@Author: Vaibhav Khopade

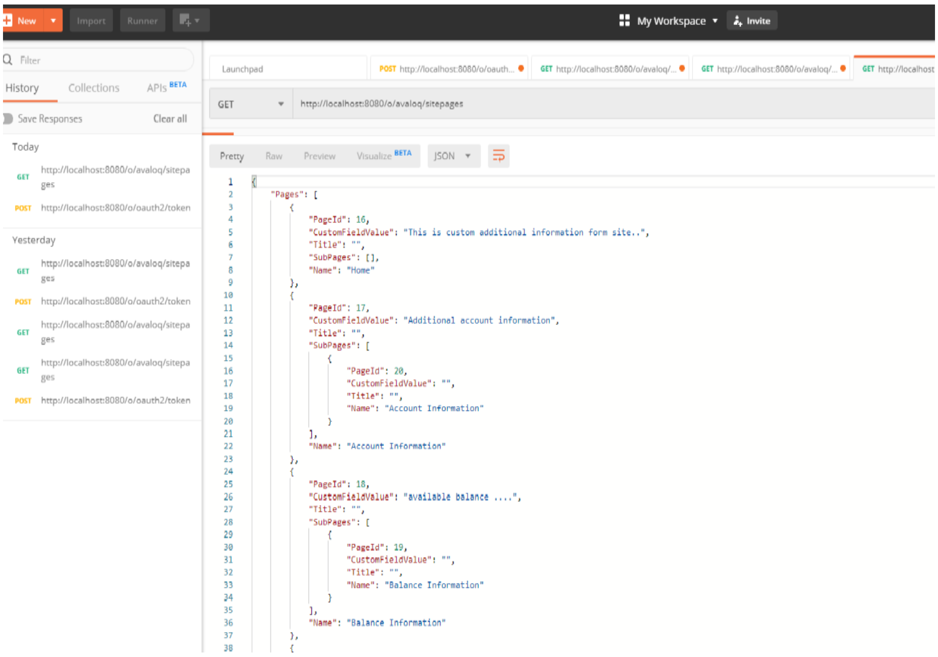
Note: All Avaloq test Assignments are developed using Liferay DXP 7.2 (CE).

## Assignment 1:

### 'AvaloqRestPortlet' Module:

This rest portlet liferay osgi module is developed to receive http rest calls which will return page hierarchy of configured Site. This rest call returns with JSON formatted page properties like name,title,id,custom field and list of child pages (sub pages)with their details.

Following screenshot shows the JSON output with site page hierarchy:



GET http://localhost:8080/o/avaloq/sitepages

Today
- GET http://localhost:8080/o/avaloq/sitepages
- POST http://localhost:8080/o/oauth2/token

Yesterday
- GET http://localhost:8080/o/avaloq/sitepages
- POST http://localhost:8080/o/oauth2/token
- GET http://localhost:8080/o/avaloq/sitepages
- GET http://localhost:8080/o/avaloq/sitepages
- POST http://localhost:8080/o/oauth2/token

```
{
    "Pages": [
        {
            "PageId": 16,
            "CustomFieldValue": "This is custom additional information form site..",
            "Title": "",
            "SubPages": [],
            "Name": "Home"
        },
        {
            "PageId": 17,
            "CustomFieldValue": "Additional account information",
            "Title": "",
            "SubPages": [
                {
                    "PageId": 20,
                    "CustomFieldValue": "",
                    "Title": "",
                    "Name": "Account Information"
                }
            ],
            "Name": "Account Information"
        },
        {
            "PageId": 18,
            "CustomFieldValue": "available balance ....",
            "Title": "",
            "SubPages": [
                {
                    "PageId": 19,
                    "CustomFieldValue": "",
                    "Title": "",
                    "Name": "Balance Information"
                }
            ],
            "Name": "Balance Information"
        },
        {
```

