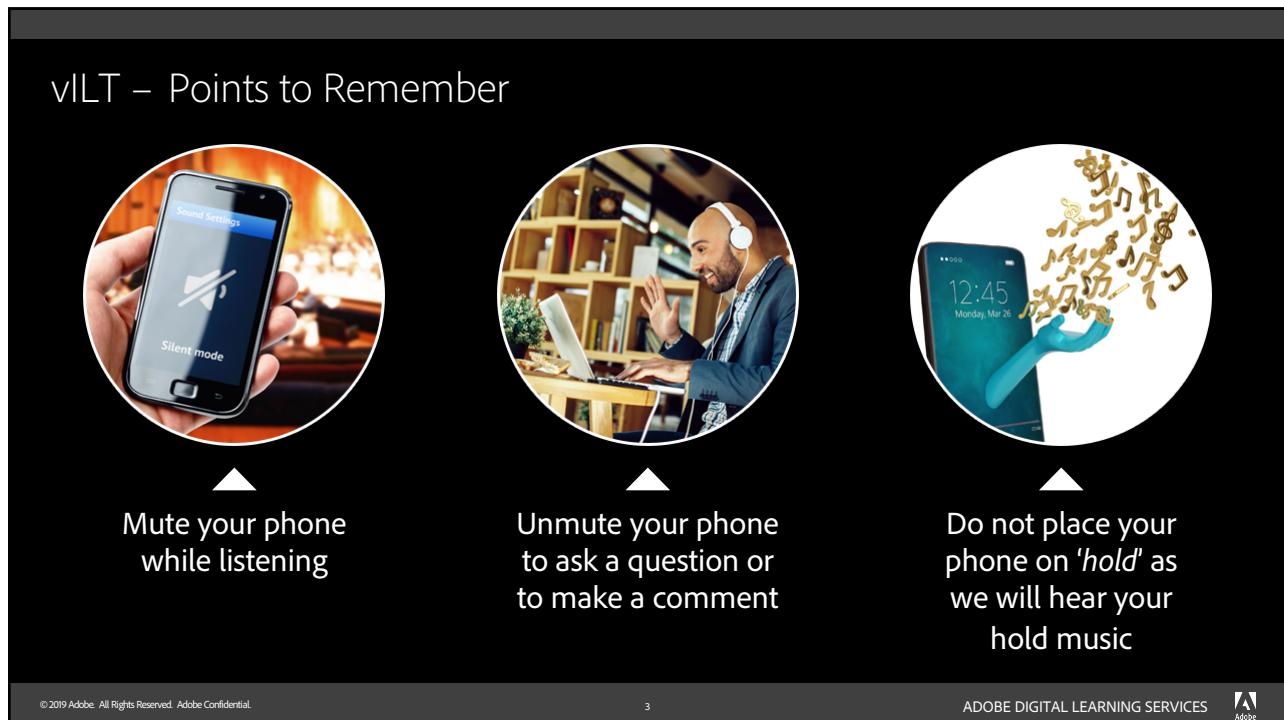




The slide features the Adobe logo in the top left corner. The title "Adobe Marketing Cloud" and subtitle "Getting Started with React and AEM Single-Page Application (SPA) Editor" are displayed in white text at the top. Below the text is a large, abstract circular graphic composed of concentric, glowing blue and teal arcs against a dark background. In the bottom left corner, there is a small icon of a person inside a circle, followed by the text "Adobe Experience Manager".

1

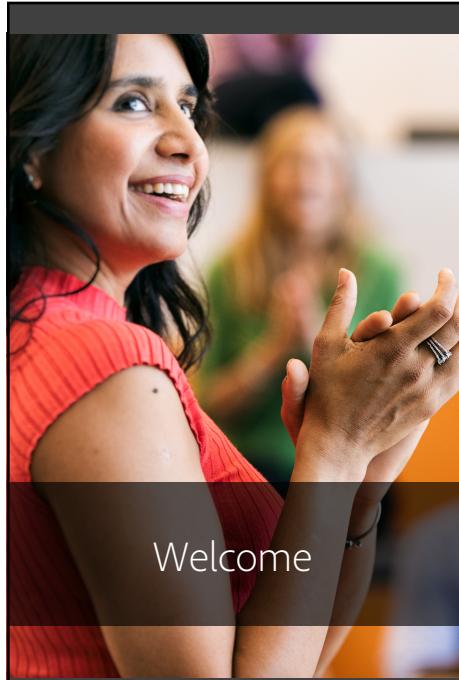


The slide is titled "vILT – Points to Remember" in white text at the top. It contains three circular icons with arrows pointing down to corresponding text instructions:

-  Mute your phone while listening
-  Unmute your phone to ask a question or to make a comment
-  Do not place your phone on 'hold' as we will hear your hold music

At the bottom of the slide, there is a footer bar with the text "© 2019 Adobe. All Rights Reserved. Adobe Confidential." on the left, a page number "3" in the center, and "ADOBEDIGITAL LEARNING SERVICES" along with the Adobe logo on the right.

3



Welcome

Welcome to
Getting Started with React and AEM SPA Editor!

- What is your name?
- Where are you today?
- What company are you with?
- What is your role?
- How will you be using AEM SPA?
- What is your development experience with AEM, if any?
- What are your expectations from this training?

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

4

ADOBEDIGITAL LEARNING SERVICES 

4



Course Agenda

Agenda:

Day 1

- Module 1: Overview
- Module 2: Project Setup
- Module 3: Editable Components

Day 2

- Module 3: Editable Components
- Module 4: Front-End Development
- Module 5: Navigation and Routing

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

5

ADOBEDIGITAL LEARNING SERVICES 

5

2



Daily Schedule (all time Central)

Class runs from 9:00 to 5:00

Morning Break 15 minutes around 10:30

Lunch Break of 60 minutes around noon

Afternoon Break of 15 around 3:30

Course Agenda

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

6

ADOBEDIGITAL LEARNING SERVICES 

6

Course Objectives

Upon successful completion of this course, you will be able to:

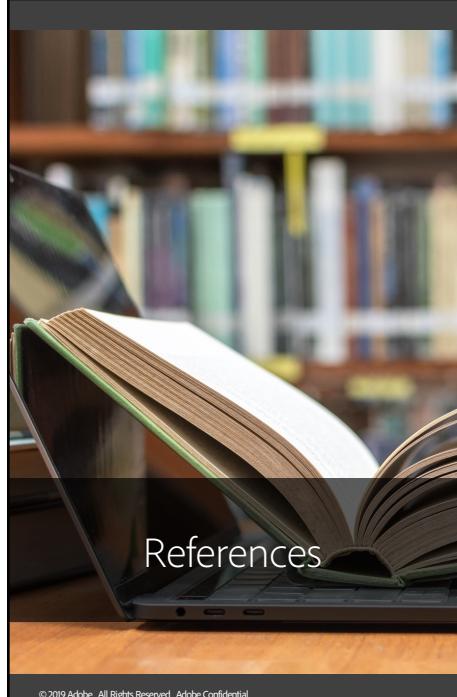
- Implement a Single-Page Application (SPA) in React that can be edited in Adobe Experience Manager (AEM) with the AEM SPA Editor.
- Understand the end-to-end creation of the SPA and the integration with AEM.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

7



7



© 2019 Adobe. All Rights Reserved. Adobe Confidential.

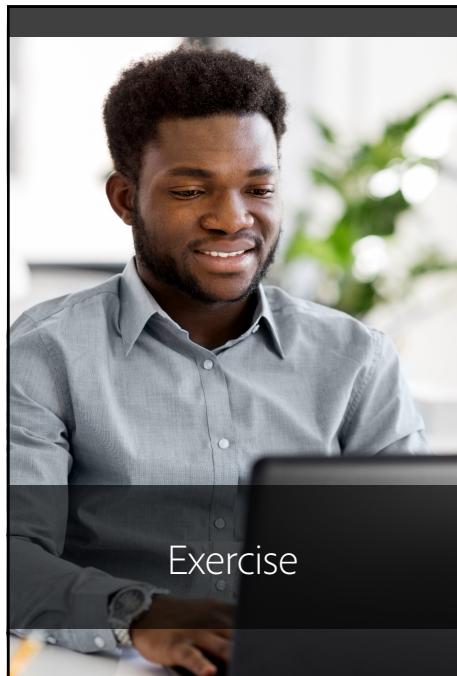
Course Resources

Hands-on instructions and content provided by:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react.html>

All of the code for this course can be found on GitHub:

- <https://github.com/Adobe-Marketing-Cloud/aem-guides-wknd-events>



© 2019 Adobe. All Rights Reserved. Adobe Confidential.

Course Exercises / Labs

Based on WKND project

- A fictional online magazine and blog that focuses on nightlife, activities, and events in several international cities.
- Each task is labeled with a persona to gain a better idea of the separation of development activities.



11

Traditional CMS Architecture

- Manages and delivers content on a single technology stack.
- Minimizes total cost of ownership with efficient system maintenance and training.
- Controls all of the templating and presentation logic.
- Outputs fully formatted HTML.

Often tightly coupled with a single channel

- Results in a monolithic software architecture
- Can make multichannel content delivery difficult and costly.

```

graph TD
    WebBrowser[Web browser] -- "HTML" --> FrontEnd[Front-end]
    FrontEnd --> ContentManagement[Content management]
    ContentManagement --> Authors[Authors]
  
```

The diagram illustrates the traditional CMS architecture. At the top is a web browser icon labeled "Web browser". An arrow labeled "HTML" points down to a blue box labeled "Front-end". From the "Front-end" box, an arrow points down to an orange box labeled "Content management". Finally, an arrow points up from the "Content management" box to a user icon labeled "Authors". A bracket on the right side of the diagram is labeled "Traditional CMS", encompassing the "Content management" and "Authors" components.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

12

Headless CMS Architecture

The diagram illustrates the Headless CMS Architecture. At the bottom, 'Authors' interact with a central 'Content management' system. This system is connected to 'HTTP APIs'. Above the APIs, a dashed box labeled 'Headless CMS' contains the 'Content management' and 'HTTP APIs' components. Arrows show data flowing from the 'Content management' to the 'HTTP APIs', and from the 'HTTP APIs' to various front-end channels: 'Mobile app', 'Single page app', 'IoT', and 'Social media'. Each channel is represented by a small icon above an arrow pointing to the 'HTTP APIs'. The entire architecture is set against a dark blue background.

- Does not deliver HTML or formatted content directly.
- Decouples content from presentation.
- Enables content to be used by a variety of front-end technologies.
- Exposes content through well-defined HTTP APIs.

Note: The term 'headless' originates from the idea that the front-end presentation layer is the "head" of the application

13



Hybrid CMS Architecture

AEM takes a hybrid approach

- Efficiency and ease of use of a traditional CMS
- Flexibility and scalability of a headless development framework

Provides a central hub to house and distribute content and experiences to any channel and delivers content either as fully formatted HTML or as JavaScript Object Notation (JSON) over HTTP APIs.

Brands can choose between traditional or headless delivery and can mix and match capabilities to meet their business needs.

The diagram illustrates the Hybrid CMS Architecture. At the bottom, 'Authors' interact with a central 'Content management' system. This system is connected to two parallel paths: 'Front-end' and 'HTTP APIs'. The 'Front-end' path leads to a 'Web browser', while the 'HTTP APIs' path leads to various front-end channels: 'Mobile app', 'Single page app', 'IoT', and 'Social media'. Each channel is represented by a small icon above an arrow pointing to the 'HTTP APIs'. The entire architecture is set against a dark blue background.

14



Multi-Channel Content Delivery and Content Services

To support all channels, a CMS must expose the managed content in a multi-format and scalable way.

Standard formats for exposing web content:

- HTML – language of the web browser
- JSON – lightweight delivery format that powers modern connected applications

Components reference content to expose as HTML.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

15



15

Enabling SPA with AEM Site SPA Editor

Developers can build SPA experiences rapidly with popular JavaScript frameworks such as React and Angular.

- Typically using decoupled or "headless" CMS

AEM Sites

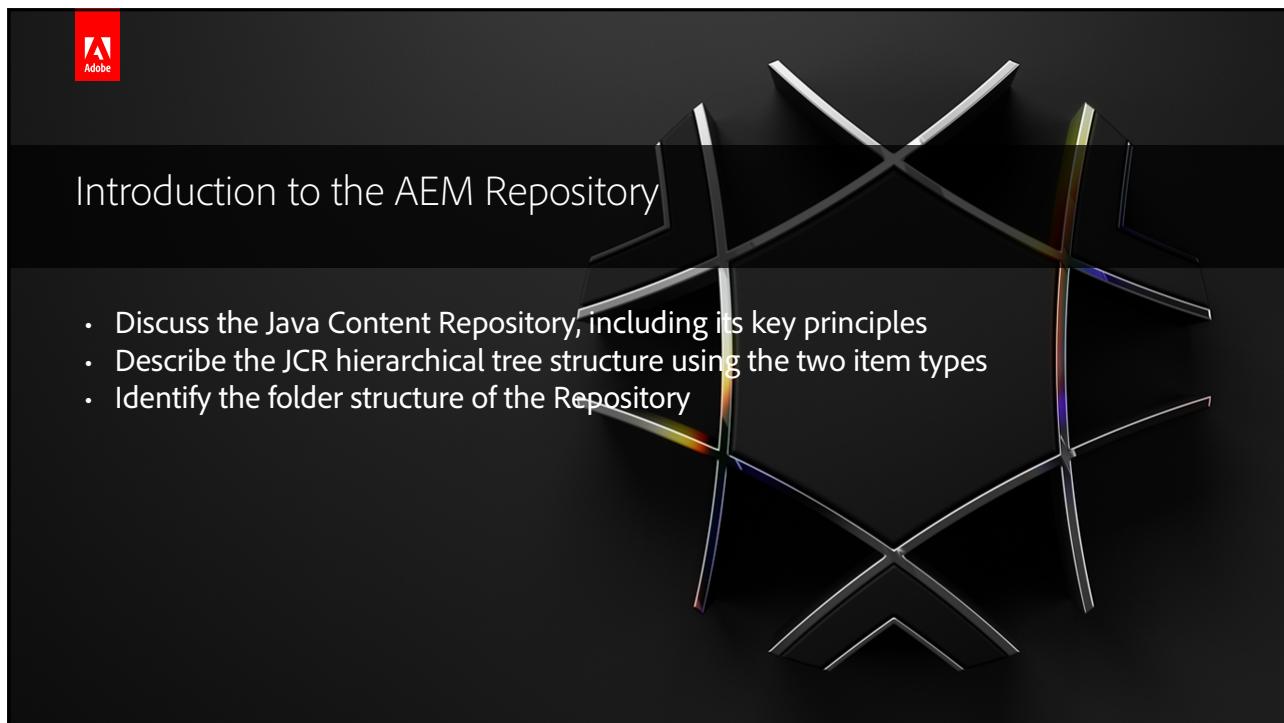
- Enables developers by integrating with familiar frameworks and development tools.
- SPA Editor offers a WYSIWYG ("What You See Is What You Get") user interface that enables marketers and authors to make in-context changes to content, layout, and presentation, just as they would with traditional webpages.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

16



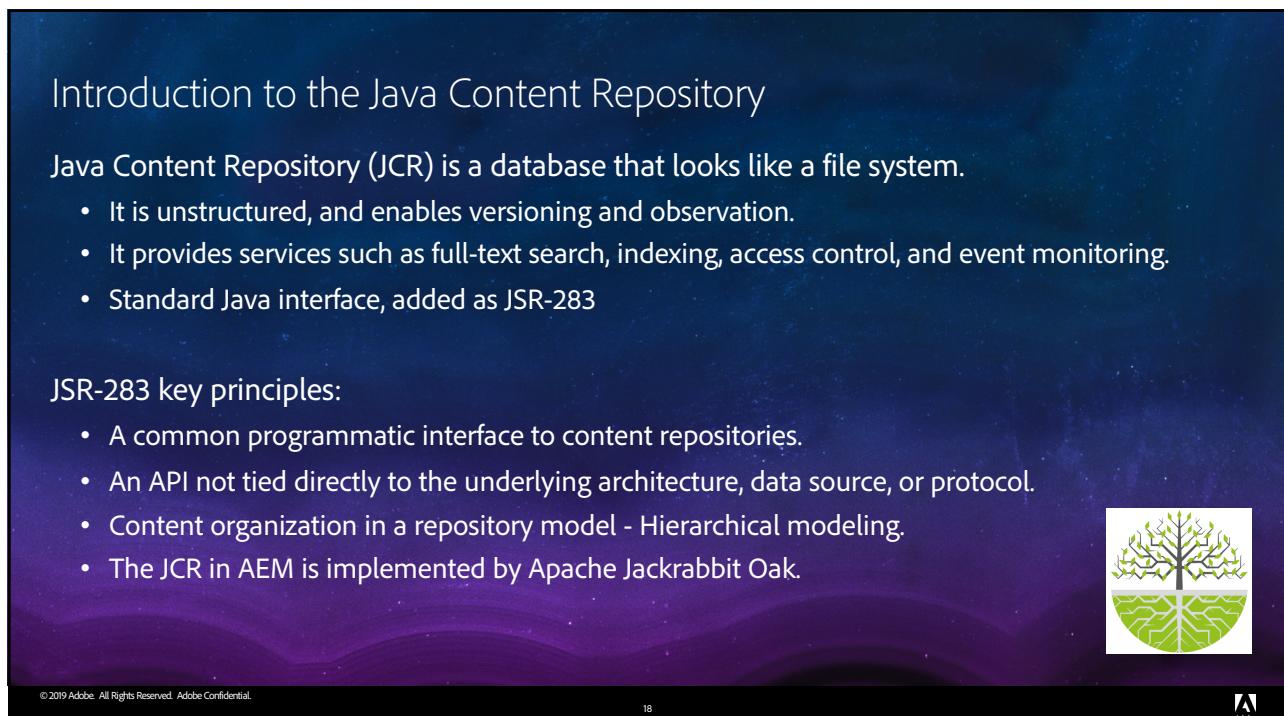
16



The slide features a dark background with a central 3D wireframe model of a tree or network structure. In the top left corner is the Adobe logo. The title "Introduction to the AEM Repository" is displayed in white text. Below the title is a bulleted list of learning objectives:

- Discuss the Java Content Repository, including its key principles
- Describe the JCR hierarchical tree structure using the two item types
- Identify the folder structure of the Repository

17



The slide has a dark blue gradient background with a stylized green tree icon in the bottom right corner. The title "Introduction to the Java Content Repository" is at the top. The text "Java Content Repository (JCR) is a database that looks like a file system." is followed by a bulleted list:

- It is unstructured, and enables versioning and observation.
- It provides services such as full-text search, indexing, access control, and event monitoring.
- Standard Java interface, added as JSR-283

The section "JSR-283 key principles:" is followed by another bulleted list:

- A common programmatic interface to content repositories.
- An API not tied directly to the underlying architecture, data source, or protocol.
- Content organization in a repository model - Hierarchical modeling.
- The JCR in AEM is implemented by Apache Jackrabbit Oak.

At the bottom left, there is small text: "© 2019 Adobe. All Rights Reserved. Adobe Confidential." and "18". At the bottom right is the Adobe logo.

