1. Overview

The loan portal is available for the Open Source community under the GNU Lesser General Public License (LGPL) version 2.

The source code of Loan portal is available for the entire community, which is free to use, modify and redistribute it on the premise of such license.

For administrators and web developers alike, there are some important bits of information you should familiarize yourself with before starting out. This document serves as a brief introduction to some of the concepts and terminology behind the loan portal. As well, where to go when you need help.

2. Downloading binary (zip file)

Download the latest version of the loan portal from https://villagepowerltd.github.io/oss-public/

3. Installation (turning zip file into a running application)

System Requirements

The loan portal is an entirely web-based application. Once installed and connected to the Internet or your private network, you can easily use the program with a browser without having to install additional software.

Hardware requirements may vary based on the number of concurrent users, repository size and system configuration. Larger implementations may require some configuration tuning to perform optimally. It is highly recommended that the hardware be dedicated to running the loan portal Installation.

The a minimum server infrastructure specification are listed below. After a successful installation, all you will need to connect to your loan portal account is a modern Internet browser **(see 'Supported Internet Browsers').**

Minimum Hardware Requirements	
Server	2.4 GHz 32-bit (x86) or 64-bit (x64) dual core processor 6 GB of system memory 5GB of storage for the application data and repository storage as needed.
Minimum Software Requirements	
Loan portal web files hosting	Red Hat Enterprise Linux 7 (32-bit and 64-bit) SUSE Linux Enterprise Server 10 (32-bit and 64-bit) CentOS Linux 6 (32-bit and 64-bit) Ubuntu 14 (32-bit and 64-bit) Microsoft Windows 2008 Server (32-bit and 64-bit) MAC OS X Lion (OS X 10.8) Apache Wamp Xamp

Supported Browsers	Firefox 40 Internet Explorer 11 Edge 25 Google Chrome 40 Safari 6

Installation

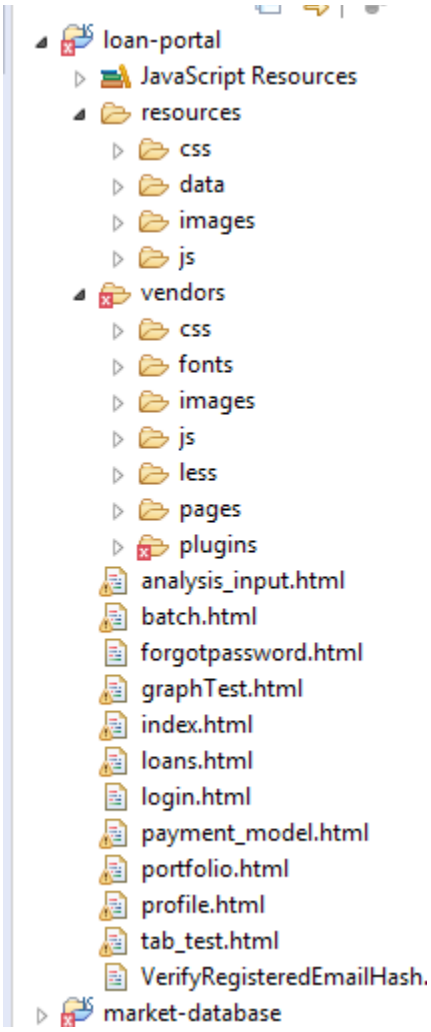
Download the loan portal bundle and uncompress on your file system disk (a good option is to uncompress on *C:*\). If you experience problems uncompressing **loan-portal-bundle.zip**,try using the WinRAR application to uncompress it. Once uncompressed, copy and paste the web files to the web server. **Eg (wamp/www/ or var/www/html)**

4. Set up

It is very easy to setup and start using the loan portal. This section will explain how to setup the loan portal after download. We will also discuss the loan portal file structure, and demonstrate its usage with an example.

File structure

You will see the following file/directory structure the uncompressed **loan-portal-bundle.zip.**



There are CSS and JS (bootstrap.*), as well as compiled and minified CSS and JS (bootstrap.min.*). Fonts from Glyphicons are included in the loan portal directory.

The files under resources/js, css/, data/ ,images/are the customised source code for loan portal JS, CSS,test data and images (respectively).

The files under vendors/js, css/,images/ and fonts/ are the source code for Bootstrap CSS, JS, and icon fonts (respectively).