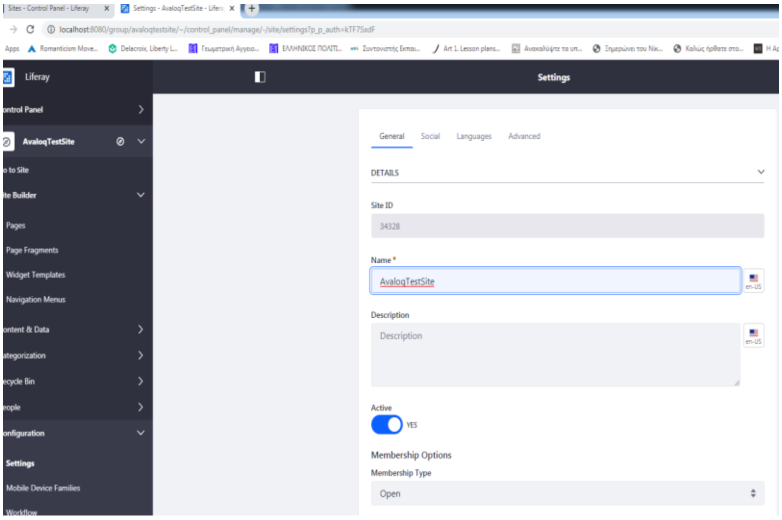
### Steps for Deployment of 'AvaloqRestPortlet' Module:

1) Create a Site from control panel

2) Copy generated Site Id and Modify 'ConstantsKeys.java' class .Replace value of SITE_GROUP_ID with your generated long group id.

e.g. public static final long SITE_GROUP_ID = **34328l**;

Following screenshot of Site with its ID.



Site ID
34328

Name *
AvaloqTestSite

Description
Description

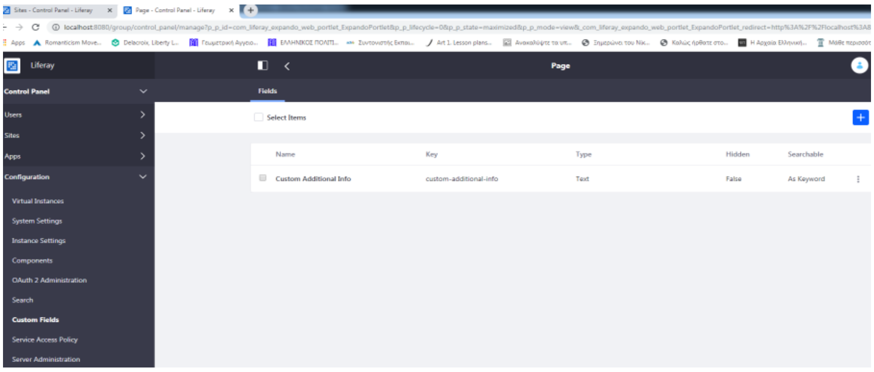
Active
YES

Membership Options

Membership Type
Open

3) Create Custom field on Page with key ' custom-additional-info'.

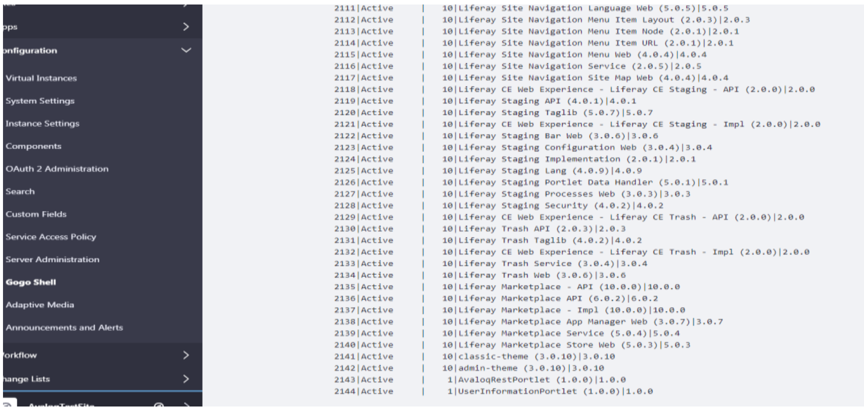
e.g.



| Name | Key | Type | Hidden | Searchable |
|---|---|---|---|---|
| Custom Additional Info | custom-additional-info | Text | False | As Keyword |

4) Deploy portlet in osgi on Liferay dxp 7.2.