18

Basic JCR Structure: Nodes and Properties

- JCR Repository is a tree of Items
 - Nodes provide structure
 - Properties store data
- Properties are attached to nodes
- Items names should have meaning
- Items addressed by paths
 - /blog/jan/01
 - /blog/jan/31/title

```

graph TD
    root["/ (root)"] --> blog["/ blog"]
    blog --> jan["/ jan"]
    blog --> feb["/ feb"]
    jan --> zeroOne["/ 01"]
    jan --> thirtyOne["/ 31"]
    ...
    thirtyOne --> title["/title: A Very Long Day"]
    thirtyOne --> body["/body: Today seemed..."]
    thirtyOne --> keywords["/keywords[]: winter, snow"]
  
```

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

20

Adobe

20

JCR Nodes and Properties (cont.)

parent node

↳ this node

- node type: nt:folder nt:file nt:unstructured rep:User rep:ACL cq:page ...
- mixin types: mix:versionable mix:lockable sling:VanityPath cq:Taggable ...
- single value properties: value string long date path binary ...
- multi-value properties: values[] string long date path binary ...
- child nodes

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

22

Adobe

22

Common Node Types			
JCR		AEM	
Node Type	Description	Node Type	Description
nt:file	Represents a file in a filesystem	cq:page	Stores content and properties for a page in a website
nt:folder	Represents a folder in a filesystem	cq:template	Defines a template used to create pages
nt:unstructured	Allows any combination of child nodes and properties. It also supports client-orderable child nodes and stores unstructured content and commonly used (for touch UI) dialog boxes.	cq:clientLibraryFolder	Defines a library of client-side JavaScript & CSS
		cq:editConfig	Defines editing configuration for a component including drag-and-drop and in-place editing

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

23



23

Folder Structure of the Repository

The screenshot shows the CRXDE Lite interface with a tree view of the repository structure. A red box highlights the top-level folders: /apps, /bin, /conf, /content, /etc, /home, /jcr.system, and /libs. Blue lines connect these to numbered callouts on the right:

- 1 /apps – clientlibs, components, bundles, **front-end developers' work**
- 2 /conf – context-aware and global configurations such as editable templates, metadata schemas, metadata profiles, search filters, and cloud configurations
- 3 /content – page content authored by **authors**
- 4 /libs – foundation components, css, and library files

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

24



24

Folder Structure of the Repository

Apps Overrides Libs
Code and Configuration in /apps
overrides code and configuration in /libs

- 1 /apps – clientlibs, components, bundles, **front-end developers' work**
- 2 /conf – context-aware and global configurations such as editable templates, metadata schemas, metadata profiles, search filters, and cloud configurations
- 3 /content – page content authored by **authors**
- 4 /libs – foundation components, css, and library files

25

Client-Side Libraries

In AEM, you can include CSS and JS libraries just like in traditional site development, but....

- *This may not be a good idea* because:
 - It increases page overhead—files are included in all pages whether they are needed or not.
 - Components will not be portable.
 - You have to manually examine/modify the new site templates to ensure that the component has all necessary libraries.

Instead...

Convenience wrapper for com.day.cq.widget.HtmlLibraryManager service interface

- Manages client-side inclusions like JS and CSS
- Determines which files should be included on the page
- Ensures duplicate files are not sent

© 2019 Adobe. All Rights Reserved. Adobe Confidential.



26

Client-Side Libraries (cont.)

AEM uses client-side library folders to organize the client-side libraries logically.

The basic goals of client-side libraries are to:

- Store CSS/JS in small discrete files for easier development and maintenance.
- Manage dependencies on third-party frameworks in an organized fashion.
- Minimize the number of client-side requests by concatenating CSS/JS into one or two requests.
- Minimize CSS/JS that is delivered to optimize speed/performance of a site.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

27



27

Structure of Client-Side Libraries

A client-side library folder is a repository node of type `cq:ClientLibraryFolder`.

The definition in Compact Namespace and Node Type Definition (CND) notation is:

```
[cq:ClientLibraryFolder] - jcr:primaryType
  - categories (String[])
  - embed (String[])
  - dependencies (String[])
  - allowProxy = true
  + css.txt (nt:file)
  + js.txt (nt:file)
```

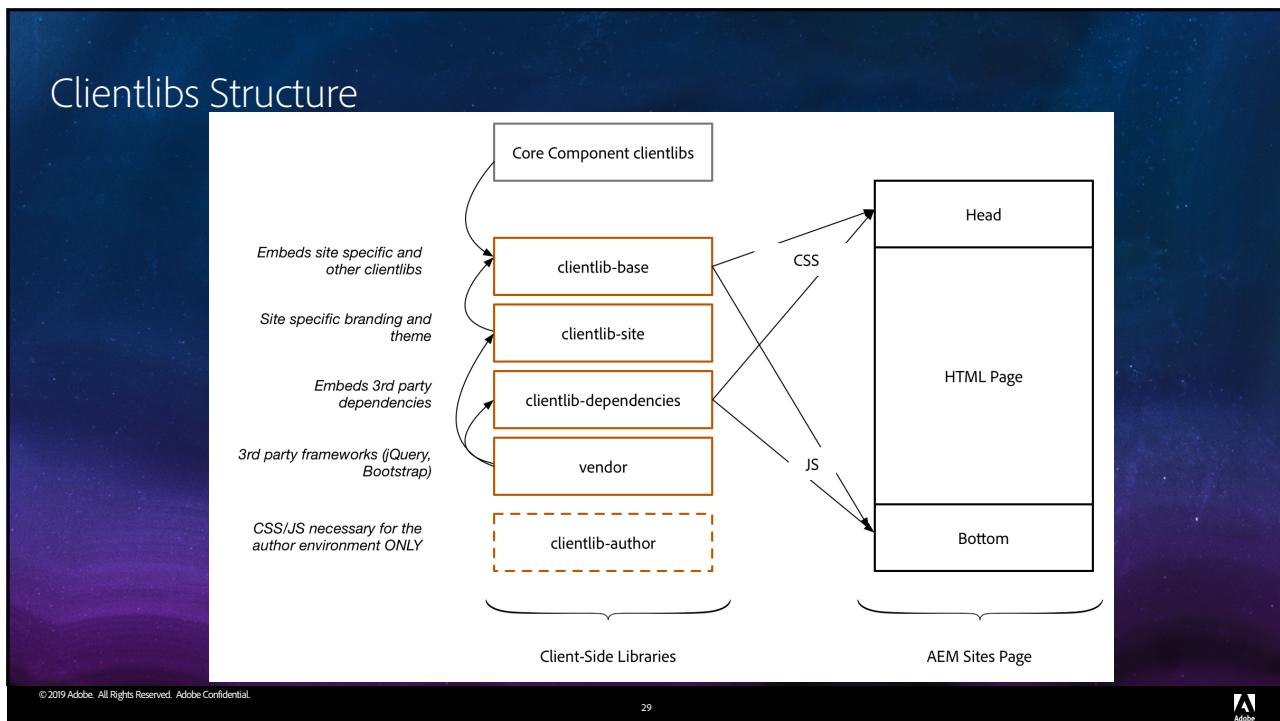
Name	Type	Value
1 jcr:primaryType	Name	cq:ClientLibraryFolder
2 jcr:createdBy	String	admin
3 jcr:created	Date	2018-03-13T13:25:27.488Z
4 allowProxy	Boolean	true
5 embed	String[]	we.train.site, we.train.fonts
6 categories	String[]	we.train.base

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

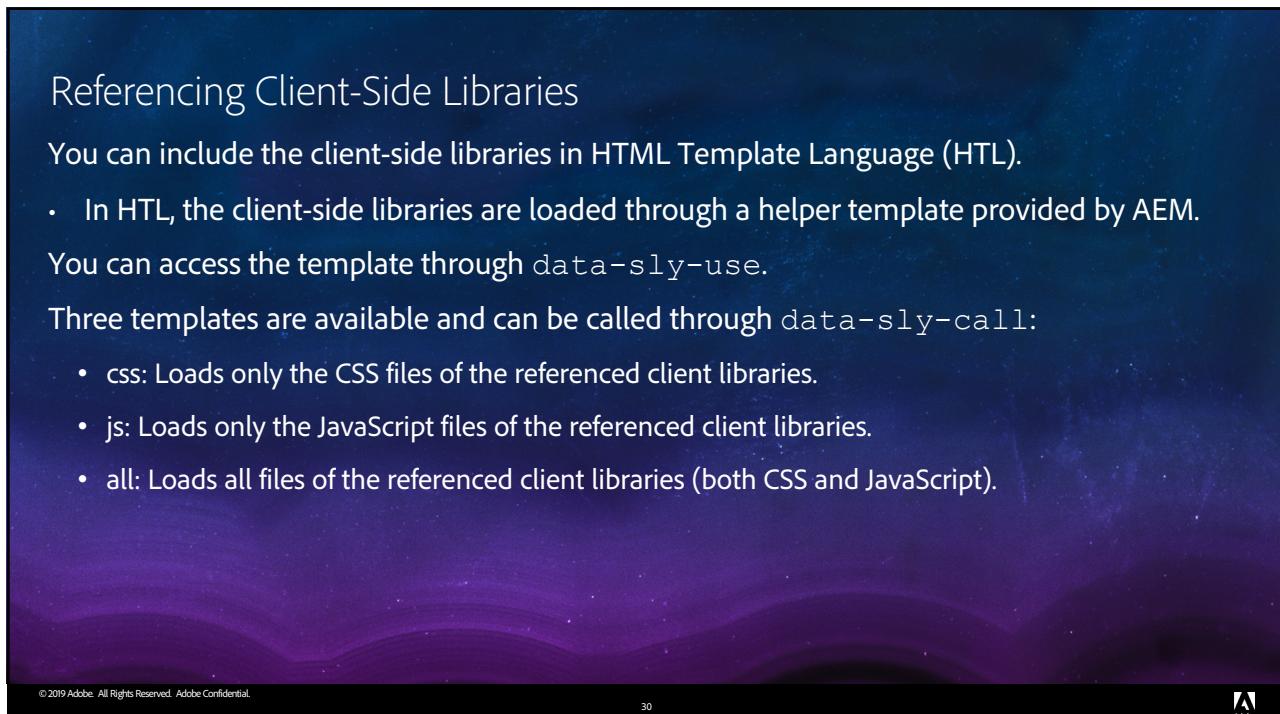
28



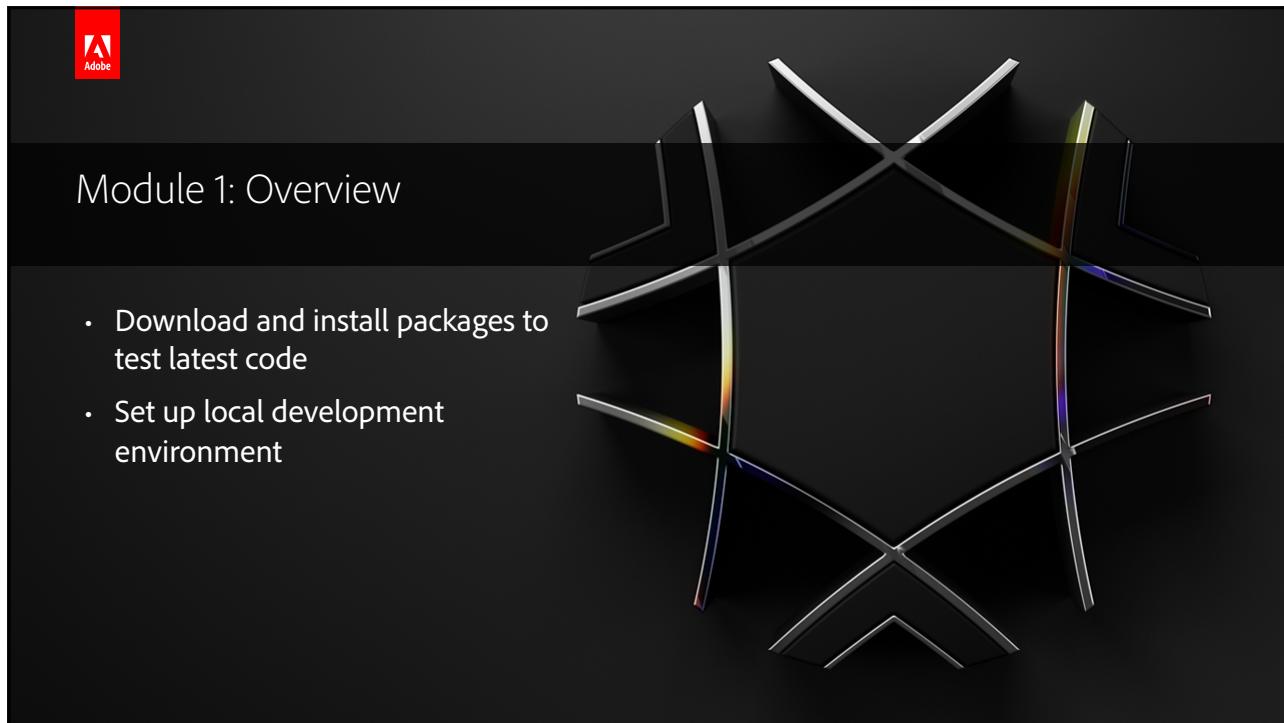
28



29



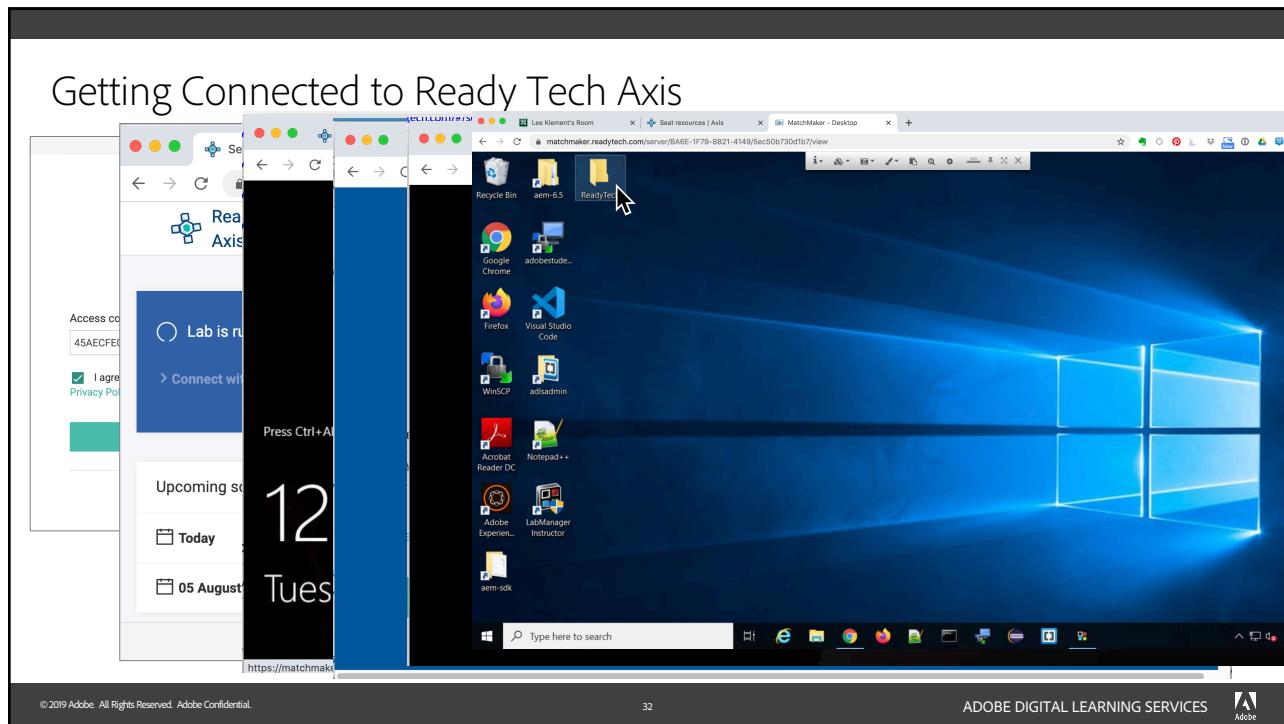
30



Module 1: Overview

- Download and install packages to test latest code
- Set up local development environment

31



Getting Connected to Ready Tech Axis

Access code: 45AECFED

I agree Privacy Policy

Lab is running

Upcoming schedule

Today

05 August

12 Tues

https://matchmaker.readytech.com

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

32

ADobe DIGITAL LEARNING SERVICES

Adobe

32

14

Basic Content Structure for this Course

Two editable templates

- WKND App Template - represent the root of the application.
- WKND Event Page - normal content page.

Page Structure

- WKND Events
 - > React App
 - > Home

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

33

Prerequisites: Local Development Environment

Software

- Java 1.8
- AEM 6.5 or AEM 6.4 + SP2
- Apache Maven (3.3.9 or newer)
- Node.js V10+
- npm 6+
- Eclipse or other IDE

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

34

ADOBEDIGITAL LEARNING SERVICES

Adobe



35

Setting up the AEM Plug-in for Eclipse

The plug-in is based on the Eclipse plug-in for Apache Sling.

It has the following benefits:

- Synchronizes both content and OSGi bundles.
- Supports debugging and code hot-swapping.
- Includes a project creation wizard to simplify bootstrap of AEM projects.
- Enables edition of JCR properties.
- Seamlessly integrates with AEM instances through Eclipse Server Connector.



AEM Developer Tools

The Eclipse Plug-In that brings you the full connection to the Adobe Experience Manager.

» [Online Documentation](#)

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

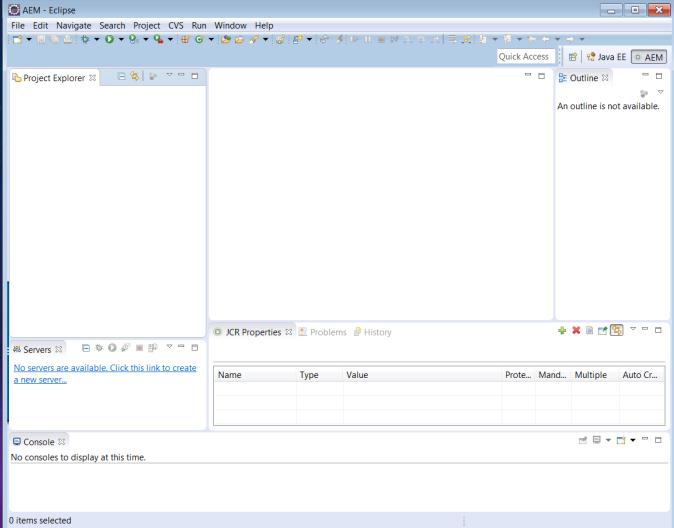


36

Configuring the AEM Perspective

The AEM perspective offers a complete control over all your AEM projects and instances.

To open the AEM perspective, navigate to **Windows > Open Perspectives > Other** and select **AEM**.



© 2019 Adobe. All Rights Reserved. Adobe Confidential.

37

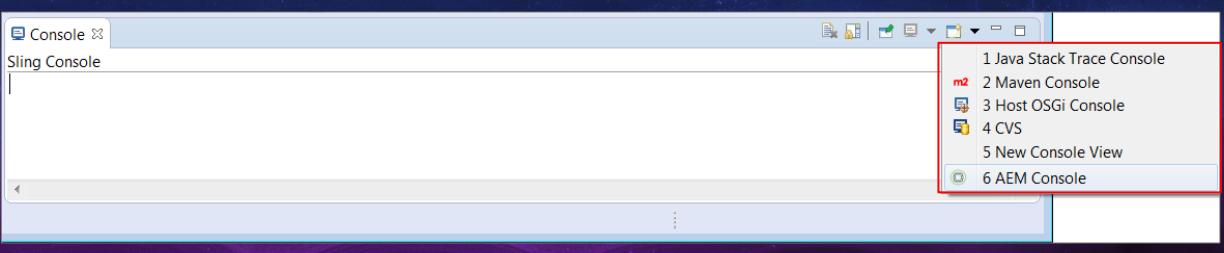
Configuring the AEM Console

The AEM Console displays the progress of repository synchronization.

- View the progress of all the check ins and check outs of nodes and properties to and from the repository.