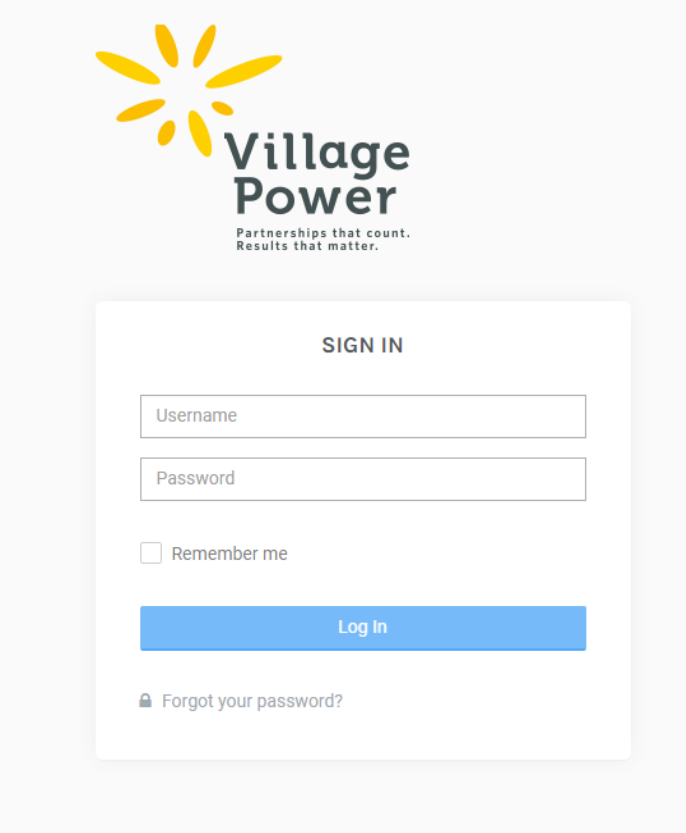
The loan-portal/ folder includes everything listed in the uncompressed download section above and all *.html files.

The loan-portal contants.js (resources/js/constants.js) file that contains all CONSTANTS that point **url** (end points) API. You will need to modify the **SERVER_PATH** constant and set the IP address to the server hosting the API.

```
var SERVER_PATH='http://localhost:9090/loa-1.0-SNAPSHOT/';

var CREATE_BATCH=SERVER_PATH+'webresources/batch/createBatch';

var UPDATE_BATCH=SERVER_PATH+'webresources/portfolio/updateBatch';

var DELETE_BATCH=SERVER_PATH+'webresources/portfolio/deleteBatch';

var RETRIEVE_BATCH=SERVER_PATH+'webresources/batch/allBatches';

var RETRIEVE_BATCH_PORTFOLIO=SERVER_PATH+'webresources/portfolio/allBatchesInPortfolio';
```

5. **log in**

In order to log on to the system, you have to point your browser to the URL where the loan portal is installed (this may be an address on your network) eg http://localhost/loan-portal/ . Once there, a login form is presented in which you have to enter your username, password. On this screen you can also choose to save the credentials on your local machine for quicker access in the future. The initial administrator login credentials are pre-installed but subject to change on the first login.



If you forgot your credentials, please click on the “Forgot password” link, and an email will be sent guiding you through the password reset process.

Once logged in, a desktop is displayed with the following four main areas:

- The title bar with the application logo
- The main navigation menu
- The work area panel