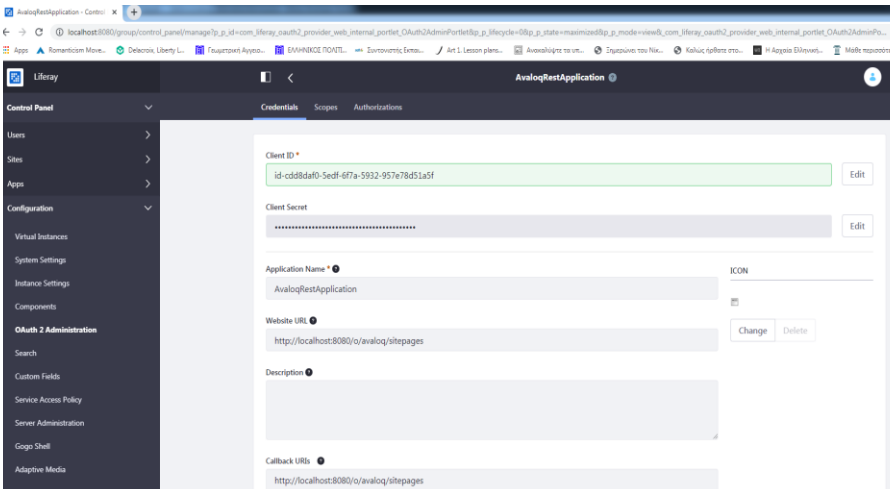
E.g.: Following Screenshot of GoGo Shell shows successfully deployment of this module.



```
2111|Active     |   10|Liferay Site Navigation Language Web (5.0.5)|5.0.5
2112|Active     |   10|Liferay Site Navigation Menu Item Layout (2.0.3)|2.0.3
2113|Active     |   10|Liferay Site Navigation Menu Item Node (2.0.1)|2.0.1
2114|Active     |   10|Liferay Site Navigation Menu Item URL (2.0.1)|2.0.1
2115|Active     |   10|Liferay Site Navigation Menu Web (4.0.4)|4.0.4
2116|Active     |   10|Liferay Site Navigation Service (2.0.5)|2.0.5
2117|Active     |   10|Liferay Site Navigation Site Map Web (4.0.4)|4.0.4
2118|Active     |   10|Liferay CE Web Experience - Liferay CE Staging - API (2.0.0)|2.0.0
2119|Active     |   10|Liferay Staging API (4.0.1)|4.0.1
2120|Active     |   10|Liferay Staging Taglib (5.0.7)|5.0.7
2121|Active     |   10|Liferay CE Web Experience - Liferay CE Staging - Impl (2.0.0)|2.0.0
2122|Active     |   10|Liferay Staging Bar Web (3.0.6)|3.0.6
2123|Active     |   10|Liferay Staging Configuration Web (3.0.4)|3.0.4
2124|Active     |   10|Liferay Staging Implementation (2.0.1)|2.0.1
2125|Active     |   10|Liferay Staging Lang (4.0.9)|4.0.9
2126|Active     |   10|Liferay Staging Portlet Data Handler (5.0.1)|5.0.1
2127|Active     |   10|Liferay Staging Processes Web (3.0.3)|3.0.3
2128|Active     |   10|Liferay Staging Security (4.0.2)|4.0.2
2129|Active     |   10|Liferay CE Web Experience - Liferay CE Trash - API (2.0.0)|2.0.0
2130|Active     |   10|Liferay Trash API (2.0.3)|2.0.3
2131|Active     |   10|Liferay Trash Taglib (4.0.2)|4.0.2
2132|Active     |   10|Liferay CE Web Experience - Liferay CE Trash - Impl (2.0.0)|2.0.0
2133|Active     |   10|Liferay Trash Service (3.0.4)|3.0.4
2134|Active     |   10|Liferay Trash Web (3.0.6)|3.0.6
2135|Active     |   10|Liferay Marketplace - API (10.0.0)|10.0.0
2136|Active     |   10|Liferay Marketplace API (6.0.2)|6.0.2
2137|Active     |   10|Liferay Marketplace - Impl (10.0.0)|10.0.0
2138|Active     |   10|Liferay Marketplace App Manager Web (3.0.7)|3.0.7
2139|Active     |   10|Liferay Marketplace Service (5.0.4)|5.0.4
2140|Active     |   10|Liferay Marketplace Store Web (5.0.3)|5.0.3
2141|Active     |   10|classic-theme (3.0.10)|3.0.10
2142|Active     |   10|admin-theme (3.0.10)|3.0.10
2143|Active     |    1|AvaloqRestPortlet (1.0.0)|1.0.0
2144|Active     |    1|UserInformationPortlet (1.0.0)|1.0.0
```

5) Create 'AvaloqRestApplication'  from control panel/configuration/ OAuth 2 Administration.

e.g.