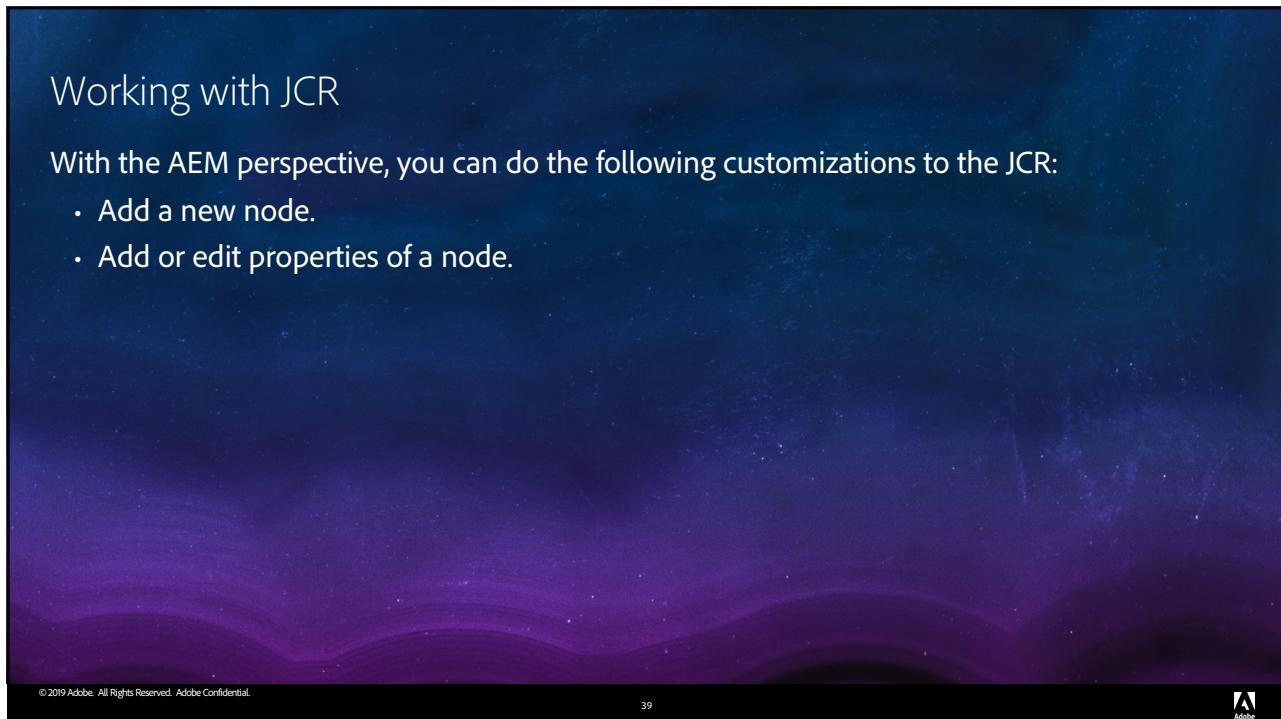
To configure the AEM console, from the **Console** tab:

- Click the Open Console icon.
- Select AEM Console.

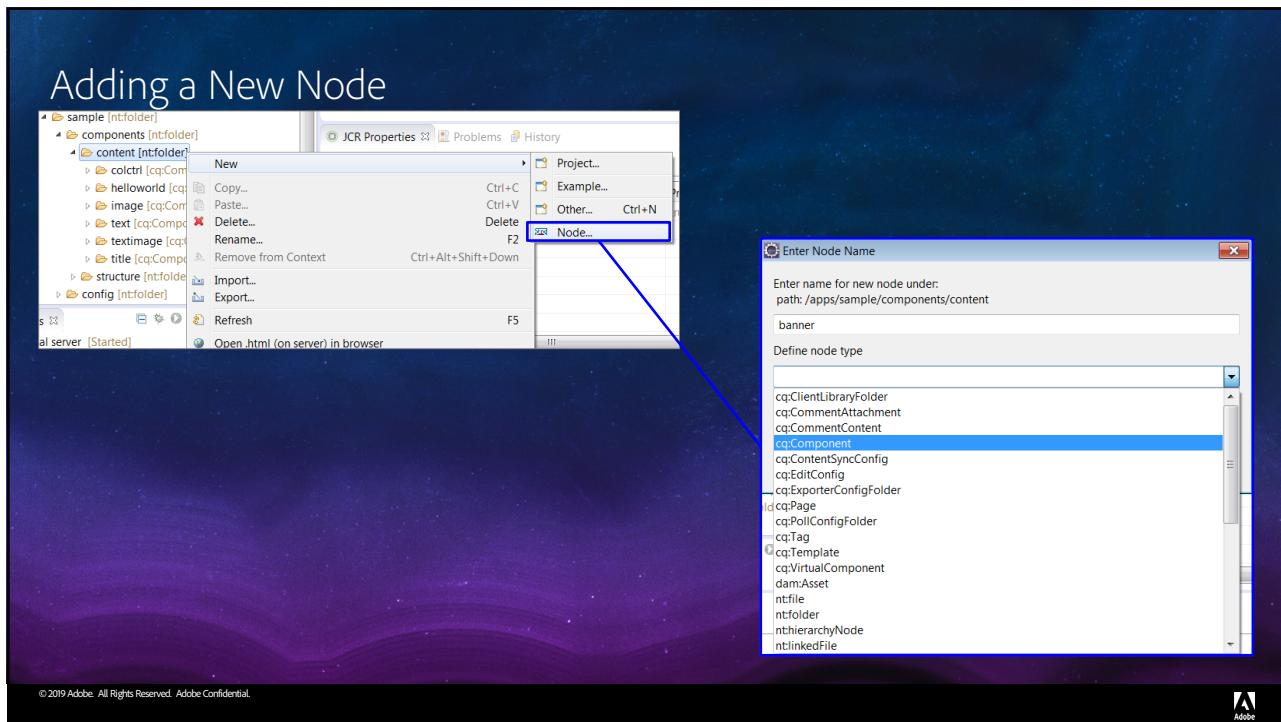


© 2019 Adobe. All Rights Reserved. Adobe Confidential.

38



39

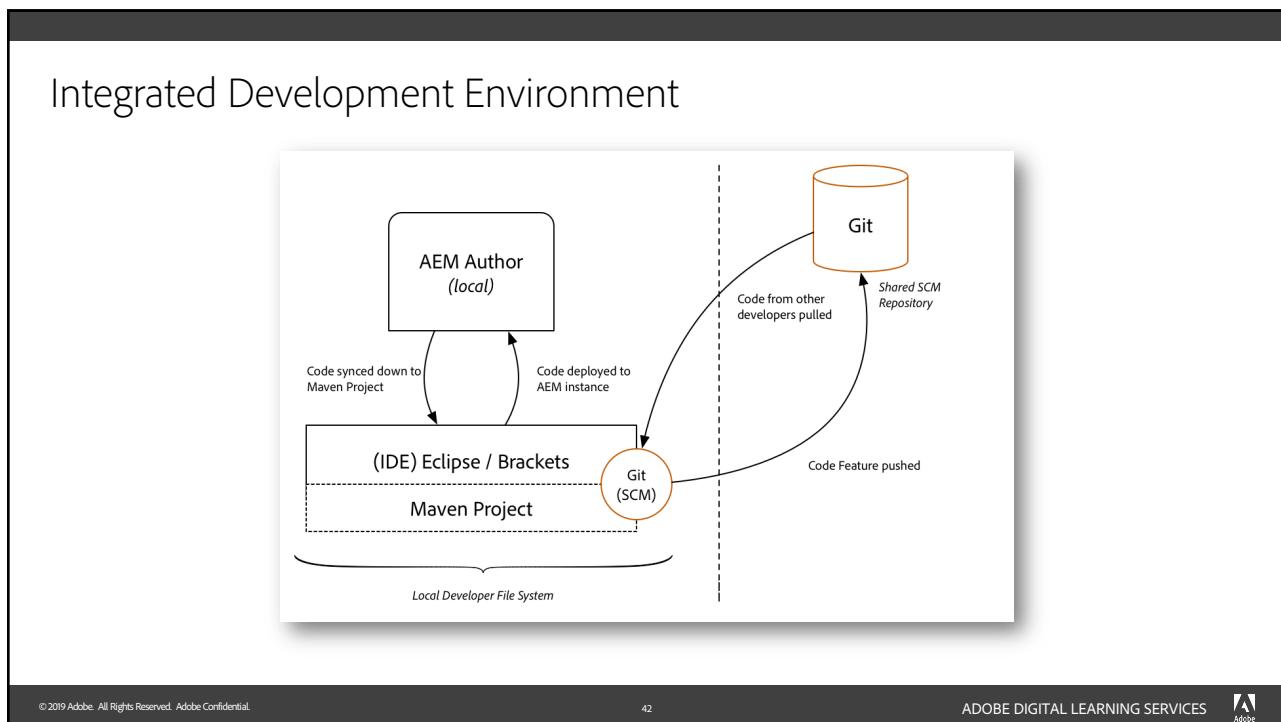


40

Adding or Editing a Node Property

The screenshot shows the AEM authoring interface. On the left is a tree view of the site structure under 'sample [nt:folder]'. In the center, the 'Editor' window is open. The 'Source' tab is selected, showing XML properties for a node named 'banner'. The 'Node' section includes properties like 'version="1.0" encoding="UTF-8"', 'jcr:root', and 'jcr:title'. Below this, a table titled 'Node properties' lists two properties: 'jcr:primaryType' (cq:Component) and 'jcr:title' (Banner). The bottom of the interface shows tabs for 'JCR Properties', 'Problems', and 'History', along with a toolbar.

41



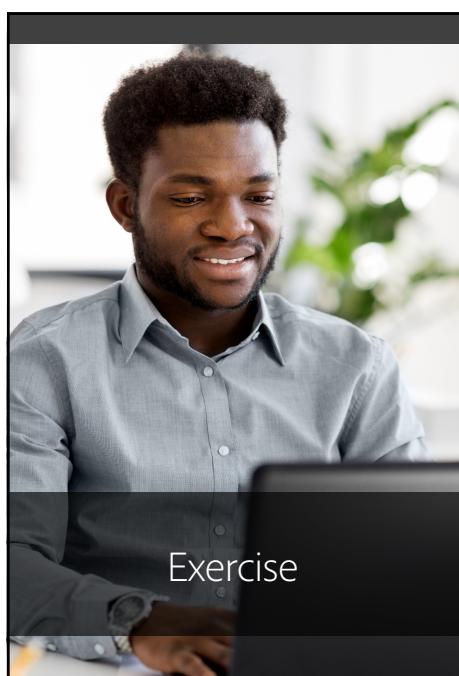
42

Adding or Editing a Node Property

The screenshot shows the AEM authoring interface. On the left is a tree view of the site structure under 'sample [nt:folder]'. A node named 'Node' is selected. The main area contains an 'Editor' window with tabs for 'Design' and 'Source'. The 'Source' tab is active, showing XML code for the selected node. Below the XML is a table titled 'Node properties' with two rows:

Name	Type	Value	Prote...	Mand...	Multiple	Auto Cr...
jcr:primaryType	Name	cq:Component	true	true	false	true
jcr:title	String	Banner		false	false	false

43



Exercise: Local Development Environment

In this exercise, you need to verify that the tools are installed on your desktop and available via the command line path.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Page 7

- Resource: <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react.html#local-dev-env>

Note: If using the ADLS virtual classroom, the Development Environment will be set up in advance. Your instructor will review and verify the environment with you.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

44

ADOBE DIGITAL LEARNING SERVICES



Apache Maven

Software project management and build tool.

Based on the concept of a project object model (POM) file:

- Manage a project's build, reporting, and documentation from a central piece of information

```
ware project management and build tool.

ed on the concept of a project object
del (POM) file:

Manage a project's build, reporting, and
documentation from a central piece of
information

[INFO] -----
[INFO] Reactor Summary:
[INFO]   WND Sites Project - Core ..... SUCCESS [ 12.641 s]
[INFO]   WND Sites Project - Core ..... SUCCESS [ 29.298 s]
[INFO]   WND Sites Project - UI and Data ..... SUCCESS [ 18.936 s]
[INFO]   WND Sites Project - Integration Tests ..... SUCCESS [ 1.587 s]
[INFO]   WND Sites Project - Integration Tests Launcher ..... SUCCESS [ 53.748 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] Total time: 03:00 min
[INFO] Finished at: 2018-09-13T10:22:22
[INFO] Final Memory: 48M/448M
[INFO] -----
[INFO] ----- [Apache/WND-Courses-site-wndmaven.war]
[INFO] Copyright 2018 Adobe Systems Incorporated
[INFO] License under the Apache License, Version 2.0 (the "License");
[INFO] you may not use this file except in compliance with the License.
[INFO] You may obtain a copy of the License at
[INFO] http://www.apache.org/licenses/LICENSE-2.0
[INFO] -----
[INFO] Unless required by applicable law or agreed to in writing, software
[INFO] distributed under the License is provided on an "AS IS" BASIS,
[INFO] WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
[INFO] See the License for the specific language governing permissions and
[INFO] limitations under the License.
[INFO] -----
[INFO] <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
[INFO]   <modelVersion>4.0.0</modelVersion>
[INFO]   <groupId>com.adobe.granite.archetypes</groupId>
[INFO]   <artifactId>adobe-guides-wnd</artifactId>
[INFO]   <version>1.0-SNAPSHOT</version>
[INFO]   <description>WND Sites Project</description>
[INFO]   <modules>
[INFO]     <module>core</module>
[INFO]     <module>ui</module>
[INFO]     <module>data</module>
[INFO]     <module>tests</module>
[INFO]     <module>launcher</module>
[INFO]   </modules>
[INFO]   <properties>
[INFO]     <scmCommentPrefix>[scmCommentPrefix]</scmCommentPrefix>
[INFO]     <scmPort>4242</scmPort>
[INFO]     <scmPubPort>4243</scmPubPort>
[INFO]     <scmPort>4243</scmPort>
[INFO]     <scmUser>admin</scmUser>
[INFO]     <scmPasswd>password</scmPasswd>
[INFO]     <projectBuildSourceEncoding>UTF-8</projectBuildSourceEncoding>
[INFO]     <projectReportingOutputEncoding>UTF-8</projectReportingOutputEncoding>
[INFO]   </properties>
[INFO]   <build>
[INFO]     <plugins>
[INFO]       <plugin> Maven Release Plugin </plugin>
[INFO]         <configuration>
[INFO]           <groupIds>org.apache.maven.plugins</groupIds>
[INFO]           <version>3.5.1</version>
[INFO]           <goals> release-repository </goals>
[INFO]           <includes>${scmCommentPrefix}*[scmCommentPrefix]</includes>
[INFO]           <goals> install </goals>
[INFO]           <includes>${scmCommentPrefix}*[scmCommentPrefix]</includes>
[INFO]         </configuration>
[INFO]       </plugin>
[INFO]     </plugins>
[INFO]   </build>
[INFO] 
```

© 2019 Adobe. All Rights Reserved. Adobe Confidential

45

 Adobe

45

Contents of a POM File

Sample POM file

POM files are XML files that contain the identity and structure of the project, build configuration, and other dependencies.

A POM file contains:

- General project information
 - Build settings
 - Build environment
 - POM relationships and dependencies
 - Snapshot versions
 - Property references
 - Plug-ins

```
<?xml version="1.0"?>
- <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.adobe.training</groupId>
  <artifactId>parent</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  - <properties>
    <company.name>Training Portal</company.name>
    <cq.host>localhost</cq.host>
    <cq.port>4502</cq.port>
    <cq.user>admin</cq.user>
    <cq.password>admin</cq.password>
    <module.prefix>company</module.prefix>
  </properties>
  <packaging>pom</packaging>
  <name>Company Portal</name>
  - <modules>
    <module>${module.prefix}-core</module>
    <module>${module.prefix}-ui</module>
  </modules>
  - <build>
    - <pluginManagement>
      - <plugins>
        - <plugin>
          <groupId>org.apache.maven.plugins</groupId>
```

© 2019 Adobe. All Rights Reserved. Adobe Confidential

A
10

46

Maven Archetype Project Structure

- **Parent POM** - deploys Maven modules and manages dependency versions.
- **core** - Java bundle containing all core functionality
 - OSGi services, listeners, or schedulers, servlets, filters
 - component-related Java code
- **ui.apps** - contains the /apps parts of the project
 - clientlibs, components, and configs
- **ui.content** - contains structural content and configurations (/content, /conf)
- **ui.tests** - Java bundle containing server-side JUnit tests
- **ui.launcher** - contains glue code that deploys **ui.tests bundle** and triggers remote JUnit execution
- **react-app*** - webpack project for the React application. Will become a Maven module.

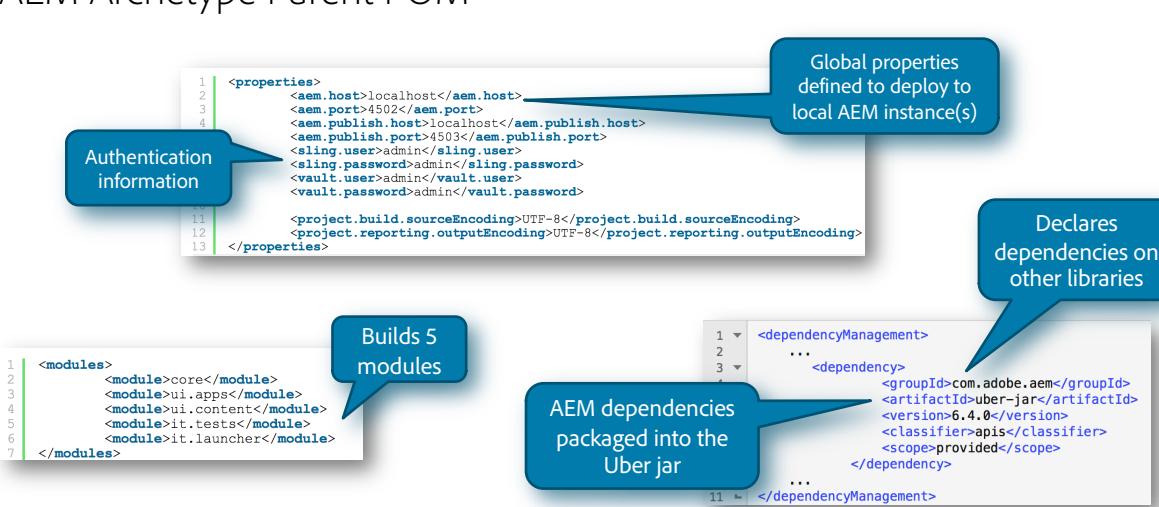
© 2019 Adobe. All Rights Reserved. Adobe Confidential.

47



47

AEM Archetype Parent POM



```

<properties>
  <aem.host>localhost</aem.host>
  <aem.port>4502</aem.port>
  <aem.publish.host>localhost</aem.publish.host>
  <aem.publish.port>4503</aem.publish.port>
  <sling.user>admin</sling.user>
  <sling.password>admin</sling.password>
  <vault.user>admin</vault.user>
  <vault.password>admin</vault.password>
</properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<modules>
  <module>core</module>
  <module>ui.apps</module>
  <module>ui.content</module>
  <module>it.tests</module>
  <module>it.launcher</module>
</modules>

```

Note: Authentication information can be overridden on command line so POM files do not have to change.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

48

ADORE DIGITAL LEARNING SERVICES 

48

UberJar

UberJar is a special JAR file provided by Adobe.

- Used to reduce the number of dependencies.

UberJar file contains:

- All public Java APIs exposed by AEM.
- Limited external libraries—all public APIs available in AEM
 - from Apache Sling, Apache Jackrabbit, Apache Lucene, Google Guava
 - two libraries used for image processing
- Interfaces and classes exported by an OSGi bundle in AEM
- MANIFEST.MF file with the correct package export versions for all the exported packages.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.


49

Using UberJar

To use UberJar, you must add the following elements to your pom.xml file.

- Dependency element
- Repository element

Dependency element

```
<dependency>
  <groupId>com.adobe.aem</groupId>
  <artifactId>uber-jar</artifactId>
  <version>6.4.0</version>
  <classifier>obfuscated-apis</classifier>
  <scope>provided</scope>
</dependency>
```

Repository element

```
<repositories>
  <repository>
    <id>adobe-public-releases</id>
    <name>Adobe Public Repository</name>

    <url>https://repo.adobe.com/nexus/content/groups/public/</url>
    <layout>default</layout>
  </repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <id>adobe-public-releases</id>
    <name>Adobe Public Repository</name>

    <url>https://repo.adobe.com/nexus/content/groups/public/</url>
    <layout>default</layout>
  </pluginRepository>
</pluginRepositories>
```

© 2019 Adobe. All Rights Reserved. Adobe Confidential.