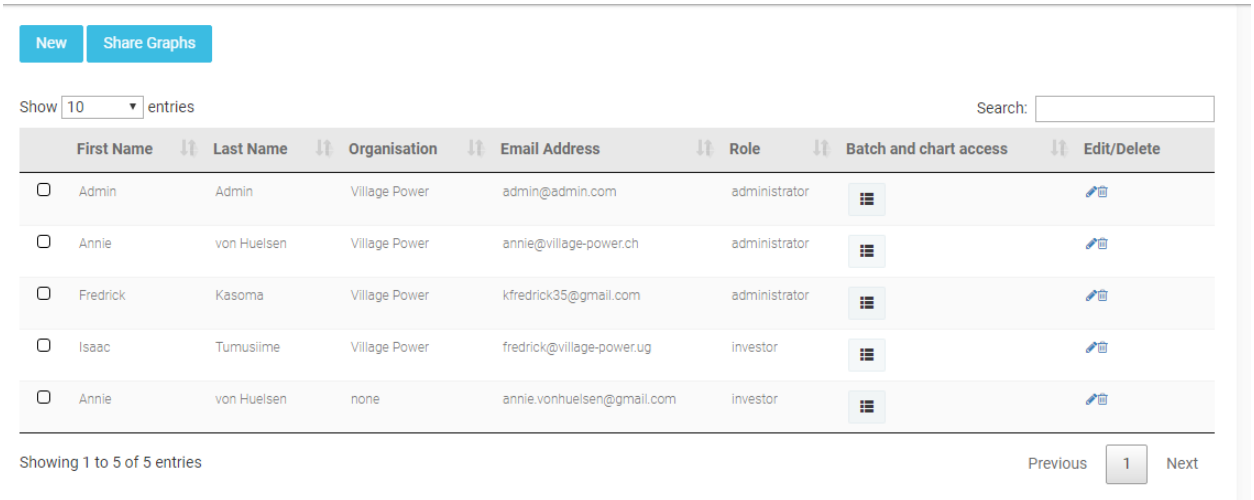
Main navigation menu

The main navigation menu contains the items needed to access the systems functionality.

6. Setting up user profiles

Inside the User profile section you can setup all the users accounts while setting their different roles on the system.

This panel shows the list of all users currently existing into the system.



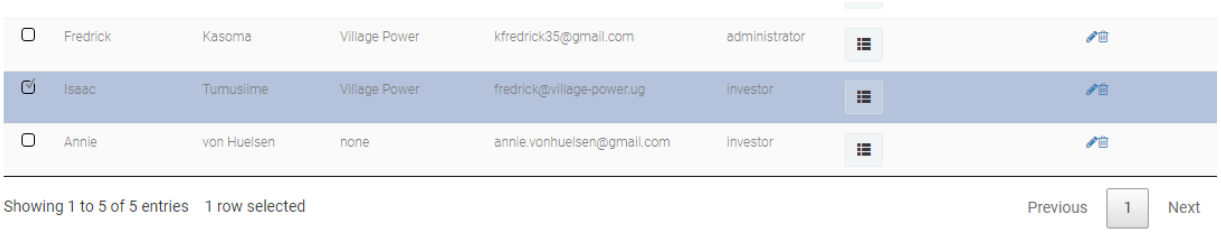
To add a new user, you have to click on **New** button and then fill all the required data:

- First Name
- Last Name
- Email
- Organisation
- User role

Each user can belong to one organisation and one have one role. The security policies are always expressed in relation to the user role and company specific users.

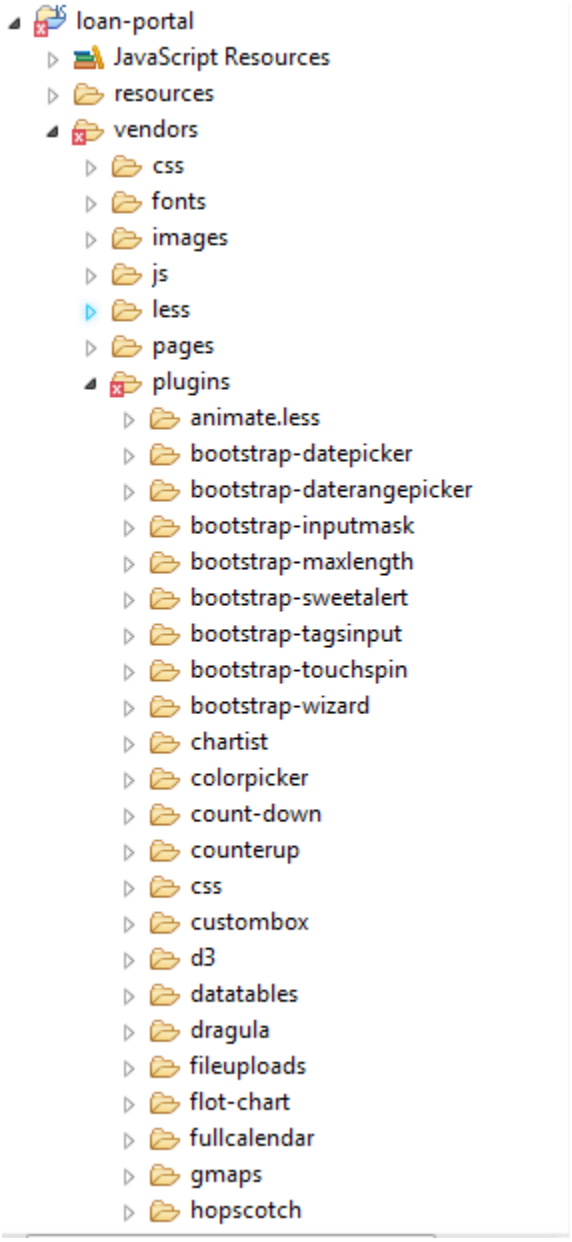
When you click the Save button (assuming all the required information have been correctly entered), a new user is created and added to the database. Then the user receives a welcome email containing the username and a link guiding on how to set a password.

