

Smarter Balanced RFP #11/Program Management

Program Management / User Interface

Scenario Description:

Program Management UI when accessed via URL

Version Control

Version #	Date	Author	Description
1.0	10/03/2013	Ryan Marinello	Initial Draft

Test Scripts:

The following scripts, links and pages consist of validated Program Management UI

Test Components:

This test scenario covers the following high-level test requirements (see scripts below for specific requirements covered by each test script):

- 1.1 Home Page
- 1.2 Manage Tenants
- 1.3 Components
- 1.4 System Property Configs
- 1.5 Asset Pool,Group
- 1.6 Skinnable
- 1.7 Sticky Search

User Groups:

System Admin User

Requirements Traceability:

PRPM.1, PRPM.1.2, PRPM.1.3.2, PRPM.1.5, PRPM.3.9, PRPM.3.10, PRPM.3.12 PRPM.3.5,OTHPM.1

Testing Requirements:

- Ability to access Program Management UI cross browsers IE, Fire Fox, Chrome and Safari

Smarter Balanced RFP #11/Program Management

Program Management User Interface

- **Ability to interact with Program Management cross Operating Systems Windows, Linux, Macintosh**
- **User has access to PMI dashboard (SSO or permissions for accounts/logins do not exist – component not in scope, not owned by DRC)**

Script #:1.1 PMI Home Page

Purpose:

When PMI component is accessed using browser, Home page displays

Prerequisites:

- Webapp is online and user is able to access URL to dashboard

Test case Steps

Step #	Test Action	Expected Results	Pass/Fail
1	Open a supported browser and access link http://localhost:8080.com	PMI Dashboard home page displays	
2	Smarter Balanced branding	Branding images, colors, fonts and alignment meet standards	
3	User should be able to access different links from the home page	User is able to access different links from the Home page	

Test Execution

Date/Time	Tester	Test ID	Test Phase	Status
09/30/2013	RM	UI	1	Pass

Smarter Balanced RFP #11/Program Management

Program Management / User Interface

Script #:1.2 Manage Tenant

Purpose:

Verify creation of a Tenant and searching for a Tenant

Below are test scenarios

- 1.2: Create a new Tenant

Purpose:

Add a tenant

Prerequisites:

- User should be able to access tenant creation functionality on PMI dashboard
- A functional DB connection supports the dashboard
- Interaction with data entry boxes triggers bold warning to save data and a window boarded indicating user is actively working on page
- Navigation away from pages without saving work will trigger a pop-up to user seeking confirmation of data loss due to navigation off active page

Test Case Steps

Step #	Test Action	Expected Results	Pass/Fail
1	Open a supported browser and access link http://localhost:8080.com	PMI Dashboard home page displays	
2	Click new button	Tenant text box, Type selection list and Description text box display	
3	Enter data	User inputs for Tenant, Type and Description appear and write to a DB collection from Save	
4	Requirements	See above	

Test Execution

Date/Time	Tester	Test ID	Test Phase	Status
09/30/2013	RM	UI	1	Pass

Smarter Balanced RFP #11/Program Management

Program Management User Interface

- **1.2.b: Search Tenant**

Purpose:

Verify Search Tenant

Prerequisites:

- User is on PMI Tenant dashboard
- Stored data exists in DB collection

Test Case Steps

Step #	Test Action	Expected Results	Pass/Fail
1	Tenant text box inputs	Search widget returns context based search matches.	
2	Tenant Type Sorting	Type column sorts Tenant Types A-Z and Z-A	
3	Tenant Type filtering	Selected type returns all matching Tenants	
4	Tenant Sorting	Tenant column sorts Tenants A-Z, Z-A	
5	Prefill creation	If zero results return from input search data a user has the option to jump directly to creating a new tenant from search entry.	

Test Execution

Date/Time	Tester	Test ID	Test Phase	Status
09/30/2013	RM	UI	1	Pass

Smarter Balanced RFP #11/Program Management

Program Management / User Interface

Script #:1.3 Component

Purpose:

Verify Creation of a Component.

Below are test scenarios to create a Component

Purpose:

Verify creation of a Component

Prerequisites:

- User should be able to access PMI dashboard
- A functional DB connection supports the dashboard. Interaction with data entry boxes triggers bold warning to save data and a window boarded indicating user is actively working on page
- Navigation away from pages without saving work will trigger a pop-up to user seeking confirmation of data loss due to navigation off active page

Step #	Test Action	Expected Results	Pass/Fail
1	Open a supported browser and access link http://localhost:8080.com	PMI Dashboard home page displays	
2	Click Components button	Creation page displays	
3	Input component name	Text field accepts user description of component	
4	Save	Write creates an entry in DB collection and component name description appears in UI	
5	Subscription Status	Filled check box indicates a component has Paid their bills.	