50

Core Module

core module includes all Java code for the implementation

```
//core/pom.xml
<plugins>
  ...
  <plugin>
    <groupId>org.apache.felix</groupId>
    <artifactId>maven-bundle-plugin</artifactId>
    <extensions>true</extensions>
    <configuration>
      ...
      <instructions>
        <!-- Import any version of javax.inject, to allow running on multiple versions of AEM -->
        <Import-Package>javax.inject;version=0.0.0,*</Import-Package>
        <Sling-Model-Packages>
          com.adobe.aem.guides.wkndevents.core
        </Sling-Model-Packages>
      </instructions>
    </configuration>
  </plugin>
  ...
</plugins>
```

Maven sling plugin allows the core bundle to be deployed without deploying ui.apps and ui.content

```
//core/pom.xml
<plugin>
  <groupId>org.apache.sling</groupId>
  <artifactId>maven-sling-plugin</artifactId>
</plugin>
```

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

51

ADOBEDIGITAL LEARNING SERVICES 

UI.apps Module

UI.apps module includes all rendering code beneath /apps

package manager group

Embedded, compiled core bundle to be included in the package

```
//ui.apps/pom.xml
<!-- ===== V A U L T P A C K A G E P L U G I N -->
<!-- =====>
<plugin>
  <groupId>com.day.jcr.vault</groupId>
  <artifactId>content-package-maven-plugin</artifactId>
  <extensions>true</extensions>
  <configuration>
    <filterSource>src/main/content/META-INF/vault/filter.xml</filterSource>
    <verbose>true</verbose>
    <failOnErrors>true</failOnErrors>
    <group>aem-guides/wknd-events</group>
    <embedded>
      <embedded>
        <groupId>com.adobe.aem.guides</groupId>
        <artifactId>aem-guides-wknd-events.core</artifactId>
        <target>/apps/wknd-events/install</target>
      </embedded>
    </embedded>
    <subPackages>
      <subPackage>
        <groupId>com.adobe.cq</groupId>
        <artifactId>core.wcm.components.all</artifactId>
        <filter>true</filter>
      </subPackage>
    </subPackages>
  </configuration>
</plugin>
```

Content package maven plugin creates a content package that can be deployed to AEM

filterSource points to filter.xml that defines the jcr paths included in the package

jcr path of core bundle

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

52

ADOBEDIGITAL LEARNING SERVICES 

UI.content Module

UI.content module contains content beneath /content and configurations defined in /conf

package manager group

```

<!-- ===== V A U L T   P A C K A G E   P L U G I N -->
<plugin>
    <groupId>com.day.jcr.vault</groupId>
    <artifactId>content-package-maven-plugin</artifactId>
    <extensions>true</extensions>
    <configuration>
        <filterSource>src/main/content/META-INF/vault/filter.xml</filterSource>
        <verbose>true</verbose>
        <failOnErrors>true</failOnErrors>
        <group>aem-guides/wknd-events</group>
        <properties>
            <acHandling>merge_preserve</acHandling>
        </properties>
    </configuration>
</plugin>

```

compiled into a content package

filterSource points to filter.xml that defines the jcr paths included in the package

Preserves access control information for editable templates

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

53

ADOBEDIGITAL LEARNING SERVICES



filter.xml

defines jcr paths included in the package

mode=merge keeps package deployment from overwriting changes made in AEM directly

```

<?xml version="1.0" encoding="UTF-8"?>
<workspaceFilter version="1.0">
    <filter root="/conf/wknd-events"/>
    <filter root="/content/wknd-events" mode="merge"/>
    <filter root="/content/dam/wknd-events" mode="merge"/>
</workspaceFilter>

```

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

54

ADOBEDIGITAL LEARNING SERVICES



Core Components Included in Maven Build

```

dependency defined in parent pom
1 <dependencies>
2 ...
3 <dependency>
4   <groupId>com.adobe.cq</groupId>
5   <artifactId>core.wcm.components.all</artifactId>
6   <type>zip</type>
7   <version>2.0.4</version>
8 </dependency>
9 ...
10 </dependencies>
11 ...
12

//ui.apps/pom.xml
1 <dependencies>
2 ...
3 <dependency>
4   <groupId>com.adobe.cq</groupId>
5   <artifactId>core.wcm.components.all</artifactId>
6   <type>zip</type>
7 </dependency>
8 ...
9 </dependencies>
10 ...
11 ...
12

```

```

Included as a subPackage in UI.apps pom
1 //ui.apps/pom.xml
2 <!---- V A U L T P A C K A G E P L U G I N ---->
3 <!-- V A U L T P A C K A G E P L U G I N -->
4 <!-- V A U L T P A C K A G E P L U G I N -->
5 <!-- V A U L T P A C K A G E P L U G I N -->
6 <!-- V A U L T P A C K A G E P L U G I N -->
7 <!-- V A U L T P A C K A G E P L U G I N -->
8 <!-- V A U L T P A C K A G E P L U G I N -->
9 <!-- V A U L T P A C K A G E P L U G I N -->
10 <!-- V A U L T P A C K A G E P L U G I N -->
11 <!-- V A U L T P A C K A G E P L U G I N -->
12 <!-- V A U L T P A C K A G E P L U G I N -->
13 <!-- V A U L T P A C K A G E P L U G I N -->
14 <!-- V A U L T P A C K A G E P L U G I N -->
15 <!-- V A U L T P A C K A G E P L U G I N -->
16 <!-- V A U L T P A C K A G E P L U G I N -->
17 <!-- V A U L T P A C K A G E P L U G I N -->
18 <!-- V A U L T P A C K A G E P L U G I N -->
19 <!-- V A U L T P A C K A G E P L U G I N -->
20 <!-- V A U L T P A C K A G E P L U G I N -->
21 <!-- V A U L T P A C K A G E P L U G I N -->
22 <!-- V A U L T P A C K A G E P L U G I N -->
23 <!-- V A U L T P A C K A G E P L U G I N -->
24 <!-- V A U L T P A C K A G E P L U G I N -->
25 <!-- V A U L T P A C K A G E P L U G I N -->
26 <!-- V A U L T P A C K A G E P L U G I N -->
27 <!-- V A U L T P A C K A G E P L U G I N -->
28 <!-- V A U L T P A C K A G E P L U G I N -->
29 <!-- V A U L T P A C K A G E P L U G I N -->
30 </plugin>

```

dependency defined in UI.apps pom

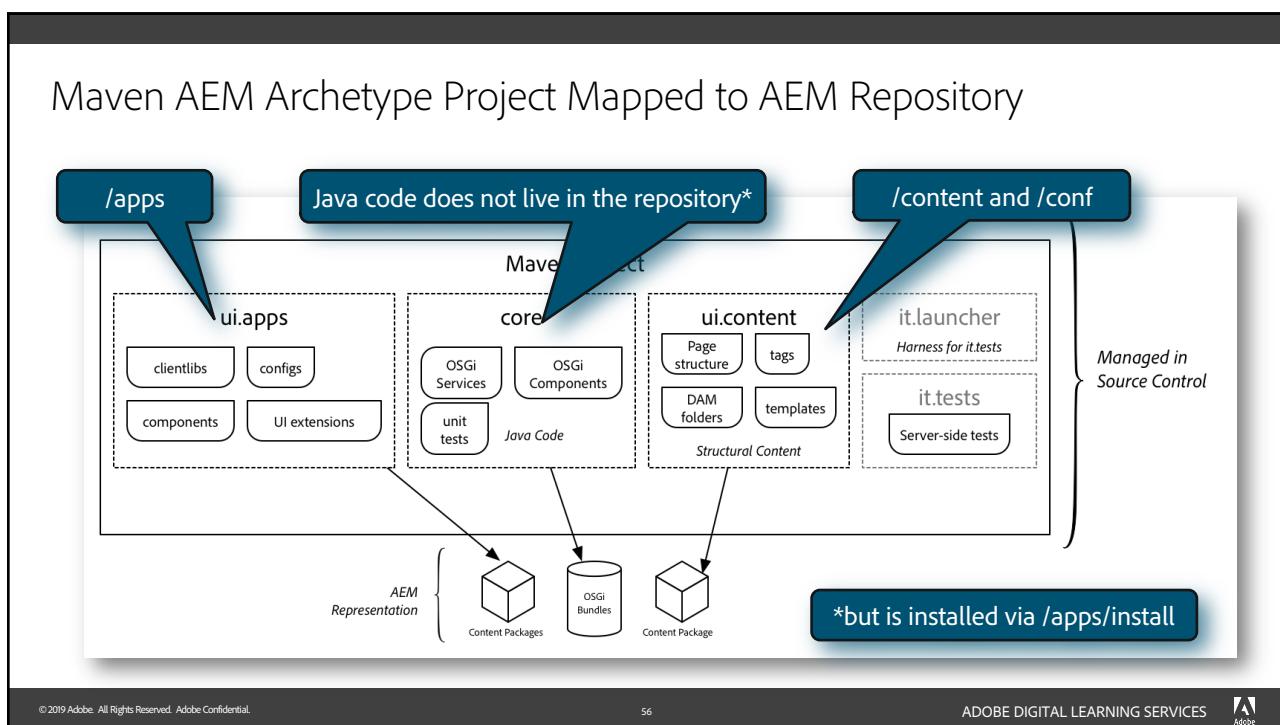
© 2019 Adobe. All Rights Reserved. Adobe Confidential.

55

ADORE DIGITAL LEARNING SERVICES



55



© 2019 Adobe. All Rights Reserved. Adobe Confidential.

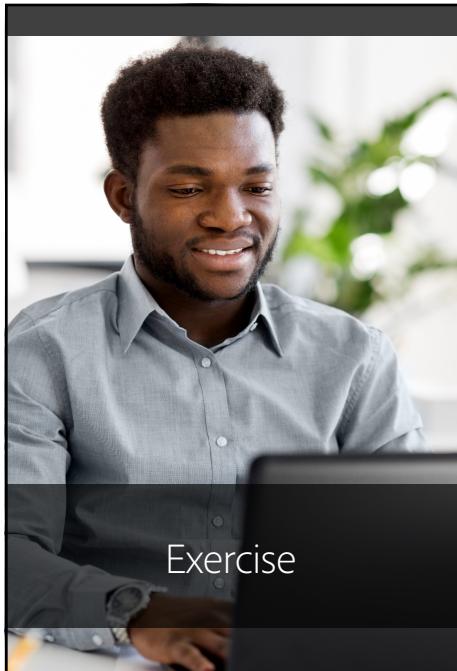
56

ADORE DIGITAL LEARNING SERVICES



56

26



AEM Project

In this exercise, you will download and investigate the SPA starter project.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

- Resource: <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-0.html#starter-project>

Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

57

ADOBEDIGITAL LEARNING SERVICES 

Maven Repositories

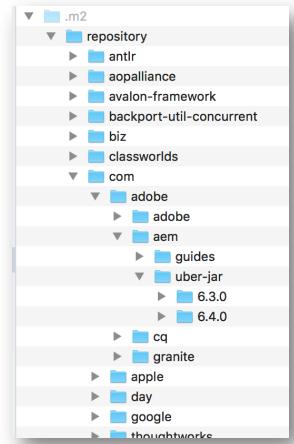
A Maven repository is a directory that houses project jars, library jar, plugins, or any other project-specific artifacts to be used in a Maven project.

Two types of repositories: local and remote

- Local repository refers to a copy on your own installation that is a cache of the remote downloads.
 - Local repository resides in the .m2 directory in your user home (for example: \${user.home}/.m2/repository).
- Remote repositories refer to any other type of repository, accessed by a variety of protocols.

Default maven repository:
<http://repo.maven.apache.org/maven2/>

Adobe repository (Adobe Public Profile in the POM file):
<https://repo.adobe.com/nexus/content/groups/public>



© 2019 Adobe. All Rights Reserved. Adobe Confidential.

58

ADOBEDIGITAL LEARNING SERVICES 

settings.xml

Contains elements used to define values that configure Maven execution in various ways, like the pom.xml, but are not associated with any specific project.

- Local repository location, alternate remote repository servers, authentication information

User/Project-specific settings.xml can be created in the local .m2 directory.

```

44 <!-- settings defined="http://maven.apache.org/SETTINGS/1.0.4"
45   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
46   xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.4 http://maven.apache.org/xsd/settings-1.0.4.xsd">
47   | The path to the local repository maven will use to store artifacts.
48   | Default: ${user.home}/.m2/repository
49   | LocalRepository</path/to/local/repo>localRepository>
50   |
51   |<!-- interactiveMode
52   | This will determine whether maven prompts you when it needs input. If set to false,
53   | maven will use a sensible default value, perhaps based on some other setting, for
54   | the parameter in question.
55   | Default: true
56   |<interactiveMode>true</interactiveMode>
57   |
58   |<!-- offline
59   | This will determine whether maven should attempt to connect to the network when executing a build.
60   | This will have an effect on artifact downloads, artifact deployment, and others.
61   | Default: false
62   |<offline>false</offline>
63   |
64   |<!-- pluginGroups
65   | This is a list of additional group identifiers that will be searched when resolving plugins by their prefix, i.e.
66   | when invoking a command line like "mvn prefix:goal". Maven will automatically add the group identifiers
67   | "org.apache.maven.plugins" and "org.codehaus.mojo" if these are not already contained in the list.
68   |<-->
69   |<pluginGroups>
70   |<!-- pluginGroup
71   | Specifies a further group identifier to use for plugin lookup.
72   |<!-- pluginGroup><!-->
73   |</pluginGroups>
74   |
75   |<!-- proxies
76   | This is a list of proxies which can be used on this machine to connect to the network.
77   | Unless otherwise specified (by system property or command-line switch), the first proxy
78   | in the configuration in this list marked as active will be used.
79   |<-->
80   |<proxies>
81   |<!-- proxy
82   | Specifies for one proxy, to be used in connecting to the network.
83   |<!-- proxy
84   |<!-- optional:id>
85   |<!-- active=true/active
86   |<!-- active=true/active
87   |<!-- active=true/active
88   |<!-- active=true/active
89   |<!-- active=true/active
90   |<!-- active=true/active
91   |<!-- active=true/active
92   |<!-- active=true/active
93   |<!-- active=true/active
94   |<!-- active=true/active
95   |<!-- active=true/active
96   |<!-- active=true/active
97   |<!-- active=true/active
98   |<!-- active=true/active
99   |<!-- active=true/active
100  |<!-- active=true/active
101  |<!-- active=true/active
102  |<!-- active=true/active
103  |<!-- active=true/active
104  |<!-- active=true/active
105  |<!-- active=true/active
106  |<!-- active=true/active
107  |<!-- active=true/active
108  |<!-- active=true/active
109  |<!-- active=true/active
110  |<!-- active=true/active
111  |<!-- active=true/active
112  |<!-- active=true/active
113  |<!-- active=true/active
114  |<!-- active=true/active
115  |<!-- active=true/active
116  |<!-- active=true/active
117  |<!-- active=true/active
118  |<!-- active=true/active
119  |<!-- active=true/active
120  |<!-- active=true/active
121  |<!-- active=true/active
122  |<!-- active=true/active
123  |<!-- active=true/active
124  |<!-- active=true/active
125  |<!-- active=true/active
126  |<!-- active=true/active
127  |<!-- active=true/active
128  |<!-- active=true/active
129  |<!-- active=true/active
130  |<!-- active=true/active
131  |<!-- active=true/active
132  |<!-- active=true/active
133  |<!-- active=true/active
134  |<!-- active=true/active
135  |<!-- active=true/active
136  |<!-- active=true/active
137  |<!-- active=true/active
138  |<!-- active=true/active
139  |<!-- active=true/active
140  |<!-- active=true/active
141  |<!-- active=true/active
142  |<!-- active=true/active
143  |<!-- active=true/active
144  |<!-- active=true/active
145  |<!-- active=true/active
146  |<!-- active=true/active
147  |<!-- active=true/active
148  |<!-- active=true/active
149  |<!-- active=true/active
150  |<!-- active=true/active
151  |<!-- active=true/active
152  |<!-- active=true/active
153  |<!-- active=true/active
154  |<!-- active=true/active
155  |<!-- active=true/active
156  |<!-- active=true/active
157  |<!-- active=true/active
158  |<!-- active=true/active
159  |<!-- active=true/active
160  |<!-- active=true/active
161  |<!-- active=true/active
162  |<!-- active=true/active
163  |<!-- active=true/active
164  |<!-- active=true/active
165  |<!-- active=true/active
166  |<!-- active=true/active
167  |<!-- active=true/active
168  |<!-- active=true/active
169  |<!-- active=true/active
170  |<!-- active=true/active
171  |<!-- active=true/active
172  |<!-- active=true/active
173  |<!-- active=true/active
174  |<!-- active=true/active
175  |<!-- active=true/active
176  |<!-- active=true/active
177  |<!-- active=true/active
178  |<!-- active=true/active
179  |<!-- active=true/active
180  |<!-- active=true/active
181  |<!-- active=true/active
182  |<!-- active=true/active

```

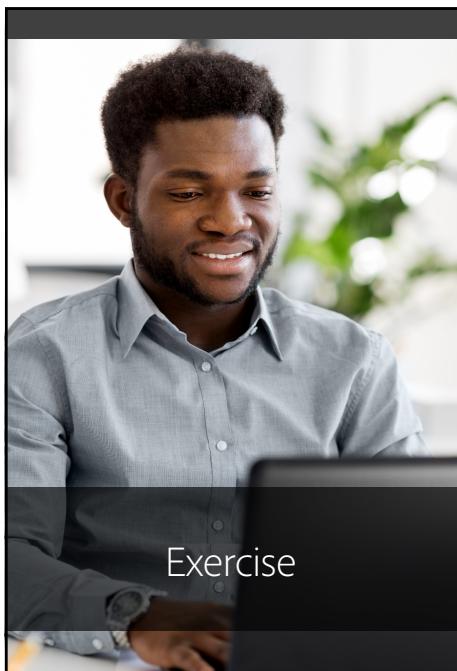
© 2019 Adobe. All Rights Reserved. Adobe Confidential.

59

ADobe DIGITAL LEARNING SERVICES



59



AEM Project

In this exercise, you will build, deploy, and verify the deployment containing structural content and configurations.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 11 – 13

- Resource: <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-0.html#inspect>

Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

60

ADobe DIGITAL LEARNING SERVICES



60

28

Prerequisites: Integrated Development Environment (IDE)

Tools

- Eclipse IDE with AEM Developer Tools
- IntelliJ with AEM IDE Tooling
- Visual Studio – AEM Sync
- AEM Repo Tool

Node.js

- Asynchronous, event-driven JavaScript runtime
- Executes JavaScript outside of a browser
 - Command line tools
 - Server-side scripting to produce dynamic webpage content
- Cross platform
- Open-source



npm

Default package manager for Node.js.

Consists of three components:

- Website – discover packages, set up profiles, and manage npm.
- Command line interface (CLI) – interact with npm.
- Registry – public database of JavaScript (JS) software and meta-information.

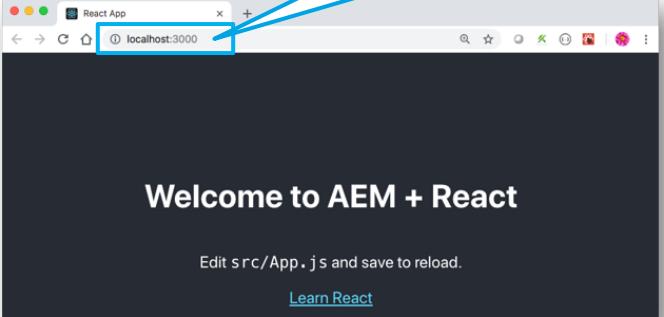
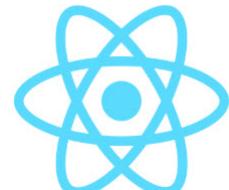


© 2019 Adobe. All Rights Reserved. Adobe Confidential. 63 ADOBE DIGITAL LEARNING SERVICES 

63

ReactJS

React app on a local development server, port:3000

JavaScript Library for building user interfaces.