By selecting a user item you can see all the user details under the list highlighted. Here you can edit the item's data.



7. Plug in installation

The loan portal builds on top bootstrap and JQuery platform. All Bootstrap and JQuery plugins come pre-installed in uncompressed folders. This means you will not need to install any external plugins. You will see the following file/directory structure the uncompressed **loan-portal-bundle.zip.**



Bootstrap documentation can be accessed from **http://getbootstrap.com/getting-started/**

8. Customizing the logo

You can customize the logo for the loan portal that appears on the login page. the loan portal provides a default one, which is the recognizable village power logo. Using the default logo is fine but if you really want to make your site stand out, you should provide your own.

Navigate to resources/ images/ where there are customized images for the loan portal and replace logo1-1-1.png. The changes may not appear immediately in your browser, so completely clear your browser's cache and reload the page. Also, if you've bookmarked your site, you may need to delete the bookmark and then create it again so that the new logo is used instead.

9. Setting up the API

The loan portal uses a built-in Web Services as a middleware which can also be integrated with other systems. The Web Service module(API) is part of the loan portal core distribution and is compliant with W3C specifications **RESTful web services**. Using **RESTful web services** over HTTP allows for easier communication through proxies and firewalls than previous remote execution technology.

REST specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web. Data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs), typically links on the Web. The resources are acted upon by using a set of simple, well-defined operations.

The REST architectural style constrains an architecture to a client/server architecture and is designed to use a stateless communication protocol, typically HTTP. Clients and servers exchange representations of resources by using a standardized interface and protocol.

All end points to the API on the loan portal are exposed in one file http://localhost/loan-portal/resources/js/constants.js. You can find more information about each endpoint on http://localhost/loan-portal/swaga.doc

Note: The access path to the API is protected with basic authentication, to be able to run the endpoints you must log in using the credentials of a loan portal user. A successful login is assigned an authorization token ID that is later passed to the request. This token is stored as a cookie (cookie name: loginToken).

The loan portal uses ajax to send requests and receive responses to and from the endpoints respectively.

Here is an example to retrieve all batches registered on the loan portal through an ajax call:

```
.ajax({
    headers : {
        'Accept' : 'application/json',
        'Content-Type' : 'application/json',
    'Authorization': $.cookie('loginToken')
    },
    type : "GET",
    url : 'http://localhost:9090/loa-1.0-SNAPSHOT/webresources/batch/allBatches',
    dataType : "json",
    success : function(data, status) {
        // code executed on success
    },
    error : function(jqXHR, textStatus,
        //code executed on failure
    }
                });
```

Here is an example to save batch on the loan portal.

```
19
20      var $batchName = $('#batch_name');
21      var $batchDescription = $('#batch_description');
22
23      var batch = {
24          "type" : "object",
25          "batch" : {
26              name : $batchName.val(),
27              description : $batchDescription.val()
28          }
29      };
30
31      var batchJSONString = JSON.stringify(batch);
32
33
34
35      $.ajax({
36          headers : {
37              'Accept' : 'application/json',
38              'Content-Type' : 'application/json',
39          'Authorization': $.cookie('loginToken')
40          },
41          type : "POST",
42          url : 'http://localhost:9090/loa-1.0-SNAPSHOT/webresources/batch/createBatch',,
43          data : batchJSONString,
44          dataType : "json",
45          success : function(data, status) {
46          // code executed on success
47
48          },
49          error : function(jqXHR, textStatus,
50                  errorThrown) {
51          // code executed on failure
52          }
53      });
54
55
```