Client ID *
id-cdd8daf0-5edf-6f7a-5932-957e78d51a5f

Client Secret
••••••••••••••••••••••••••••••••••••••••••••

Application Name *
AvaloqRestApplication

Website URL
http://localhost:8080/o/avaloq/sitepages

Description

Callback URIs
http://localhost:8080/o/avaloq/sitepages

**Assignment 2:**

Assumption Made:

It is assumed that this rest point will be accessed by applications outside Liferay DXP. So no logged in user authorization is taken under consideration for the time being.

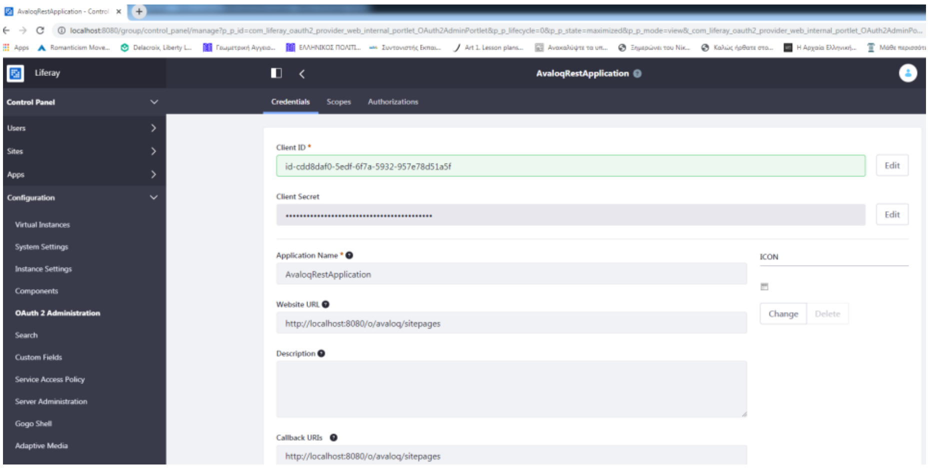
If logged in user authorization is requirement then it can be easily implement by adding Authorization Filters to this end point which can check if user is logged in and user has proper roles to access this end point. Further deeper permission checks filters can be applied on access rights for pages.

1) **End Point Security Permissions:**

This end point is protected by OAUTH 2.0 authorization.

So OAuth 2.0 application is created to provide access to rest service. For testing purpose 'Headless Server profile' authorization type is used. There are many other options available for production as per OAuth2.0 specifications.

Following screenshot shows the configuration of 'AvaloqRestApplication' where client-id and client-secret is generated for authorization.



Client ID *
id-cdd8daf0-5edf-6f7a-5932-957e78d51a5f

Client Secret
••••••••••••••••••••••••••••••••••••••••••••

Application Name *
AvaloqRestApplication

Website URL
http://localhost:8080/o/avaloq/sitepages

Description

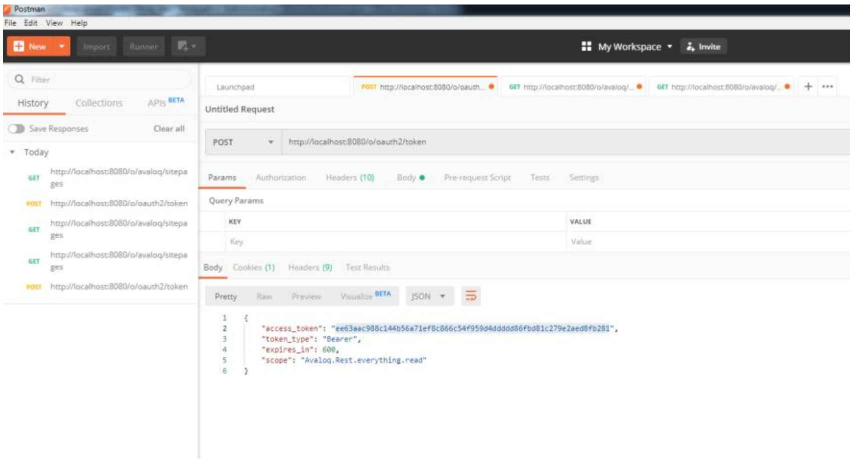
Callback URIs
http://localhost:8080/o/avaloq/sitepages

Getting access:

1) Client can get access token by curl with client-id and client-secret.

e.g.

curl http://localhost:8080/o/oauth2/token -d 'grant_type=client_credentials&client_id=id-cdd8daf0-5edf-6f7a-5932-957e78d51a5f&client_secret=secret-5053cd5c-e79b-74f4-2ac7-316cf18b862'


Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Filter
History
Collections
APIs BETA
Save Responses
Clear all
Today
GET http://localhost:8080/o/avaloq/sitepages
POST http://localhost:8080/o/oauth2/token
GET http://localhost:8080/o/avaloq/sitepages
GET http://localhost:8080/o/avaloq/sitepages
POST http://localhost:8080/o/oauth2/token
Launchpad
POST http://localhost:8080/o/oauth...
GET http://localhost:8080/o/avaloq/...
GET http://localhost:8080/o/avaloq/...
Untitled Request
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
Query Params
KEY
VALUE
Key
Value
Body
Cookies (1)
Headers (9)
Test Results
Pretty
Raw
Preview
Visualize BETA
JSON

POST http://localhost:8080/o/oauth2/token

```
{
    "access_token": "ee63aac988c144b56a71ef8c866c54f959d4ddddd86fbd81c279e2aed8fb281",
    "token_type": "Bearer",
    "expires_in": 600,
    "scope": "Avaloq.Rest.everything.read"
}
```

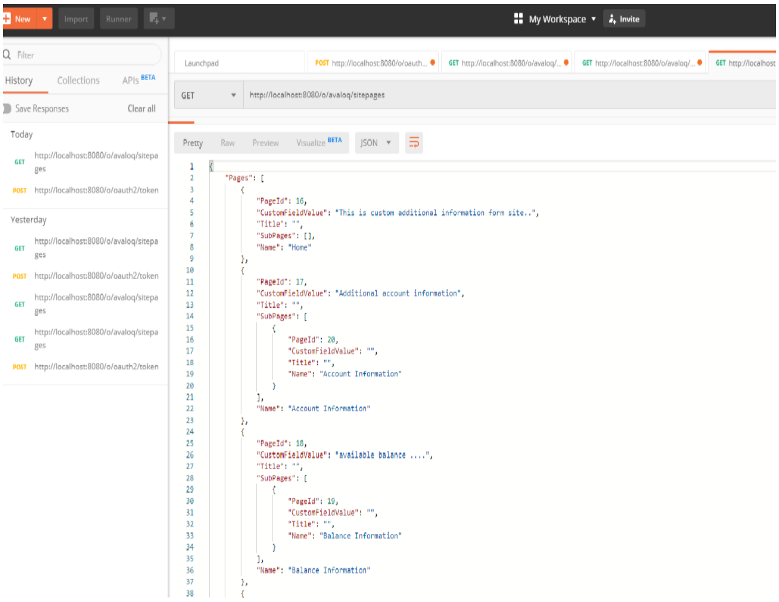
2) Make a call to end point using generated access_token .

e.g. access_token: "ee63aac988c144b56a71ef8c866c54f959d4ddddd86fbd81c279e2aed8fb281"

Making call to end ' http://localhost:8080/o/avaloq/sitepages':

curl --header "Authorization: Bearer ee63aac988c144b56a71ef8c866c54f959d4ddddd86fbd81c279e2aed8fb281" http://localhost:8080/o/avaloq/sitepages

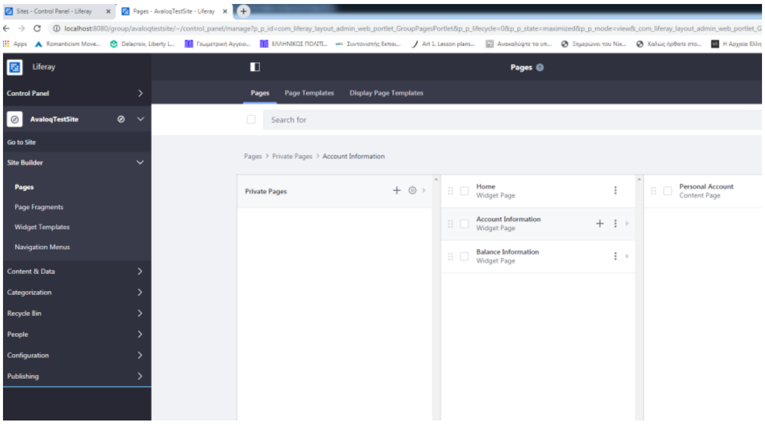
Following screenshot shows the json output with site page hierarchy.