- Maintained by Facebook.
- Base for development of SPA or Mobile applications.

© 2019 Adobe. All Rights Reserved. Adobe Confidential. 64 ADOBE DIGITAL LEARNING SERVICES 

64

React build

Files created by **npm run build**

```

    aem-guides-wknd-events
    ├── ui.content
    ├── ui.apps
    ├── core
    └── react-app
        └── build
            ├── static
            └── js
                ├── main.js.map
                ├── main.css.map
                ├── index.html
                └── manifest.json

```

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

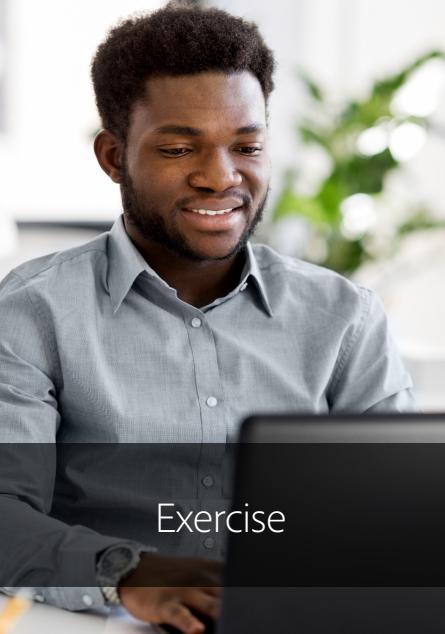
65

ADOBEDIGITAL LEARNING SERVICES



65

React App

Exercise

In this exercise, you will run the React app locally. Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 14 - 15

- Resource: <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-0.html#react-app>

Note: The step-by-step instructions are available in your Student Guide.

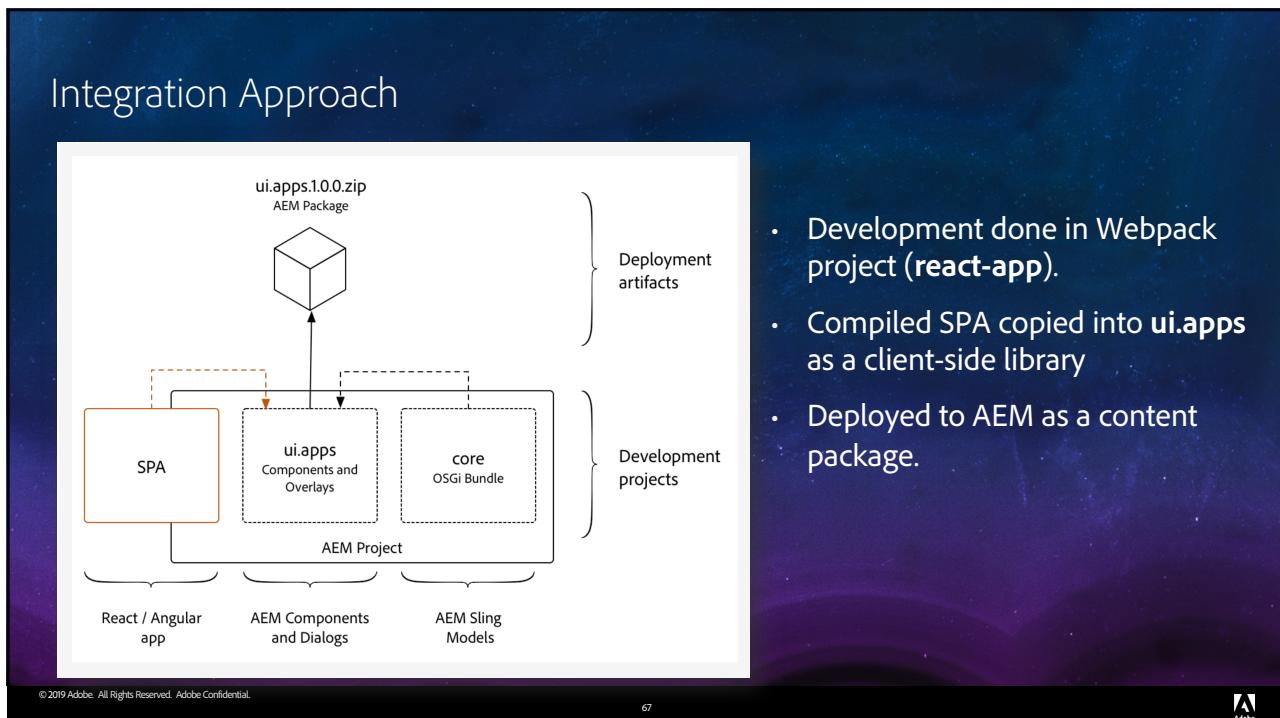
© 2019 Adobe. All Rights Reserved. Adobe Confidential.

66

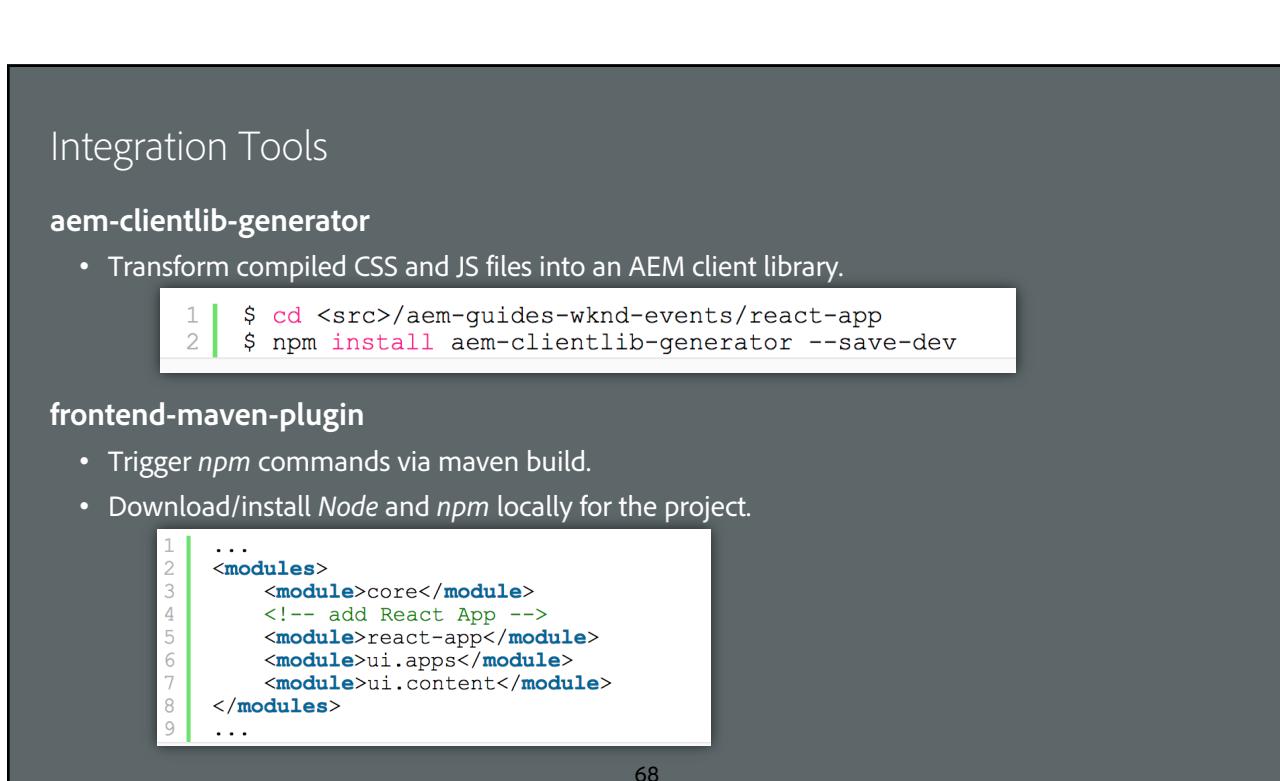
ADOBEDIGITAL LEARNING SERVICES



66



67



68

Configure *aem-clientlibs-generator*

```

1 | $ cd <src>/aem-guides-wknd-events/react-app
2 | $ npm install aem-clientlib-generator --save-dev

```



```

1 //package.json
2 {
3   ...
4
5   "devDependencies": {
6     "aem-clientlib-generator": "^1.4.1"
7   }
8 }

```

react-app/package.json
updated with
dependencies for aem-clientlib-generator

69

Configure client libraries - */react-app/clientlib.config.js*

```

1 module.exports = {
2   // default working directory (can be changed per 'cwd'
3   context: __dirname,
4
5   // path to the clientlib root folder (output)
6   clientLibRoot: "../../ui.apps/src/main/content/jcr_root",
7
8   libs: {
9     name: "react-app",
10    allowProxy: true,
11    categories: ["wknd-events.react"],
12    serializationFormat: "xml",
13    jsProcessor: ["min:gcc"],
14    assets: {
15      js: [
16        "build/static/**/*.*js"
17      ],
18      css: [
19        "build/static/**/*.*css"
20      ]
21    }
22  }
23};

```

Create a client library in the ui.apps module at the specific path

Specify client library category

Include all .css and .js files beneath react-app/build/static

70

Configure *front-end-maven-plugin*

```

1  ...
2  <properties>
3    <aem.host>localhost</aem.host>
4    <aem.port>4502</aem.port>
5    <aem.publish.host>localhost</aem.publish.host>
6    <aem.publish.port>4503</aem.publish.port>
7    <sling.user>admin</sling.user>
8    <sling.password>admin</sling.password>
9    <vault.user>admin</vault.user>
10   <vault.password>admin</vault.password>
11
12  <!-- Update: Used by frontend-maven-plugin -->
13  <frontend-maven-plugin.version>1.6</frontend-maven-plugin.version>
14  <node.version>v10.8.0</node.version>
15  <npm.version>6.2.0</npm.version>
16  <!-- end update -->
17
18  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
19  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
20
21 ...

```



```

1 ...
2 <modules>
3   <module>core</module>
4   <!-- add React App -->
5   <module>react-app</module>
6   <module>ui.apps</module>
7   <module>ui.content</module>
8 </modules>
9 ...

```

Specify front-end-maven-plugin, node and npm versions in parent reactor POM

In parent reactor POM, add react-app as a module to build

71

react-app Module POM file

```

1 //react-app/pom.xml
2
3 ...
4 <!-- ===== BUILD DEFINITION ===== -->
5 <build>
6   <plugins>
7     <plugin>
8       <groupId>com.github.eirslett</groupId>
9       <artifactId>frontend-maven-plugin</artifactId>
10      <version>${frontend-maven-plugin.version}</version>
11
12      <executions>
13        <execution>
14          <id>install node and npm</id>
15          <goals>
16            <goal>install-node-and-npm</goal>
17          </goals>
18          <configuration>
19            <nodeVersion>${node.version}</nodeVersion>
20            <npmVersion>${npm.version}</npmVersion>
21          </configuration>
22        </execution>
23        <execution>
24          <id>npm install</id>
25          <goals>
26            <goal>npm</goal>
27          </goals>
28          <!-- Optional configuration which provides for running any npm command -->
29          <configuration>
30            <arguments>install</arguments>
31          </configuration>
32        </execution>
33        <execution>
34          <id>pm run build</id>
35          <goals>
36            <goal>run</goal>
37          </goals>
38          <configuration>
39            <arguments>run build</arguments>
40          </configuration>
41        </execution>
42      </executions>
43    </plugin>
44  </plugins>
45 </build>
46 ...

```

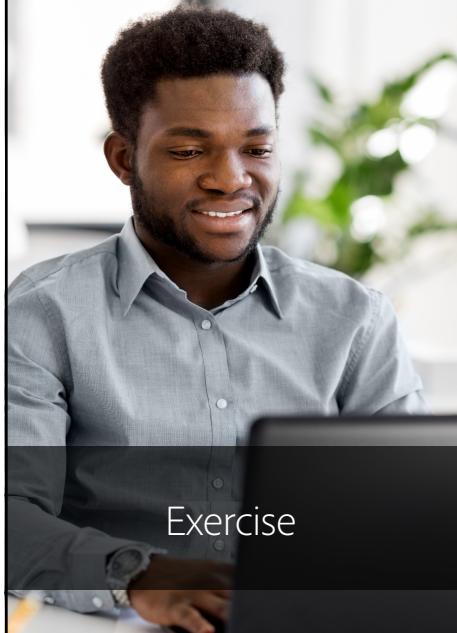
Front-end-maven-plugin defintion

Install npm

Configure node and npm

Run npm build

72



React App

In this exercise, you will run the React app locally. Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 14 - 15

- Resource: <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-0.html#react-app>

Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

73

ADOBEDIGITAL LEARNING SERVICES 

Sites vs SPA

Sites built using common SPA frameworks such as React and Angular.

- load their content via dynamic JSON
- do not provide the HTML structure necessary for the AEM Page Editor to be able to place edit controls

Enabling editing of SPAs within AEM requires mapping between JSON output of the SPA and the content model in the AEM repository.

SPA support in AEM introduces a thin JS layer that interacts with the SPA JS code when loaded in the Page Editor:

- events can be sent
- location for the edit controls can be activated to allow in-context editing

Builds upon the Content Services API Endpoint concept.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

74

ADOBEDIGITAL LEARNING SERVICES 

Page Component Differences

Page component for a SPA doesn't provide the HTML elements of its child components via the JSP or HTL file.

- Operations delegated to the SPA framework

Representation of child components or model fetched as a JSON data structure from the repository.

SPA components added to the page using this structure.

Page Model Management

Resolution and the management of the page model is delegated to a provided **PageModel** library.

- Provided indirectly to the AEM Page component via the **cq-react-editable-components** npm module
- Page Model is an interpreter between AEM and the SPA and therefore always must be present

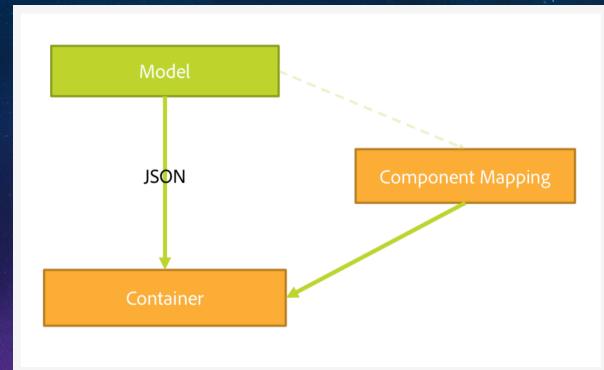
SPA must use the Page Model library in order to be initialized and be authored by the SPA Editor.

An additional library **cq.authoring.pagemodel.messaging** must be added in order to enable the communication with the Page Editor.

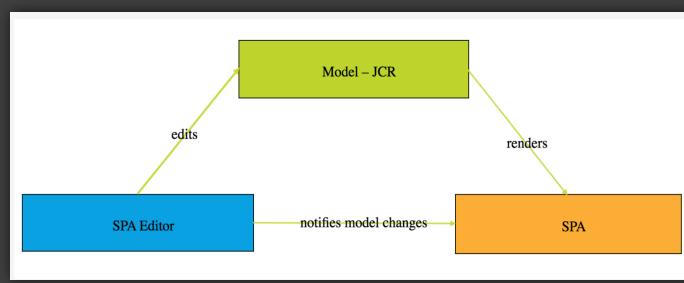
Page Model Management (cont.)

For each resource in the exported model, the SPA maps an actual component that will do the rendering.

The model, represented as JSON, is then rendered using the component mappings within a container.



Communication Data Type



cq.authoring.pagemodel.messaging
client library sends a message to the
Page Editor to establish the JSON
communication data type.

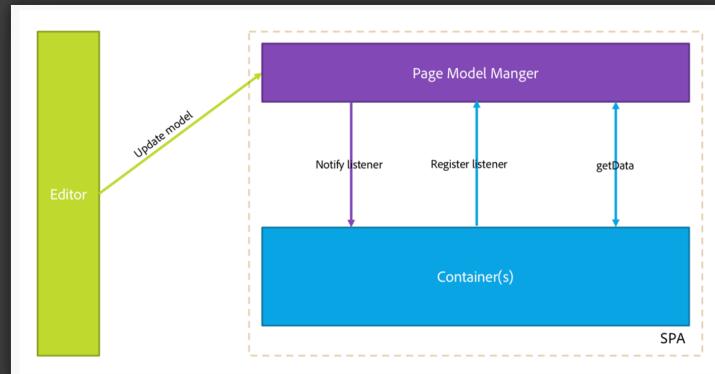
When the communication data type is
set to JSON, GET requests will
communicate with the Sling Model
end-points of a component.

After an update occurs in the page editor:

- A JSON representation of the updated component is sent to the Page Model library.
- Page Model library informs the SPA of updates.

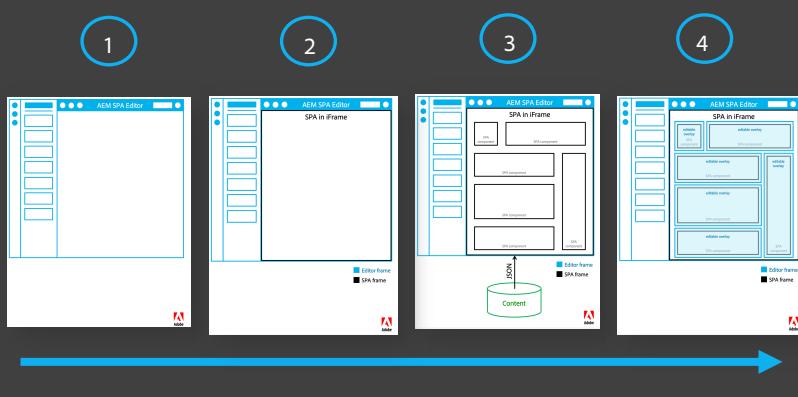
Flow of Interaction Between SPA and AEM

- SPA Editor Framework is mediator.
- Communication between Page Editor and SPA is JSON, not HTML.
- Page Editor provides the latest version of the page model to the SPA via iframe and messaging API.
- Page Model Manager notifies editor that it is ready and passes page model as a JSON structure.



Editor does not alter or access the DOM structure of the page being authored, rather it provides the latest page model.

How Editing a SPA appears to the Author



1. SPA Editor loads.
2. SPA is loaded in a separate frame.
3. SPA requests JSON content and renders components client-side.
4. SPA Editor detects rendered components and generates overlays.

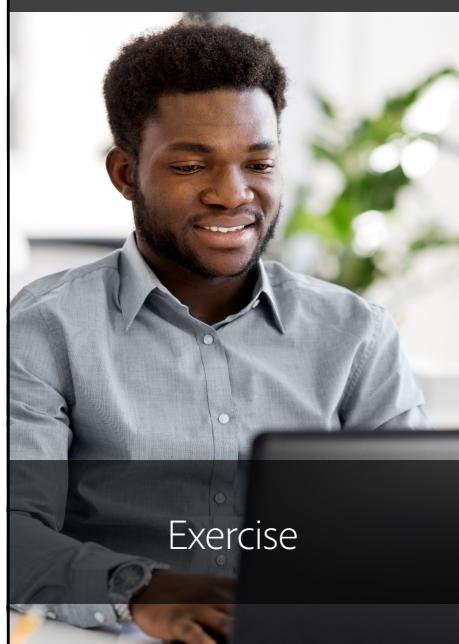
How Editing a SPA appears to the Author (cont.)

5. Author clicks overlay, displaying the component's edit toolbar.
6. SPA Editor persists edits with a POST request to the server.
7. SPA Editor requests updated JSON to the SPA Editor, which is sent to the SPA with a DOM Event.
8. SPA re-renders the concerned component, updating its DOM.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

81

ADOBEDIGITAL LEARNING SERVICES



Configure the frontend-maven-plugin

In this exercise, you will

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

pages 20 – 22

NOTE: ERROR; on page 22, step 4: This must be done from the main parent project folder – one step ABOVE the react-app folder

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-0.html#frontend-maven-plugin>

Note: The step-by-step instructions are available in your Student Guide.