Test Execution

Date/Time	Tester	Test ID	Test Phase	Status
09/30/2013	RM	UI	1	Pass

Smarter Balanced RFP #11/Program Management

Program Management User Interface

Script #:1.4 System Property Configs

Purpose:

Verify creation and updates of Property Configurations

Below are test scenarios to verify Property Configs

- 1.4:Creation Criteria Property Config

Purpose:

Verify user creation writes to DB collection and inputs search requests trigger reads from collection and event type and metadata data match, date and timestamp audit exists

Prerequisites:

- User should be able to access PMI dashboard
- A functional DB connection supports the dashboard. Interaction with data entry boxes triggers bold warning to save data and a window boarded indicating user is actively working on page
- Navigation away from pages without saving work will trigger a pop-up to user seeking confirmation of data loss due to navigation off active page

Step #	Test Action	Expected Results	Pass/Fail
1	Open a supported browser and access link http://localhost:8080.com	PMI Dashboard home page displays	
2	Click System Property Config button	Property Config page displays	
3	Click New	Name textbox and Environment Name text box accept input	
4	Key Value pairs	Key + Value saves	
5	Encryption	Non-human readable masked values display	
6	History of actions expander	Add, show in green, Updates in red , unchanged in no color	
7	Date and time stamp	Grid contains date and time per line item	
8	Grid functions	Sorting per column header data sorts A-Z and Z-A	

Smarter Balanced RFP #11/Program Management

Program Management / User Interface

Test Execution

Date/Time	Tester	Test ID	Test Phase	Status
09/30/2013	RM	UI	1	Pass

Script #:1.5 Asset Pool and Asset Groups

Purpose:

Verify functionality of Asset Pool and Asset Group UI

Below are test scenarios to verify Assets dashboards

1.5. a: Creation of an Asset Pool (file upload)

Purpose:

To verify file upload on Asset Pool page

Prerequisites:

- User is able to access PMI dashboard
- A functional DB connection supports the dashboard. Interaction with data entry boxes triggers bold warning to save data and a window boarded indicating user is actively working on page
- Navigation away from pages without saving work will trigger a pop-up to user seeking confirmation of data loss due to navigation off active page

Step #	Test Action	Expected Results	Pass/Fail
1	Open a supported browser and access link http://localhost:8080.com	PMI Dashboard home page displays	
2	Click text box	Context sensitive Tenant search widget returns matching results based off user input	
3	Tenant Selection	Tenant returned ready for file uploading	
4	File Upload	Browse to a directory location for file(s) or drag and drop files. Upload Files was Successful message displays.	
5	Upload all	System uploads all users added files to tenant	
6	Upload individual file	Single file additions upload	
7	Cancel	Cancels upload(s)	

Smarter Balanced RFP #11/Program Management

Program Management User Interface

Test Execution

Date/Time	Tester	Test ID	Test Phase	Status
09/30/2013	RM	UI	1	Pass

- **1.5.b: Asset Group**

Purpose:

Verify Asset Group (tenant in good standing + component + asset files from pool)

Prerequisites:

- User is able to access PMI dashboard
- A functional DB connection supports the dashboard. Interaction with data entry boxes triggers bold warning to save data and a window boarded indicating user is actively working on page
- Navigation away from pages without saving work will trigger a pop-up to user seeking confirmation of data loss due to navigation off active page

Test Case Steps

Step #	Test Action	Expected Results	Pass/Fail
0	Open a supported browser and access http://localhost:8080.com	PMI Dashboard Home displays	
1	User is at the Asset Group page	Asset Groups landing page returns	
3	Click New	New asset Group page returns	
4	Create Group	Enter tenant and select component	
5	Click icon to add asset	A single row to add asset information displays	
6	Save	Writes asset group information to a DB collection and UI displays saved work	
7	Cloning by clicking the copy clipboard icon	Any Previously created asset groups can be cloned to a component combination that does not yet exist	

Test Execution

Date/Time	Tester	Test ID	Test Phase	Status
09/30/2013	RM	UI	1	Pass

Smarter Balanced RFP #11/Program Management

Program Management / User Interface

Script #:1.6 Skinnable Demo

Purpose: Verify Skinnable Assets

Below are test scenarios

- **1.6: Display of Skinnable assets**

Purpose:

Verify (logo,title,main-header,background and meme display for appropriate tenant type and tenant chain) Skinnable assets

Prerequisites:

- User is able to access PMI dashboard
- A functional DB connection supports the dashboard
- Asset Pool contains files
- Asset Group has been created (tenant + component)

Test Case Steps

Step #	Test Action	Expected Results	Pass/Fail
1	Open a supported browser and access http://localhost:8080.com	PMI dashboard Home displays	
2	Click Skinnable assets button	Supporting page for skinnable assets displays	
3	Select Mock user data	Defines user selected hard coded example as demo	
4	Click component 'skinnable'	DDL populates with component name	
5	Click through hard coded user names	Examples demonstrate tenant chain functionality	
6	Hard coded examples	Demo of skinned UI	
7	Hard coded examples	Demo of *has asset group	
8	Select use Attributes directly	Defines the use of Asset Group information should be displayed in the demo page	
9	Click component 'skinnable'	DDL populates with component name	
10	Enter Tenant name in a Type text box	Search widget querying begins to return context based results	
11	Skinnable demo page	Pre-configured custom assetGroup logo, title, header, background and meme display in demo page when *has assetGroup match by Tenant Type is made	

Smarter Balanced RFP #11/Program Management

Program Management User Interface

Test Execution

Date/Time	Tester	Test ID	Test Phase	Status
09/30/2013	RM	UI	1	Pass

Script #:1.7 Sticky Search

Purpose: Keystroke saving power

Below are test scenarios

- **1.7; sticky searching**
Purpose:
Verify sticky search for admin users is a nice to have

Prerequisites:

- User is able to access PMI dashboard
- A functional DB connection supports the dashboard

Test Case Steps

Step #	Test Action	Expected Results	Pass/Fail
1	Open a supported browser and access http://localhost:8080.com	PMI dashboard Home displays	
2	Tenants	Entered text will remain on default search page until user clears it out	
3	AssetGroups	Entered text remains on default search page until user clears it out	

Test Execution

Date/Time	Tester	Test ID	Test Phase	Status
09/30/2013	RM	UI	1	Pass