New
Import
Runner
My Workspace
Invite
Filter
History
Collections
APIs BETA
Save Responses
Clear all
Today
GET http://localhost:8080/o/avaloq/sitepages
POST http://localhost:8080/o/oauth2/token
Yesterday
GET http://localhost:8080/o/avaloq/sitepages
POST http://localhost:8080/o/oauth2/token
GET http://localhost:8080/o/avaloq/sitepages
GET http://localhost:8080/o/avaloq/sitepages
POST http://localhost:8080/o/oauth2/token
Launchpad
POST http://localhost:8080/o/oauth...
GET http://localhost:8080/o/avaloq/...
GET http://localhost:8080/o/avaloq/...
GET http://localhost
Pretty
Raw
Preview
Visualize BETA
JSON

GET http://localhost:8080/o/avaloq/sitepages

```
{
    "Pages": [
        {
            "PageId": 16,
            "CustomFieldValue": "This is custom additional information form site..",
            "Title": "",
            "SubPages": [],
            "Name": "Home"
        },
        {
            "PageId": 17,
            "CustomFieldValue": "Additional account information",
            "Title": "",
            "SubPages": [
                {
                    "PageId": 20,
                    "CustomFieldValue": "",
                    "Title": "",
                    "Name": "Account Information"
                }
            ],
            "Name": "Account Information"
        },
        {
            "PageId": 18,
            "CustomFieldValue": "available balance ....",
            "Title": "",
            "SubPages": [
                {
                    "PageId": 19,
                    "CustomFieldValue": "",
                    "Title": "",
                    "Name": "Balance Information"
                }
            ],
            "Name": "Balance Information"
        },
        {
```

Following screenshot shows  page level setting in AvaloqTestSite .



## 2) OSGi Significance In 'AvaloqRestPortlet':

'AvaloqRestPortlet' Module is created and deployed as liferay OSGi module.

Liferay OSGi framework takes cares of all modules dependencies and they are automatically managed by container. So this module can be installed,started,updated,stopped and uninstall without stopping running Liferay Portal. OSGi will hide all module classes. OSGi can handle different version of same module.Also Liferay service classes(Interfaces) are available to our module since we declare them in gradle file as dependency modules.

e.g. OSGi Configuration file:  bnd.bnd

```
Bundle-Name: AvaloqRestPortlet
Bundle-SymbolicName: com.avaloq.rest
Bundle-Version: 1.0.0
Liferay-Configuration-Path: /configuration
Export-Package:com.avaloq.rest.service.interfaces
```

'com.avaloq.rest.service.interfaces' is exported and made available to other modules as provider. So other modules if they requires PageInformation service then they just need to declare Interface as @reference in their classes. Actual implementation is handle by 'AvaloqRestPortlet' module . This way it is loosely coupled and managed easily.

e.g. PageInformationService.java implements PageInformationInteface.java in exported package   which has Annotation @Component and immediate=true means module is not lazy-loaded and service registry loads it as soon as its deployed.

**Assignment 3:**

**'UserInformationPortlet':**

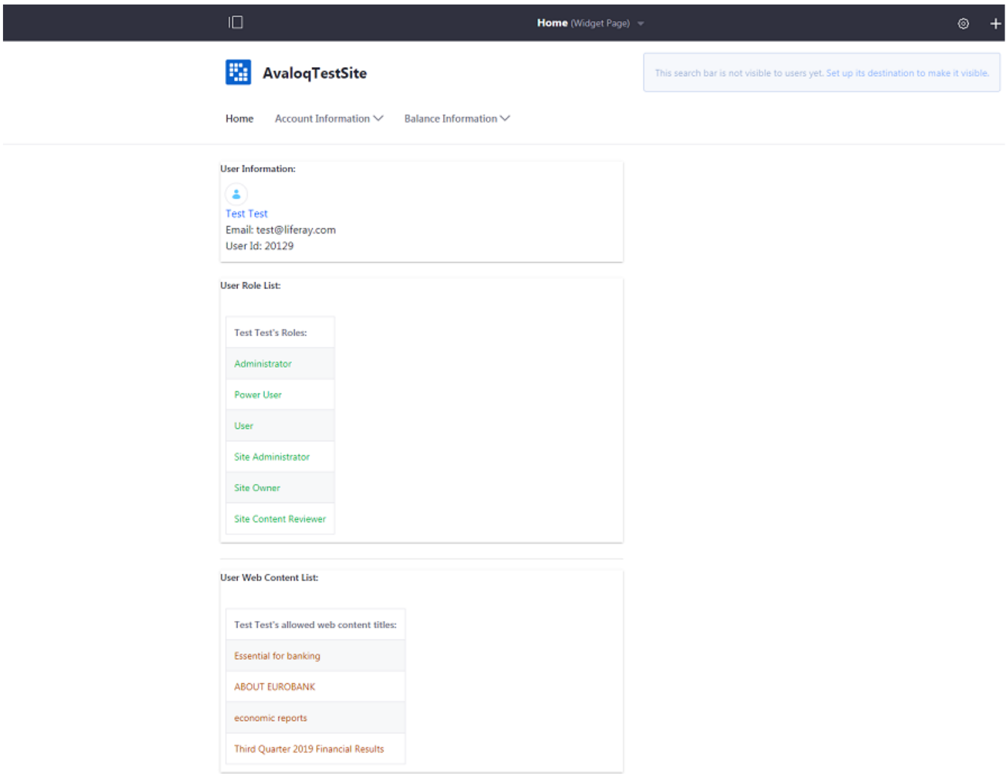
This portlet module is developed to show user information like roles,userid,username,email and web content titles which user has access.