Exercise

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

82

ADOBEDIGITAL LEARNING SERVICES

Integrating the React App on the Page – *customheaderlibs.html*

```

1  <!--/*
2   Custom Headerlibs for React Site
3  *-->
4  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
5  <meta property="cq:datatype" data-sly-test="${wcmmode.edit || wcmmode.preview}" content="JSON"/>
6  <meta property="cq:wcmode" data-sly-test="${wcmmode.edit}" content="edit"/>
7  <meta property="cq:wcmode" data-sly-test="${wcmmode.preview}" content="preview"/>
8  <meta property="cq:pagemodel_root_url"
9    data-sly-use.page="com.adobe.aem.guides.wkndevents.core.models.HierarchyPage"
10   content="${page.rootUrl}"/>
11  <sly data-sly-use.clientlib="/libs/granite/sightly/templates/clientlib.html">
12  <sly data-sly-call="${clientlib.css @ categories='wknd-events.react'}"/>
```

Used by AEM SPA editor to identify root page JSON

Load wknd-events.react client library

84

Integrating the React App on the Page – *customfooterlibs.html*

```

1  <!--/*
2   Custom footer React libs
3  *-->
4  <sly data-sly-use.clientLib="${'/libs/granite/sightly/templates/clientlib.html'}"></sly>
5  <sly data-sly-test="${wcmmode.edit || wcmmode.preview}"
6    data-sly-call="${clientLib.js @ categories='cq.authoring.pagemodel.messaging'}"></sly>
7  <sly data-sly-call="${clientLib.js @ categories='wknd-events.react'}"></sly>
```

Load wknd-events.react client library

Allows SPA editing capabilities using the AEM Sites editor

85

Integrating the React App on the Page – *body.html*

Page-rendering component's ***body.html*** contains the DOM element targeted by ***react-app/src/index.js***.

```

page component body.html
1  <!--/*
2   - body.html
3   - includes div that will be targeted by SPA
4   */-->
5   <div id="root"></div>

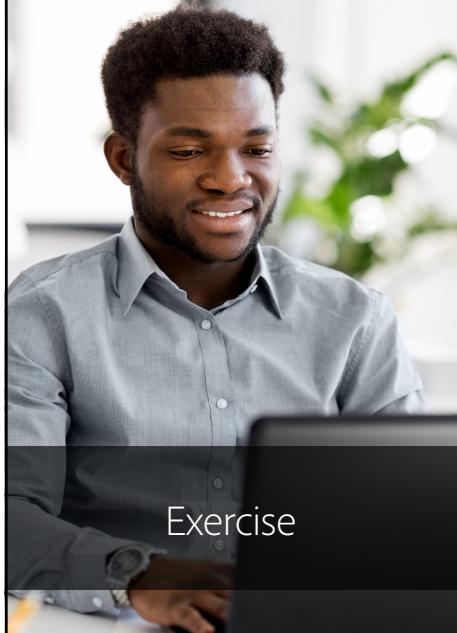
//react-app/index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5
6 ReactDOM.render(<App />, document.getElementById('root'));

```

86

SPA App in the AEM SPA Editor

87



Integrate the React app on the Page

In this exercise, you will integrate the React app onto the page via the client.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

pages 23 - 24

- Resources: <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-0.html#integrate-react-app>

Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

88

ADOBEDIGITAL LEARNING SERVICES 

Resources

Hands-on Activity

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-0.html>

Apache Maven

- <https://maven.apache.org/what-is-maven.html>
- <https://maven.apache.org/guides/introduction/introduction-to-repositories.html>
- <https://maven.apache.org/settings.html>
- <https://maven.apache.org/repository/index.html>
- <https://helpx.adobe.com/experience-manager/kb/SetUpTheAdobeMavenRepository.html>

Additional reading

- [AEM-hybrid-architecture-wp 1.8.19 white paper](#)

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

89

ADOBEDIGITAL LEARNING SERVICES 

Resources (cont.)

Building AEM Projects with templates

- <https://helpx.adobe.com/experience-manager/6-4/sites/developing/using/ht-projects-maven.html>
- <https://github.com/Adobe-Consulting-Services/lazybones-aem-templates>

Uber jar

- <https://helpx.adobe.com/experience-manager/6-4/sites/developing/using/ht-projects-maven.html#ExperienceManagerAPIDependencies>

IDEs

- <https://eclipse.adobe.com/aem/dev-tools/>
- <https://docs.adobe.com/docs/en/dev-tools/aem-brackets.html>
- <https://helpx.adobe.com/experience-manager/using/creating-aem-project-using-intellij.html>

Node.js

- <https://nodejs.org/en/>

NPM

- <https://www.npmjs.com/>

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

90

Adobe

90

Module 3: Editable Components



• Install the AEM SPA Editor JS SDK and implement two React components that map to AEM Text and Image components

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

91

AEM SPA Editor JS SDK

Collection of JS libraries.

Framework for editing contents of a SPA deployed in AEM.

AEM delivers content as JSON and SPA Editor JS SDK maps the JSON to React components.

Available via three **npm** modules:

- [`@adobe/cq-spa-component-mapping`](#) - provides helpers to map AEM Components to SPA components. This module is not tied to a specific SPA framework.
- [`@adobe/cq-spa-page-model-manager`](#) - provides the API to manage the model representation of the AEM Pages that are used to compose a SPA. This module is not tied to a specific SPA framework.
- [`@adobe/cq-react-editable-components`](#) - provides generic React helpers and components to support AEM authoring. This module also wraps the `cq-spa-page-model-manager` and `cq-spa-component-mapping` to make these available to the React framework.

AEM Page Editor and the SPA

Goal:

- AEM Page Editor edits the data exposed by a SPA framework.

Requirement:

- Project must be able to interpret the structure of the data stored for an application within the AEM repository.

Solution:

- Two framework-agnostic libraries are provided:
 - **PageModelManager**
 - **ComponentMapping**

Page Model

Content structure of a page is stored in AEM.

- Map and incorporate SPA components
- Leverage the JSON Model Exporter

SPA developers create SPA components.

- Map to AEM components
- Use the resource type (or path to the AEM component) as a unique key

SPA components must be:

- In-sync with the page model
- Updated with any changes to its content

A pattern leveraging dynamic components must be used to instantiate components on-the-fly following the provided page model structure (more on this later).

PageModelManager

Library provided as an NPM package.

Serves as a data model manager.

Abstracts the retrieval and management of the JSON structure that represents the actual content structure.

Responsible for syncing with the SPA to let it know when it has to re-render its components.

PageModelManager Initialization

1. Browser loads page.
2. Initialize ***PageModelManager***.
3. Initialize the root model of the App (via parameter, meta property, or current URL).
4. If the library identifies that the model of the current page is not part of the root model it fetches and includes it as the model of a child page.
5. Broadcast that the model is initialized.
6. Notify the listeners.
7. Return the root model.

Meta fields

Exportable sling models expose the following list of fields in order to enable the underlying libraries to interpret the data model.

- **:type:** Type of the AEM resource (default = resource type).
- **:children:** Hierarchical children of the current resource. Children are not part of the inner content of the current resource (can be found on items representing a page).
- **:hierarchyType:** Hierarchical type of a resource. The **PageModelManager** currently supports the page type.
- **:items:** Child content resources of the current resource (nested structure, only present on containers).
- **:itemsOrder:** Ordered list of the children. The JSON map object does not guarantee the order of its fields. By having both the map and the current array, the consumer of the API has the benefits of both structures.
- **:path:** Content path of an item (present on items representing a page).

ComponentMapping

Library provided as an NPM package.

Stores front-end components.

Provides a way for the SPA to map front-end components to AEM resource types:

- Enables a dynamic resolution of components when parsing the JSON model of the application.
- Each item present in the model contains a **:type** field that exposes an AEM resource type.

Front-end component can render itself using the fragment of model it has received from the underlying libraries.

Install AEM SPA Editor JS SDK

npm modules provided by Adobe include PageModelManger and ComponentMapping

Peer dependencies

Dev dependencies

```

1 //package.json
2
3 {
4   "name": "react-app",
5   "version": "0.1.0",
6   "private": true,
7   "dependencies": {
8     "@adobe/cq-react-editable-components": "^1.0.3",
9     "@adobe/cq-spa-component-mapping": "^1.0.3",
10    "@adobe/cq-spa-page-model-manager": "^1.0.1",
11    "react": "^16.5.2",
12    "react-dom": "^16.5.2",
13    "react-fast-compare": "^2.0.2",
14    "react-scripts": "2.0.4"
15  },
16  ...
17
18  "devDependencies": {
19    "aem-clientlib-generator": "^1.4.1",
20    "ajv": "^6.5.4",
21    "clone": "^2.1.2"
22  }
23}
```

Heart of the React App

Index.js - app entry point and includes CSS and the DOM to initialize the JSON model.

- Kicks off the **registerServiceWorker.js**.
 - Takes care of caching and updating files for the end-user.
 - Allows for offline capability and faster page loads after the initial visit.

App.js - handles view generation and component inclusion.

- Main app JavaScript component.
- Container for all other components.
- Written in JSX → how React adds XML syntax to JavaScript.
- Compiled to JavaScript at runtime.

Integrate AEM SPA Editor JS SDK – *react-app/src/index.js*

```

1 // src/index.js
2
3 import React from 'react';
4 import ReactDOM from 'react-dom';
5 import { ModelManager, Constants } from '@adobe/cq-spa-page-model-manager';
6 import App from './App';
7
8 function render(model) {
9   ReactDOM.render(
10     <App cqChildren={ model[Constants.CHILDREN_PROP] } cqItems={ model[Constants.ITEMS_PROP] } cqItemsOrder={ model[Constants.ITEMS_ORDER_PROP] } cqPath={ ModelManager.rootPath } locationPathname={ `window.location.pathname` } />,
11     document.getElementById('root')
12   );
13 }
14
15 ModelManager.initialize({ path: process.env.REACT_APP_PAGE_MODEL_PATH }).then(render);
16
17 
```

Initialize the App with AEM JSON Model

Integrate AEM SPA Editor JS SDK – *react-app/src/App.js*

```

1 // SRC/App.js
2 import React from 'react';
3 import { Page, withModel, EditorContext, Utils } from '@adobe/cq-react-editable-components';
4
5 // This component is the application entry point
6 class App extends Page {
7   render() {
8     return (
9       <div className="App">
10        <header className="App-header">
11          <h1>Welcome to AEM + React</h1>
12        </header>
13        <EditorContext.Provider value={ Utils.isInEditor() }>
14          { this.childComponents }
15          { this.childPages }
16        </EditorContext.Provider>
17      </div>
18    );
19  }
20}
21

```

App.js remains entry point

App.js extends Page class from @adobe/cq-react-editable-components

EditorContext only active in AEM authoring environment, based on value of Utils.isInEditor()

EditorContext facilitates communication model between AEM editor and React application

render() function this.childComponents and this.childPages include React

102

Integrate AEM SPA Editor JS SDK – *react-app/src/components/page/page.js*

```

1 /**
2 * Page.js
3 *
4 * - WKND specific implementation of Page
5 * - Maps to wknd-events/components/structure/page
6 */
7
8 import {Page, MapTo, withComponentMappingContext} from "@adobe/cq-react-editable-components";
9 require('./Page.css');
10 // This component is a variant of a React Page component mapped to the "structure/page" resource type
11 // No functionality is changed other than to add an app specific CSS class
12 class WkndPage extends Page {
13   get containerProps() {
14     let attrs = super.containerProps;
15     attrs.className = (attrs.className || '') + ' WkndPage ' + (this.props.cssClassNames || '');
16     return attrs;
17   }
18 }
19
20 MapTo('wknd-events/components/structure/page')(withComponentMappingContext(WkndPage));
21

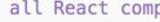
```

Map AEM resource type `wknd-events/components/structure/page` to a React component `WkndPage`

103

Integrate AEM SPA Editor JS SDK – *src/components/MappedComponents.js*

```
1  /*
2   Dedicated file to include all React components that map to an AEM component
3  */
4
5  require('./page/Page');
6  require('./text/Text');
7  require('./image/Image');
```

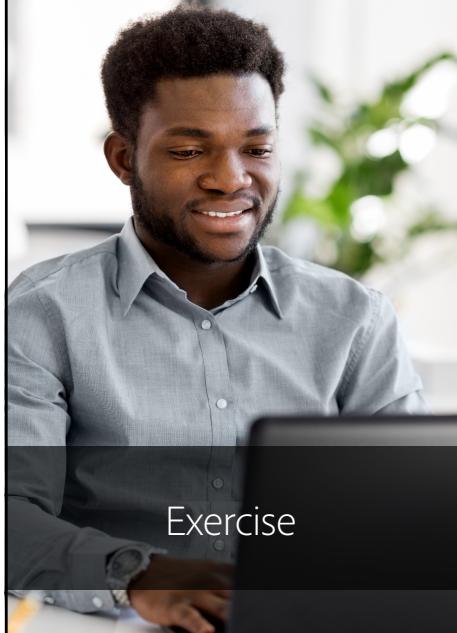


Create a dedicated file to identify all React components that map to an AEM component

104

SDK Integration Complete

```
//react-app/src/components/page/Page.js
...
attrs.className = (attrs.className || '') + ' WkndPage' + (this.props.cssClassNames || '');
...
```



Install AEM SPA Editor JS SDK

In this exercise, you will install AEM SPA Editor JS SDK as part of the react-app project using npm.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 27 - 30

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-1.html#install-spa-editor>
- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-1.html#integrate-spa-editor>

Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

106

ADOBEDIGITAL LEARNING SERVICES

Adobe

Text Component

```

1  /*
2   * Text.js
3   *
4   * Maps to wknd-events/components/content/text
5   */
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42

```

Set up a default editConfig for the Text component

Make AEM properties available as React properties

Map the AEM Text component to a React component that allows rich text authoring as part of the app

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

107

ADOBEDIGITAL LEARNING SERVICES

Adobe

home.html with Text Component in SPA Editor

HOME Edit Preview

Welcome to AEM + React

Hello World

Text

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

108 ADOBE DIGITAL LEARNING SERVICES

108

Text Component

In this exercise, you will map the AEM Text component to a React component.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 31 - 32

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-1.html#text-component>

Note: The step-by-step instructions are available in the Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

109 ADOBE DIGITAL LEARNING SERVICES

109

Image Component

Set up a default editConfig for the Image component

```

1 import React, {Component} from 'react';
2 import {MapTo} from '@adobe/cq-react-editable-components';
3
4 /**
5  * Default Edit configuration for the Image component that interact with the Core Image component and sub-types
6  *
7  * @type EditConfig
8 */
9 const ImageEditConfig = {
10   emptyLabel: 'Image',
11
12   isEmpty: function(props) {
13     return !props || !props.src || props.src.trim().length < 1;
14   }
15 }
16
17 /**
18  * Image React Component
19  */
20 class Image extends Component {
21   get content() {
22     return <img src={this.props.src} alt={this.props.alt}
23               title={this.props.displayPopupTitle && this.props.title}/>
24   }
25
26   render() {
27     return (<div className="Image">
28       {this.content}
29     </div>);
30   }
31 }
32
33
34
35
36 MapTo('wknd-events/components/content/image')(Image, ImageEditConfig);

```

Make AEM properties available as React properties

Map the AEM Image component to a React component that allows users to include an image in the app

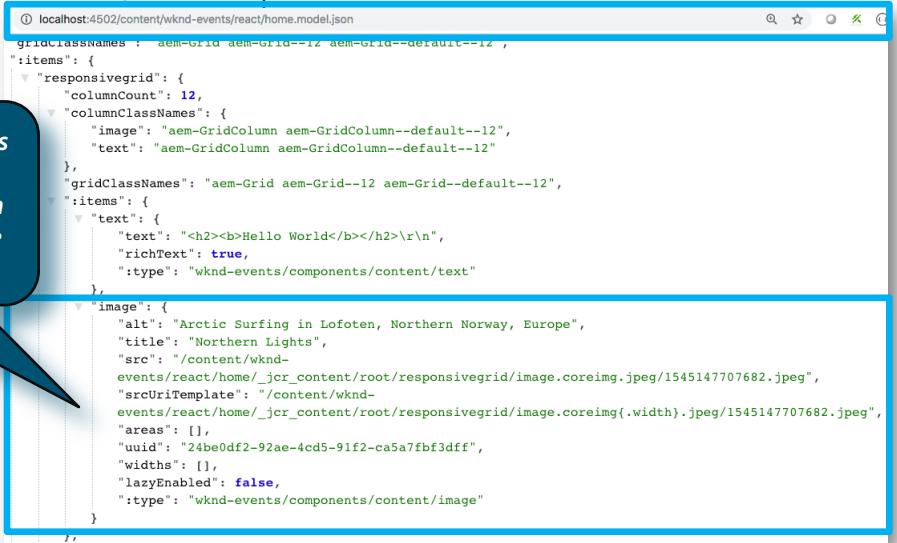
© 2019 Adobe. All Rights Reserved. Adobe Confidential.

110 ADOBE DIGITAL LEARNING SERVICES 

110

Image Component Model JSON Export

MapTo functionality exposes the JSON values of alt, title, src that can then be used by the React Image component.



```

{
  "gridClassNames": "aem-Grid aem-Grid--12 aem-Grid--default--12",
  "items": [
    {
      "responsivegrid": {
        "columnCount": 12,
        "columnClassNames": {
          "image": "aem-GridColumn aem-GridColumn--default--12",
          "text": "aem-GridColumn aem-GridColumn--default--12"
        },
        "gridClassNames": "aem-Grid aem-Grid--12 aem-Grid--default--12",
        "items": [
          {
            "text": {
              "text": "<h2><b>Hello World</b></h2>\r\n",
              "richText": true,
              "type": "wknd-events/components/content/text"
            }
          },
          {
            "image": {
              "alt": "Arctic Surfing in Lofoten, Northern Norway, Europe",
              "title": "Northern Lights",
              "src": "/content/wknd-events/react/home/_jcr_content/root/responsivegrid/image.coreimg.jpeg/1545147707682.jpeg",
              "srcUriTemplate": "/content/wknd-events/react/home/_jcr_content/root/responsivegrid/image.coreimg{.width}.jpeg/1545147707682.jpeg",
              "areas": [],
              "uuid": "24be0df2-92ae-4cd5-91f2-ca5a7fbf3dff",
              "widths": [],
              "lazyEnabled": false,
              "type": "wknd-events/components/content/image"
            }
          }
        ]
      }
    }
  ]
}

```

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

111 ADOBE DIGITAL LEARNING SERVICES 

111

home.html with Image Component in SPA Editor

The screenshot shows the AEM SPA Editor interface. At the top, there's a navigation bar with icons for file operations, a 'HOME' button, an 'Edit' dropdown, a 'Preview' button, and a user profile icon. Below the navigation is the main content area with the heading 'Welcome to AEM + React'. A toolbar with various icons (e.g., edit, delete, insert) is positioned above the content. An image component is currently selected, indicated by a blue border and the word 'Image' in the status bar. The main content area contains a 'Text' component with the text 'Hello World'. A placeholder box below it says 'Drag components here'. On the left side, a component palette lists several components: 'Image' (WKND Events - Content), 'Layout Container' (General), 'List' (WKND Events - Content), and 'Text' (WKND Events - Content). The bottom of the screen shows copyright information, page numbers (112), and Adobe branding.

112

home.html in SPA Editor

This screenshot shows the AEM SPA Editor interface with the 'Components' palette open on the left. The palette lists components under the 'Components' tab, with a 'Filter' field and a dropdown menu set to 'All'. The listed components are: 'Image' (WKND Events - Content), 'Layout Container' (General), 'List' (WKND Events - Content), and 'Text' (WKND Events - Content). The 'Text' component is currently selected, as indicated by its highlighted state. The main content area displays the 'Welcome to AEM + React' page with a 'Text' component containing the text 'Hello World'. A toolbar is visible above the content, and a 'Drag components here' placeholder is below it. The bottom of the screen shows copyright information, page numbers (113), and Adobe branding.

