AEM:

Comprehensive Assessment Case Study

You are part of the SDET team for the developing the application as per the reference URL mentioned below. Go through the complete website and study the features/functionalities. Perform the following activities as per your understanding with the application.

1) Identify the EPIC, User-stories and acceptance criteria.

2) Create the test plan and test strategy document as per the requirement.

3) Design the test scenarios and test cases using the template for all the functionalities in the application.

4) Prepare the Requirement Traceability Matrix and map scenarios and test cases.

5) Build 3 automation script based on the test cases we identified. The scripts should be part of one Java project.

6) Create a new repository in Git and push the project (Created in step-5) to the repository.

7) Configure Jenkins with Git for continuous integration of automation scripts.

8) Create test summary report to be shared with stakeholder highlighting test execution status.

9) Create a Bug highlighting different fields required to log/raise a new bug. Use word document for this activity.

Ref Link:

<https://www.nobroker.in/>

Webservice API testing using SOAP UI and automation using Groovy script

API URL: https://mindtree-emp.azurewebsites.net/swagger-ui.html#/employee-controller

1. Go through the above URL to get details about the APIs.

2. Create a rest project using the above rest service API.

2. Add test suite, test case and test step for testing webservice API for GET, POST, PUT & DELETE requests

3. Design a groovy script to read the data for competency, id, name and yearOfJoining from excel file and perform the below operation

a) Use the data we read in step-3 to process the POST request.

b) Use the GET request (https://mindtree-emp.azurewebsites.net/api/v1/employees/<empID>) within the groovy script to verify the employee data that we added in step – 3a.

c) Use PUT request to update the data for particular employee and verify with the GET request.

d) Use DELETE request to delete an employee by reading the empID from the excel file and verify with GET request to confirm if the employee is deleted.

e) Make use of assertions to verify the response.

AEM Template and Page Creation

Objective

· Inspect a page design created in Adobe XD and map it to Core Components.

· Understand the details of Editable Templates and how policies can be used to enforce granular control of page content.

· Learn how Templates and Pages are linked

What you will build

In this part of the tutorial, you will build a new Article Page Template that can be used to create new article pages and aligns with a common structure. The Article Page Template will be based on designs and a UI Kit produced in AdobeXD. This chapter is only focused on building out the structure or skeleton of the template. No styles will be implemented but the template and pages will be functional.

Update the Header and Footer with Experience Fragments

A common practice when creating global content, such as a header or footer, is to use an Experience Fragment. Experience Fragments, allows users to combine multiple components to create a single, reference-able, component. Experience Fragments have the advantage of supporting multi-site management and localization.

The AEM Project Archetype generated a Header and Footer. Next, update the Experience Fragments to match the mockups.

High level steps for below:

1) Download the sample content package

2) Upload and install the content package using Package Manager at http://localhost:4502/crx/packmgr/index.jsp

3) Update the Web Variation template, which is the template used for Experience Fragments at http://localhost:4502/editor.html/conf/wknd/settings/wcm/templates/xf-web-variation/structure.html

i) Update the policy the Container component on the template.

ii) Set the policy to XF Root.

iii) Under Allowed Components select the component group WKND Sites Project - Structure to include Language Navigation, Navigation, and Quick Search components.

Update Header Experience Fragment

1. Open the Experience Fragment that renders the Header at http://localhost:4502/editor.html/content/experience-fragments/wknd/us/en/site/header/master.html

2. Configure the root Container of the fragment. This is the outer most Container.

· Set the Layout to Responsive Grid

3. Add the WKND Dark Logo as an image to the top of the Container. The logo was included in the package installed in a previous step.

· Modify the layout of the WKND Dark Logo to be 2 columns wide. Drag the handles from right to left.

· Configure the logo with Alternative Text of “WKND Logo”.

· Configure the logo to Link to /content/wknd/us/en the Home page.

4. Configure the Navigation component that is already placed on the page.

· Set the Exclude Root Levels to 1.

· Set the Navigation Structure Depth to 1.

· Modify the layout of the Navigation component to be 8 columns wide. Drag the handles from right to left.

5. Remove the Language Navigation component.

6. Modify the layout of the Search component to be 2 columns wide. Drag the handles from right to left. All the components should now be aligned horizontally on a single row.

Update Footer Experience Fragment

1. Open the Experience Fragment that renders the Footer at http://localhost:4502/editor.html/content/experience-fragments/wknd/us/en/site/footer/master.html

2. Configure the root Container of the fragment. This is the outer most Container.

· Set the Layout to Responsive Grid

3. Add the WKND Light Logo as an image to the top of the Container. The logo was included in the package installed in a previous step.

· Modify the layout of the WKND Light Logo to be 2 columns wide. Drag the handles from right to left.

· Configure the logo with Alternative Text of “WKND Logo Light”.

· Configure the logo to Link to /content/wknd/us/en the Home page.

4. Add a Navigation component beneath the logo. Configure the Navigation component:

· Set the Exclude Root Levels to 1.

· Uncheck Collect all child pages.

· Set the Navigation Structure Depth to 1.

· Modify the layout of the Navigation component to be 8 columns wide. Drag the handles from right to left.

Create an Article Page

Next, create a new page using the Article Page template. Author the content of the page to match the site mockups.

High level steps for below:

1. Navigate to the Sites console at http://localhost:4502/sites.html/content/wknd/us/en/magazine.

2. Create a new page beneath WKND > US > EN > Magazine.

· Choose the Article Page template.

· Under Properties set the Title to “Ultimate Guide to LA Skateparks”

· Set the Name to “guide-la-skateparks”

3. Replace By Author Title with the text “By Stacey Roswells”.

4. Update the Text component to include a paragraph to populate the article. You can use the following text file as the copy: la-skate-parks-copy.txt.

5. Add another Text component.

· Update the component to include the quote: “There is no better place to shred then Los Angeles.”.

· Edit the Rich Text Editor in full-screen mode and modify the above quote to use the Quote Block element.

6. Continue populating the body of the article to match the mockups.

7. Configure the Download component to use a PDF version of the article.

· Under Download > Properties, click the checkbox to Get the title from the DAM asset.

· Set the Description to: “Get the Full Story”.

· Set the Action Text to: “Download PDF”.

8. Configure the List component.

· Under List Settings > Build List Using, select Child Pages.

· Set the Parent Page to /content/wknd/us/en/magazine.

· Under Item Settings check Link Items and check Show date

mvn archetype:generate -DarchetypeGroupId=com.adobe.granite.archetypes -DarchetypeArtifactId=aem-project-archetype -DarchetypeVersion=20

mvn clean install

mvn clean install -PautoInstallPackage