Portlet is created as Liferay MVC portlet.

Different site members can be created with different roles . Each web content need to set permissions on  user group .

E.g. Following are screenshots of different users with different roles like 'Site Members','Site owner' ,Administrator etc.

1) Test Test has Site Administrator role


Home (Widget Page)
AvaloqTestSite
This search bar is not visible to users yet. Set up its destination to make it visible.
Home
Account Information
Balance Information

User Information:

Test Test
Email: test@liferay.com
User Id: 20129

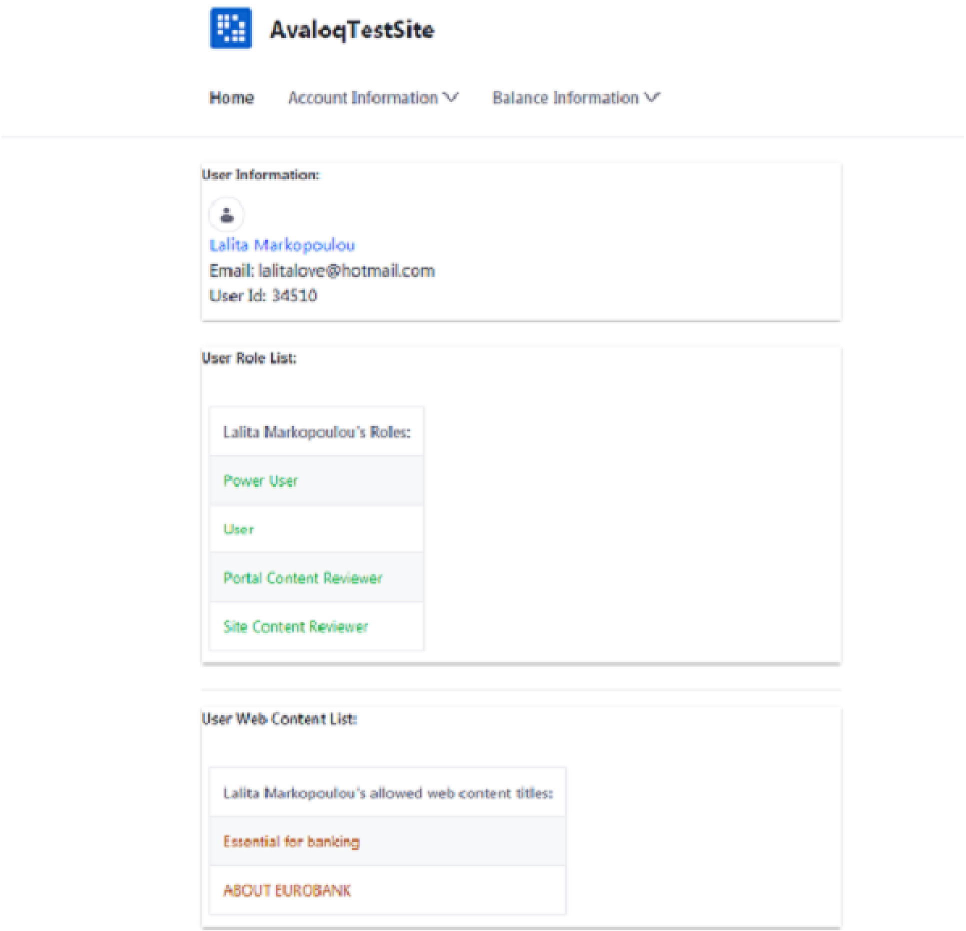
User Role List:

| Test Test's Roles: |
|---|
| Administrator |
| Power User |
| User |
| Site Administrator |
| Site Owner |
| Site Content Reviewer |

User Web Content List:

| Test Test's allowed web content titles: |
|---|
| Essential for banking |
| ABOUT EUROBANK |
| economic reports |
| Third Quarter 2019 Financial Results |

2) Lalita Markopoulu with 000.