113

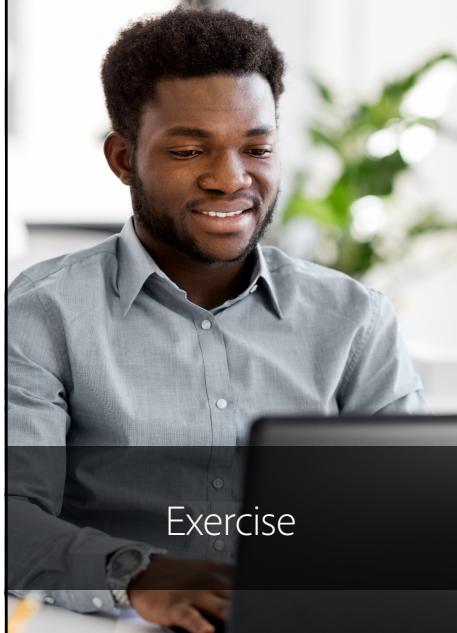


Image Component

In this exercise, you will map the AEM Image component to a React component.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 33 - 34

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-1.html#image-component>

Note: The step-by-step instructions are available in your Student Guide.

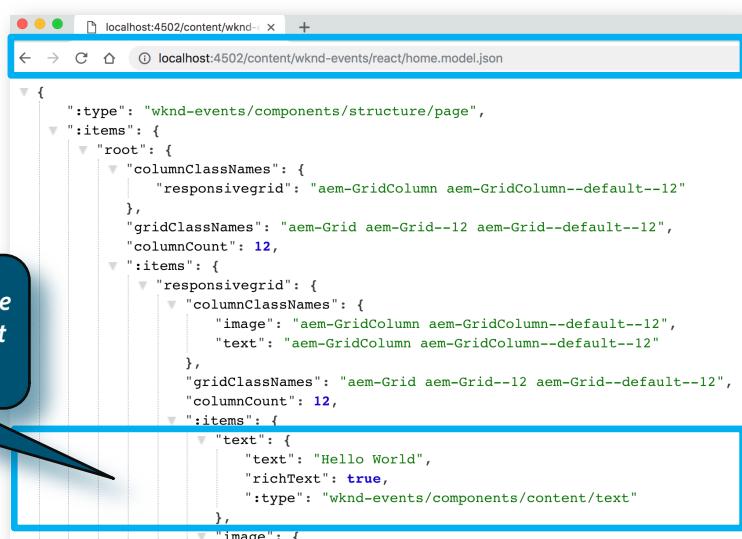
© 2019 Adobe. All Rights Reserved. Adobe Confidential.

114 ADOBE DIGITAL LEARNING SERVICES 

114

Text Component Model JSON Export

MapTo functionality exposes the JSON values of Text component properties.



```

{
  "type": "wknd-events/components/structure/page",
  "items": {
    "root": {
      "columnClassNames": {
        "responsivegrid": "aem-GridColumn aem-GridColumn--default--12"
      },
      "gridClassNames": "aem-Grid aem-Grid--12 aem-Grid--default--12",
      "columnCount": 12,
      "items": {
        "responsivegrid": {
          "columnClassNames": {
            "image": "aem-GridColumn aem-GridColumn--default--12",
            "text": "aem-GridColumn aem-GridColumn--default--12"
          },
          "gridClassNames": "aem-Grid aem-Grid--12 aem-Grid--default--12",
          "columnCount": 12,
          "items": {
            "text": {
              "text": "Hello World",
              "richText": true,
              "type": "wknd-events/components/content/text"
            },
            "image": {
              "type": "wknd-events/components/content/image"
            }
          }
        }
      }
    }
  }
}

```

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

115 ADOBE DIGITAL LEARNING SERVICES 

115

HierarchyPage Sling Model

AEM SPA JS SDK is designed to parse a JSON schema into a JavaScript Model.

HierarchyPage Sling Model exposes AEM content as JSON using the expected schema.

HierarchyPageImpl exported JSON can expose the content of multiple AEM pages in a single request.

- Allows the SPA to be initialized with most of the content of the app.
- Can remove the need for subsequent requests as a user navigates the app.

*core/src/main/java/com/adobe/aem/guides/wkndevents/core/models/
HierarchyPage.java*

```

1 package com.adobe.aem.guides.wkndevents.core.models;
2
3 import com.adobe.cq.export.json.ContainerExporter;
4 import com.adobe.cq.export.json.hierarchy.HierarchyNodeExporter;
5 import com.fasterxml.jackson.annotation.JsonIgnore;
6 import com.fasterxml.jackson.annotation.JsonProperty;
7
8 public interface HierarchyPage extends HierarchyNodeExporter, ContainerExporter {
9
10    /**
11     * Title of the page. The page title can be the result of multiple fallbacks
12     *
13     * @return
14     */
15    @JsonProperty("title")
16    public String getTitle();
17
18    /**
19     * URL to the root model of the App
20     *
21     * @return
22     */
23    @JsonIgnore
24    public String getRootUrl();
25
26    @JsonIgnore
27    public HierarchyPage getRootModel();
28
29 }

```

HierarchyPage extends 2 interfaces:

- **ContainerExporter** - Defines the JSON of a container component like a page, responsive grid, or parsys.
- **HierarchyNodeExporter** - Defines the JSON of a hierarchical node, like a root page and its child pages.

`core/src/main/java/com/adobe/aem/guides/wkndevents/core/models/impl/HierarchyPageImpl.java`

Registered as Sling Model Exporter for the `wknd-events/components/structure/page` resource type.

- For custom projects, update the `RESOURCE_TYPE` to point to your project's base page component.

Methods `getRoodModel()` and `getRootPage()` used to find and return the "root" of the application.

```

1  @Model(adaptables = SlingHttpServletRequest.class, adapters = {HierarchyPage.class, ContainerExporter.class}, resourceType = HierarchyPageImpl.RESOURCE_TYPE)
2  @Exporter(name = ExporterConstants.SLING_MODEL_EXPORTER_NAME, extensions = ExporterConstants.SLING_MODEL_EXTENSION)
3  public class HierarchyPageImpl implements HierarchyPage {
4
5      /**
6       * Resource type of associated with the current implementation
7      */
8      protected static final String RESOURCE_TYPE = "wknd-events/components/structure/page";
9
10     ...
11 }
```

Registered as a Sling Model Exporter for the
`wknd-events/components/structure/page`
resource type

118

Collection of Content

Three properties stored on an App template's policy are used to drive the collection of content:

- PR_IS_ROOT** = "isRoot" - Helps identify the `rootPage` of the application.
The `rootPage` is used as the starter point to collect all the child pages of the app.
- STRUCTURE_DEPTH_PN** = "structureDepth" - identifies how deep in the hierarchy to collect child pages.
- STRUCTURE_PATTERNS_PN** = "structurePatterns" - regular expression that can be used to ignore or exclude certain pages from automatically being collected.

SPA App Root

The screenshot shows the configuration of a React application template in CRXDE Lite. It includes:

- A blue callout pointing to the JCR content browser showing the path `/content/wknd-events/react/jcr:content`.
- A blue callout pointing to the policy configuration for the template, labeled "Policy for wknd-events-app-template template".
- A blue callout pointing to the template definition in the properties tab, labeled "Collect 2 levels of child pages".
- A blue callout pointing to the template definition in the properties tab, labeled "Ignore blog pages".

120

HierarchyPage Sling Model JSON Output

The screenshot shows the Sling Model JSON output for a hierarchy page and the corresponding JCR node structure.

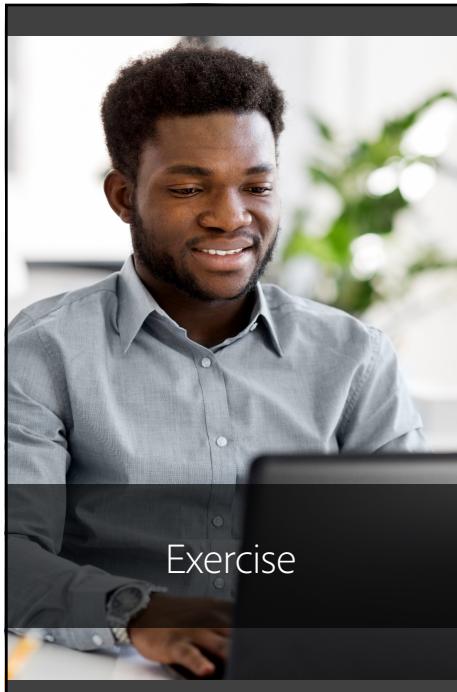
The JSON output (left) shows:

```
{
  "items": {},
  "itemsOrder": [],
  "hierarchyType": "page",
  "path": "/content/wknd-events/react",
  "children": {
    "/content/wknd-events/react/home": {
      "items": {}, // 1 item
      "itemsOrder": [
        "root"
      ],
      "hierarchyType": "page",
      "path": "/content/wknd-events/react/home",
      "title": "Home",
      "type": "wknd-events/components/structure/page"
    },
    "/content/wknd-events/react/home/homeChild1": {
      "items": {}, // 1 item
      "itemsOrder": [
        "root"
      ],
      "hierarchyType": "page",
      "path": "/content/wknd-events/react/home/homeChild1",
      "title": "Home",
      "type": "wknd-events/components/structure/page"
    }
  },
  "title": "React App",
  "type": "wknd-events/components/structure/app"
}
```

The JCR content browser (right) shows the corresponding node structure:

- App root (the root node)
- child page level 1 (the `home` node under `/content/wknd-events/react`)
- child page level 2 (the `homeChild1` node under `/content/wknd-events/react/home`)
- child page level 3 (the `homeChild2` node under `/content/wknd-events/react/home/homeChild1`)

121



Exercise

HierarchyPage Sling Model (Optional)

In this exercise, you will use the Sling model to expose content within AEM as JSON that matches the expected schema.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Resources:

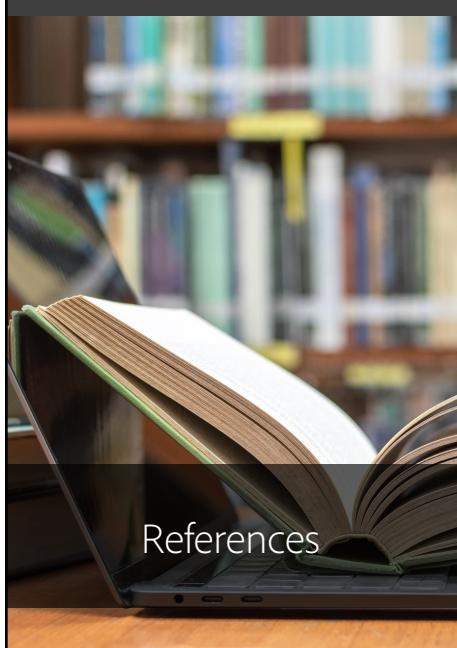
- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-1.html#hierarchypage-sling-model>

Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

122

ADOBEDIGITAL LEARNING SERVICES 



References

Hands-on Activity

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-1.html>

Core Components

- <https://helpx.adobe.com/experience-manager/core-components/using/introduction.html>
- <https://helpx.adobe.com/experience-manager/core-components/using/developing.html>

Editable Templates

- <https://helpx.adobe.com/experience-manager/6.4/sites/developing/using/page-templates-editable.html>

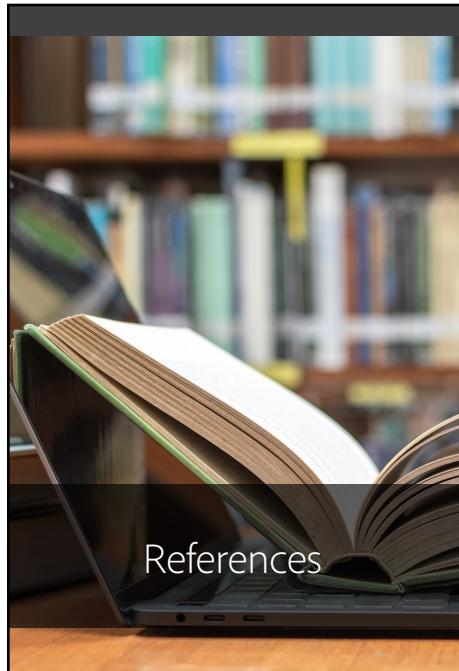
Sling Models

- <https://sling.apache.org/documentation/bundles/models.html>
- https://helpx.adobe.com/experience-manager/using/sling_models.html

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

123

ADOBEDIGITAL LEARNING SERVICES 



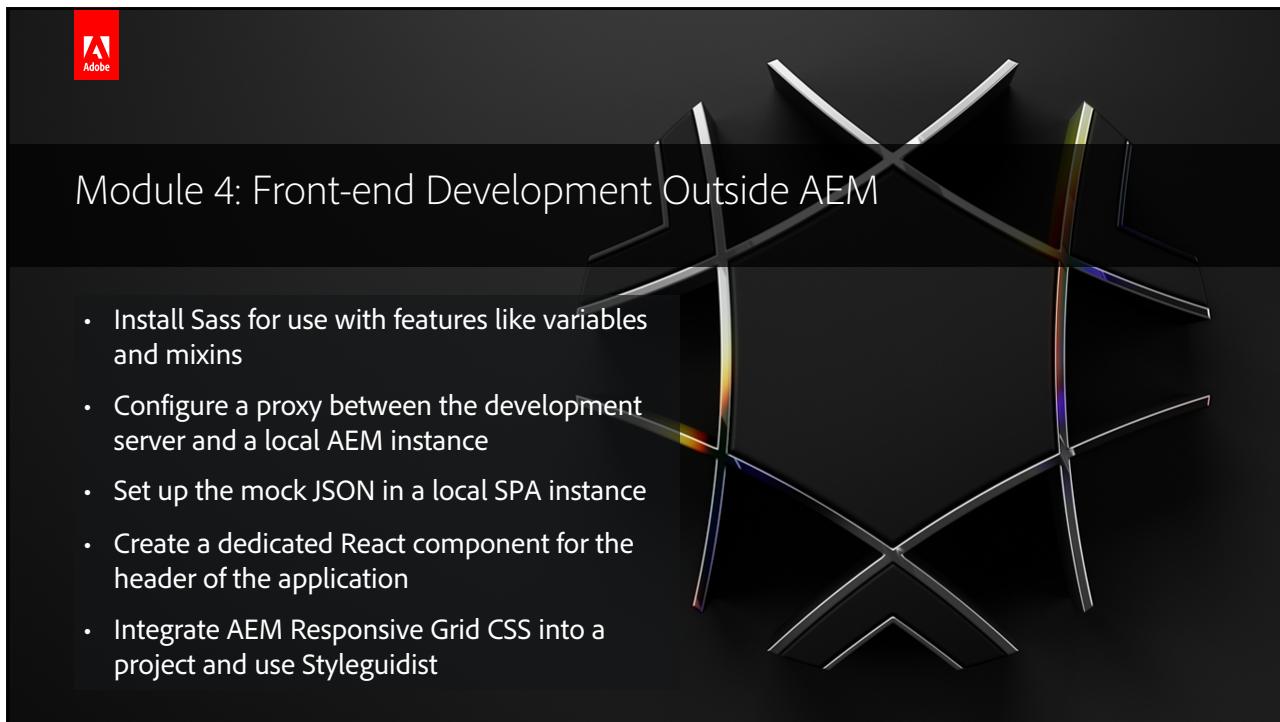
SPA Blueprint

- https://helpx.adobe.com/experience-manager/6-4/sites/developing/using/spa-blueprint.html#main-pars_header_249430796
- <https://helpx.adobe.com/experience-manager/6-4/sites/developing/using/spa-dynamic-model-to-component-mapping.html>

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

124

ADOBEDIGITAL LEARNING SERVICES 



Module 4: Front-end Development Outside AEM

- Install Sass for use with features like variables and mixins
- Configure a proxy between the development server and a local AEM instance
- Set up the mock JSON in a local SPA instance
- Create a dedicated React component for the header of the application
- Integrate AEM Responsive Grid CSS into a project and use Styleguidist

React Best Practice, CSS, Sass and SUIT

A React best practice is to keep each component modular and self-contained.

- **General recommendation:** avoid reusing the same CSS class name across components.
 - *Result:* use of preprocessors not as powerful.

This project uses Syntactically Awesome Style Sheets (Sass) for a few features, such as:

- Variables
- Mixins

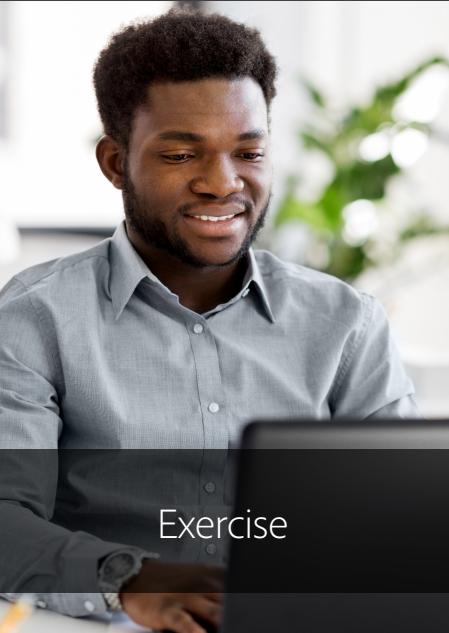
This project will also loosely follow SUIT CSS naming conventions.

- SUIT is a variation of Block Element Modifier notation (BEM), used to create consistent CSS rules.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

 Adobe

126



Install Sass Support

In this exercise, you will install Sass for use with features like variables and mixins.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 40 - 42

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-2.html#sass>

Note: Step-by-step instructions are available in the Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

127

ADobe DIGITAL LEARNING SERVICES  Adobe

127

Rapid Development Approaches

Proxy JSON

- Configure a Proxy between static React development server and AEM instance.

Mock JSON

- Use a static or mock JSON file to simulate AEM responses.

Proxy JSON Approach – *react-app/package.json*

```

1 // package.json
2 {
3   "name": "react-app",
4   "version": "0.1.0",
5   "private": true,
6   "dependencies": {
7     "@adobe/cq-react-editable-components": "^1.0.4",
8     "@adobe/cq-spa-component-mapping": "^1.0.3",
9     "@adobe/cq-spa-page-model-manager": "^1.0.5",
10    "node-sass": "^4.11.0",
11    "react": "^16.5.2",
12    "react-dom": "^16.5.2",
13    "react-fast-compare": "A2.0.4",
14    "react-scripts": "2.0.4"
15  },
16  "scripts": {
17    "start": "react-scripts start",
18    "build": "react-scripts build & clientlib --verbose",
19    "test": "react-scripts test",
20    "eject": "react-scripts eject"
21  },
22  "proxy": "http://localhost:4502",
23  "eslintConfig": {
24    "extends": "react-app"
25  },
26  "browserslist": [
27    ">=0.20",
28    "not dead",
29    "not ie <= 11",
30    "not op_mini all"
31  ],
32  "devDependencies": {
33    "aem-clientlib-generator": "A1.4.1",
34    "ajv": "A6.6.2",
35    "clone": "A2.1.2"
36  }
37 }

```

Configure proxy endpoint

Configure a proxy between the React development server and a local AEM instance.

Requests to AEM content (for example the JSON Model and images) will be served as if the request(s) had originated on the static development server.

Configure Environment Variables

```

1 // src/index.js
2
3 import React from 'react';
4 import ReactDOM from 'react-dom';
5 import { ModelManager, Constants } from '@adobe/cq-spa-page-model-manager';
6 import App from './App';
7 //include Mapped Components
8 import "./components/MappedComponents";
9
10 function render(model) {
11   ReactDOM.render(
12     <App cqChildren={ model[Constants.CHILDREN_PROP] } cqItems={ model[Constants.ITEMS_PROP] } cqItemsOrder={ model[Constants.ITEMS_ORDER_PROP] } cqPath={ ModelManager.rootPath } locationPathname={ window.location.pathname } />, document.getElementById('root'));
13 }
14
15 ModelManager.initialize({ path: process.env.REACT_APP_PAGE_MODEL_PATH }).then(render);
16

```

initialize JSON Model using the environment variable
REACT_APP_PAGE_MODEL_PATH

For example, a request to `/content/wknd-events/react.model.json` will not exist on the static server, so it will be proxied in from AEM when running the app in the development mode.

