seen, we can set the name and the description of the service. Additionally, we can also specify a set of categories that can be used to categorize and identify the service easily. To to it, just add the node `usdl:hasClassification` that will refer to a category (`skos:Concept`) embedded in the same file. Use the node `rdfs:label` to set the name of the category.

```
<skos:Concept xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" rdf:nodeID="classification1">
    <rdfs:label>Wirecloud widget</rdfs:label>
</skos:Concept>
```

Note: Offerings are categorized based on the categories of their services.

Example

You can find the previous example `here`. This example is very simple. Please, keep in mind that:

- A description file can contain more than one offering
- An offering can contain more than one price plan
- A price plan can contain more than one price component
- An offering can contain more than one service
- More than one category can be attached to a service