User Information:

Lalita Markopoulou
Email: lalitalove@hotmail.com
User Id: 34510

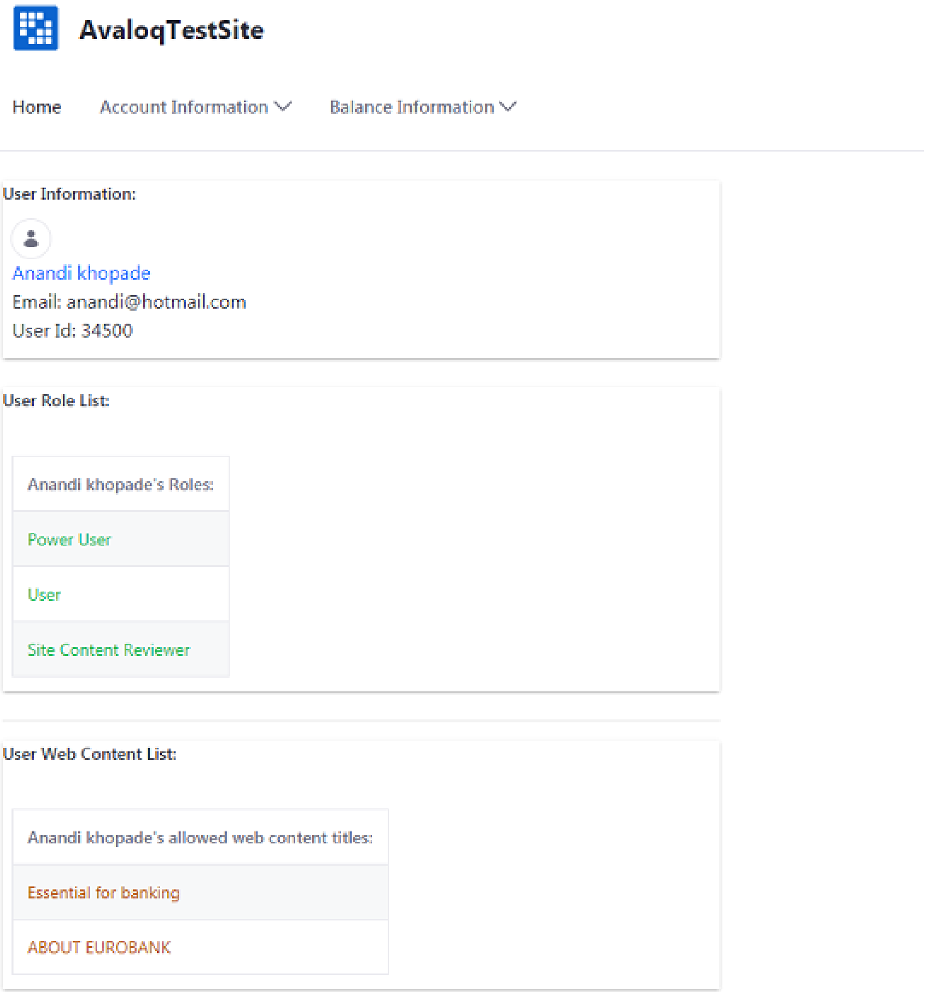
User Role List:

Lalita Markopoulou's Roles:
- Power User
- User
- Portal Content Reviewer
- Site Content Reviewer

User Web Content List:

Lalita Markopoulou's allowed web content titles:
- Essential for banking
- ABOUT EUROBANK

3) Aanadi with site content reviewer role:



User Information:

Anandi khopade
Email: anandi@hotmail.com
User Id: 34500

User Role List:

| Anandi khopade's Roles: |
|---|
| Power User |
| User |
| Site Content Reviewer |

User Web Content List:

| Anandi khopade's allowed web content titles: |
|---|
| Essential for banking |
| ABOUT EUROBANK |