Environment variables when running in development mode.

```

1 # react-app/.env.development
2
3 # Configure Proxy end point
4 REACT_APP_PAGE_MODEL_PATH=/content/wknd-events/react.model.json

```

130

AEM Successfully Proxied

React development server: 3000

Welcome to AEM + React

Hello World

AEM local host: 4502

Welcome to AEM + React

Hello World

131

Make Style Changes Outside of AEM - /src/index.scss

The image shows three separate browser windows side-by-side, each displaying the same 'react/home.html' page. The first window has 'Welcome to AEM + React' in blue text and 'Hello World' in black text. The second window has 'Welcome to AEM + React' in yellow text and 'Hello World' in black text. The third window has 'Welcome to AEM + React' in gray text and 'Hello World' in black text. Each browser window also displays the corresponding SCSS code for the styles applied.

```

23  @h1, @h2, @h3, @h4 {
24    font-family: $font-sans;
25  }
26  @h1 {
27    color: blue;
28    font-size: $h1-font-size;
29  }
30  @h2 {
31    font-size: $h2-font-size;
32  }
33  @h3 {
34    font-size: $h3-font-size;
35  }
36  @h4 {
37    font-size: $h4-font-size;
38  }

```

```

23  @h1, @h2, @h3, @h4 {
24    font-family: $font-sans;
25  }
26  @h1 {
27    color: yellow;
28    font-size: $h1-font-size;
29  }
30  @h2 {
31    font-size: $h2-font-size;
32  }
33  @h3 {
34    font-size: $h3-font-size;
35  }
36  @h4 {
37    font-size: $h4-font-size;
38  }

```

```

23  @h1, @h2, @h3, @h4 {
24    font-family: $font-sans;
25  }
26  @h1 {
27    color: gray;
28    font-size: $h1-font-size;
29  }
30  @h2 {
31    font-size: $h2-font-size;
32  }
33  @h3 {
34    font-size: $h3-font-size;
35  }
36  @h4 {
37    font-size: $h4-font-size;
38  }

```

Update client files in *react-app/src*.

132

Proxy JSON Approach

In this exercise, you will configure a proxy between the development server, which is bundled with the React project and a local AEM instance.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 43 - 47

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-2.html#proxy-json>

Note: Step-by-step instructions are available in the Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

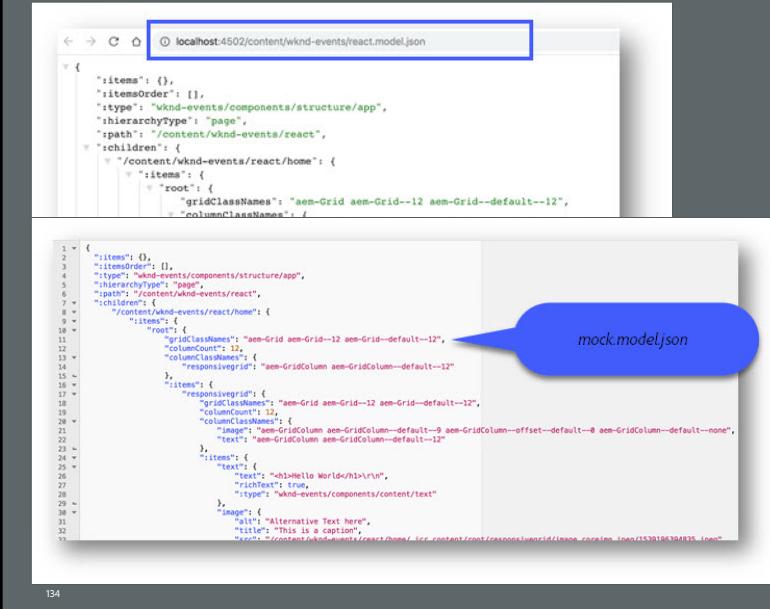
133

ADOBEDIGITAL LEARNING SERVICES

Adobe

133

Mock JSON Approach – *react-app/public/mock.model.json*



Use a static or mock JSON file.

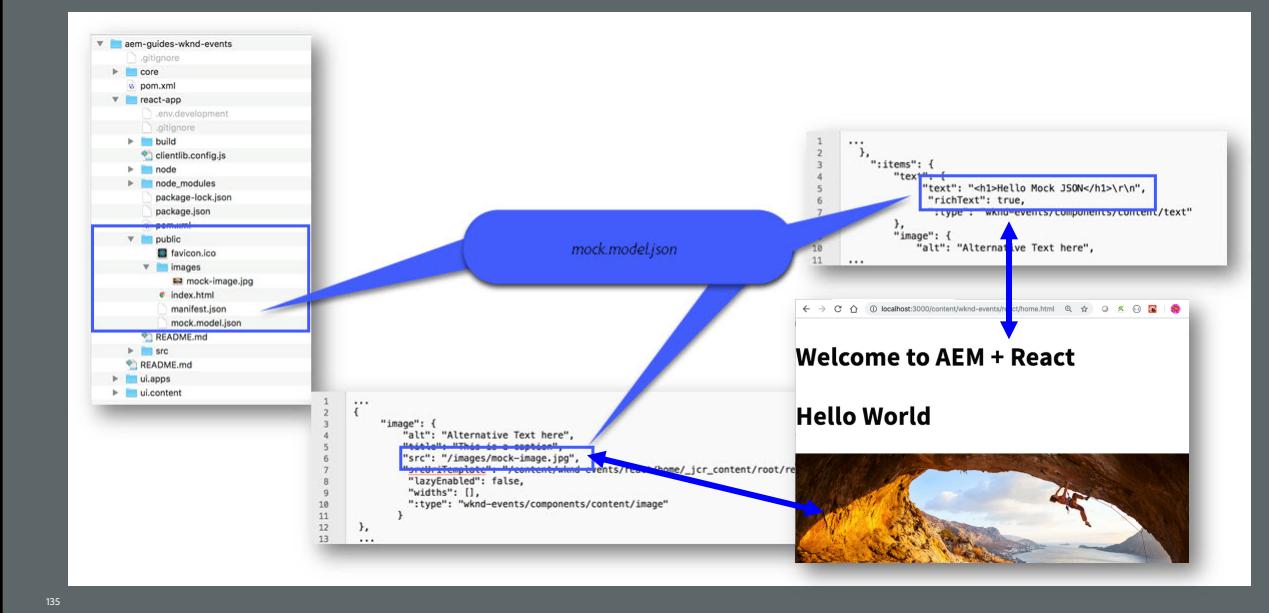
Removes dependency on a local AEM instance.

Allows the Front-End developer to update the JSON.

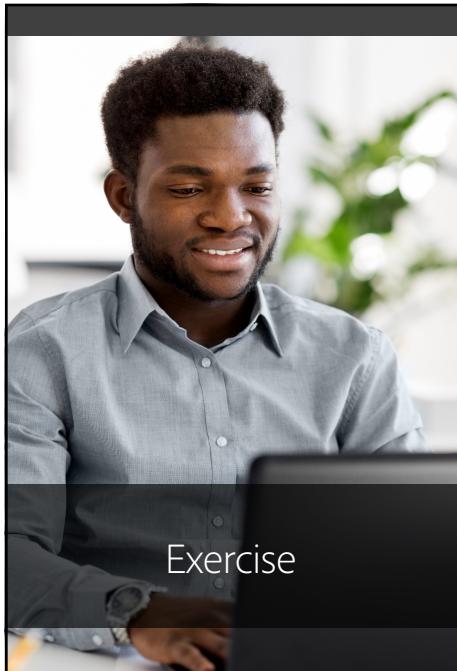
- Test functionality
 - Mock new JSON responses without dependency on a Back-End developer

134

Mock AEM Responses for Testing



135



Mock JSON Approach

In this exercise, you will setup the mock JSON in a local AEM instance.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 48 - 50

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-2.html#mock-json>

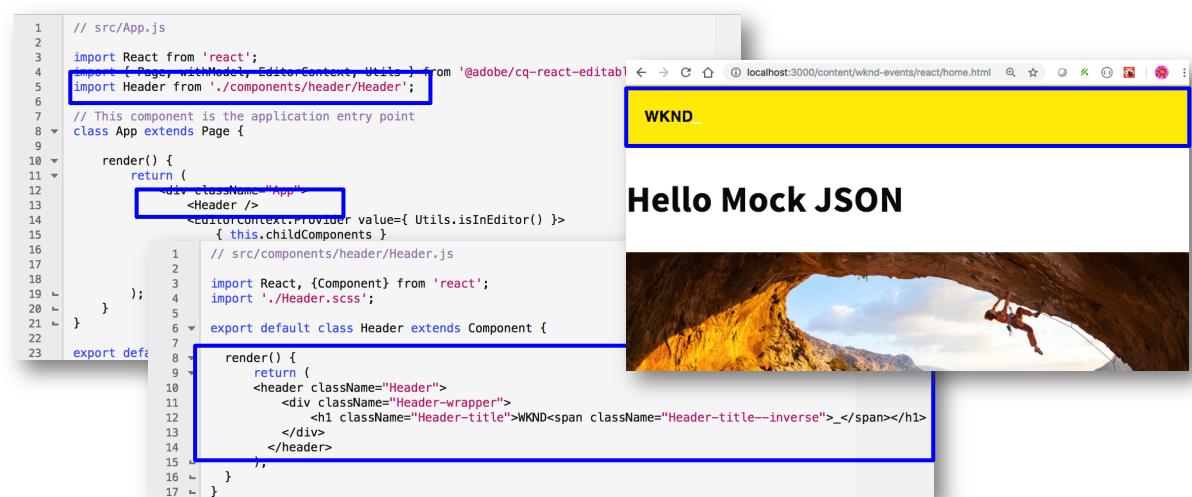
Note: Step-by-step instructions are available in the Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

136

ADOBEDIGITAL LEARNING SERVICES 

Dedicated React Components – *react-app/src/components*



The screenshot shows the code for *src/App.js* and *src/components/Header.js*, and a browser preview of the resulting application.

src/App.js code:

```

1 // src/App.js
2 import React from 'react';
3 import { Page, withModel, EditorContext, Utils } from '@adobe/cq-react-editable'
4 import Header from './components/header/Header';
5
6 // This component is the application entry point
7 class App extends Page {
8   render() {
9     return (
10       <div className="App">
11         <Header />
12         <EditorContext.Provider value={ Utils.isInEditor() } >
13           { this.childComponents }
14         </EditorContext.Provider>
15       </div>
16     );
17   }
18 }
19
20 export default App;
21
22 
```

src/components/Header.js code:

```

1 // src/components/Header.js
2 import React, {Component} from 'react';
3 import './Header.scss';
4
5 export default class Header extends Component {
6   render() {
7     return (
8       <header className="Header">
9         <div className="Header-wrapper">
10           <h1 className="Header-title">WKND<span className="Header-title--inverse">_</span></h1>
11         </div>
12       </header>
13     );
14   }
15 }
16
17 
```

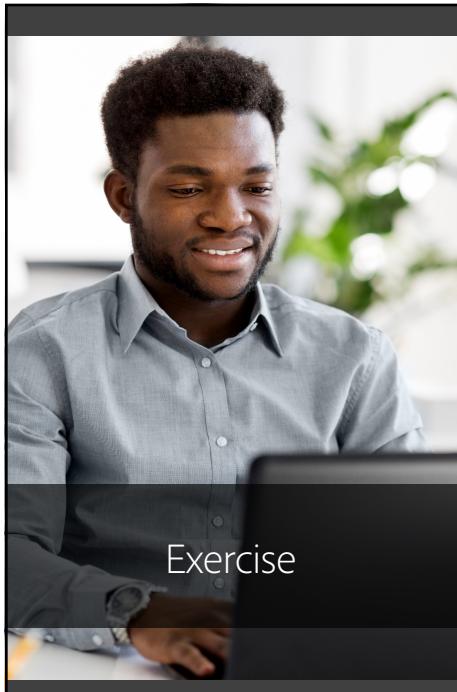
Browser preview:

The browser window shows the application running at localhost:3000/content/wknd-events/react/home.html. The header displays "WKND" and "Hello Mock JSON". Below the header is a large image of a person climbing a rock face.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

137

ADOBEDIGITAL LEARNING SERVICES 



Header Component

In this exercise, you will create a dedicated React component for the header of the application.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 51 – 53

Note; small error: screen shot for step 5 is of wrong file. Ignore the image, use the exercise file.

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-2.html#header-component>

Note: Step-by-step instructions are available in the Student Guide.

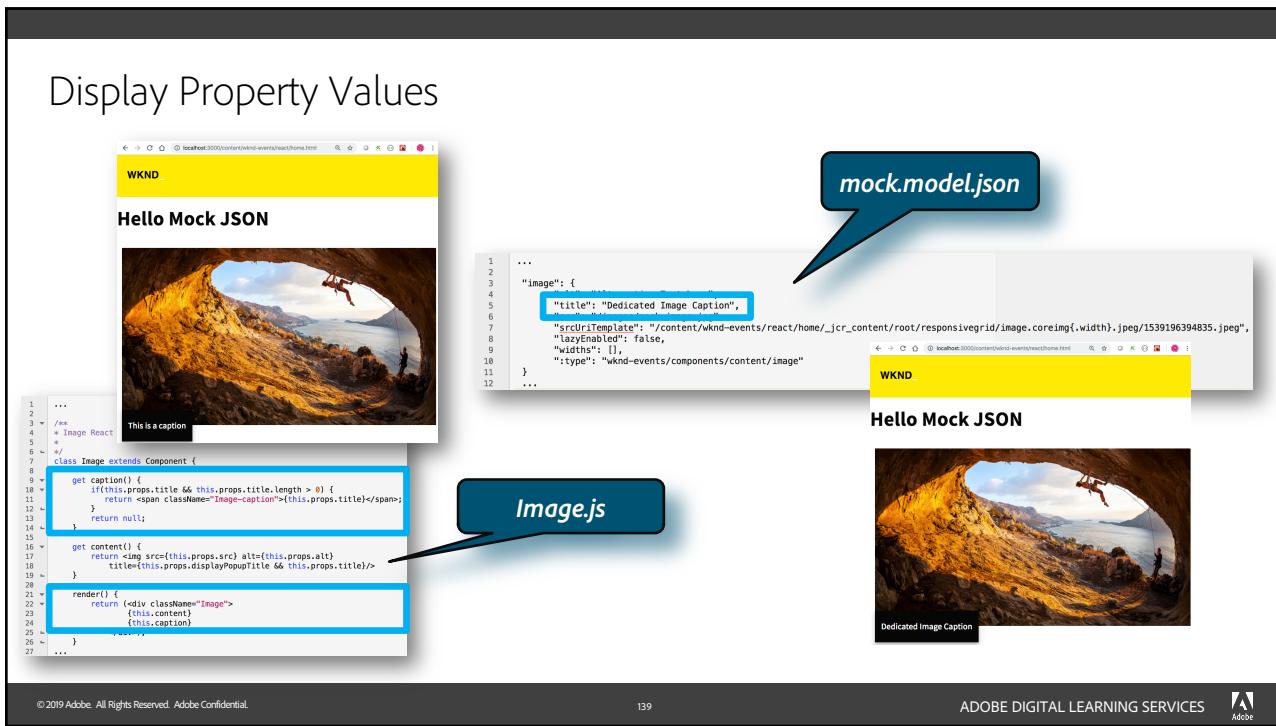
© 2019 Adobe. All Rights Reserved. Adobe Confidential.

138

ADOBEDIGITAL LEARNING SERVICES

Adobe

Display Property Values



mock.model.json

```
{
  "image": {
    "title": "Dedicated Image Caption",
    "srcUriTemplate": "/content/wknd-events/react/home/_jcr_content/root/responsivegrid/image.coreimg{.width}.jpeg/1539196394835.jpeg",
    "lazyEnabled": false,
    "widths": [],
    "type": "wknd-events/components/content/image"
  }
}
```

Image.js

```

1 ...
2 ...
3 ...
4 ...
5 ...
6 ...
7 ...
8 ...
9 ...
10 ...
11 ...
12 ...
13 ...
14 ...
15 ...
16 ...
17 ...
18 ...
19 ...
20 ...
21 ...
22 ...
23 ...
24 ...
25 ...
26 ...
27 ...

```

This is a caption

WKND

Hello Mock JSON

Dedicated Image Caption

WKND

Hello Mock JSON

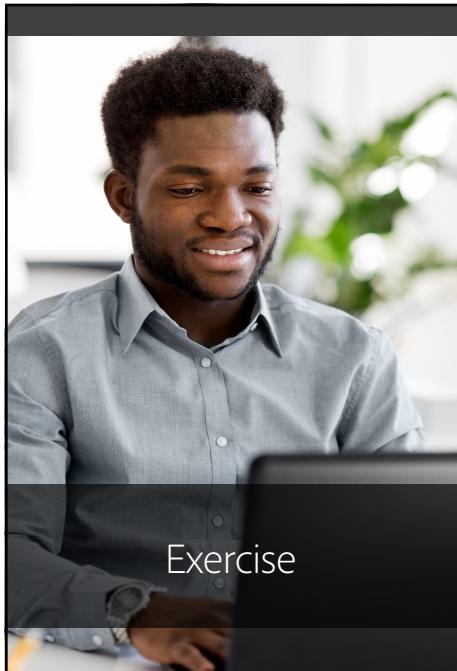
Dedicated Image Caption

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

139

ADOBEDIGITAL LEARNING SERVICES

Adobe



Update Image Component

In this exercise, you will add the option to display a caption below the image component.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 54 - 55

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-2.html#image-component>

Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

140

ADOBEDIGITAL LEARNING SERVICES 

Use Client Libraries to Promote Consistency



```
Text.js
```

```
1  /*
2   * Text.js
3   *
4   * Maps to wknd-events/components/content/text
5   */
6
7   import React, {Component} from 'react';
8   import {MapTo} from '@adobe/cq-react-editable-components';
9   require('./Text.scss');
10  ...
```

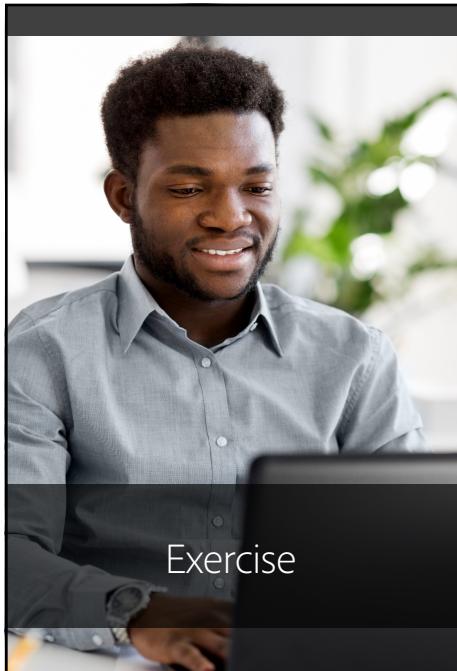
```
Text.scss
```

```
1  @import '../../../../../styles/shared';
2
3  .Text {
4    @include component-padding();
5  }
```

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

141

ADOBEDIGITAL LEARNING SERVICES 



Update Text Component

In this exercise, you will make some small modifications to the text component to promote consistency across all mapped components.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Page 56

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-2.html#text-component>

Note: Step-by-step instructions are available in the Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

142

ADOBEDIGITAL LEARNING SERVICES 

Responsive Grid

AEM ships with a special container component called the Responsive Grid.

AEM Page Editor's "Layout Mode" allows editing the grid and flow of components

- Allows authors to resize, position, and hide components to optimize content for different viewport sizes.

Native to AEM SPA Editor

- Only need to integrate Responsive Grid CSS into components.

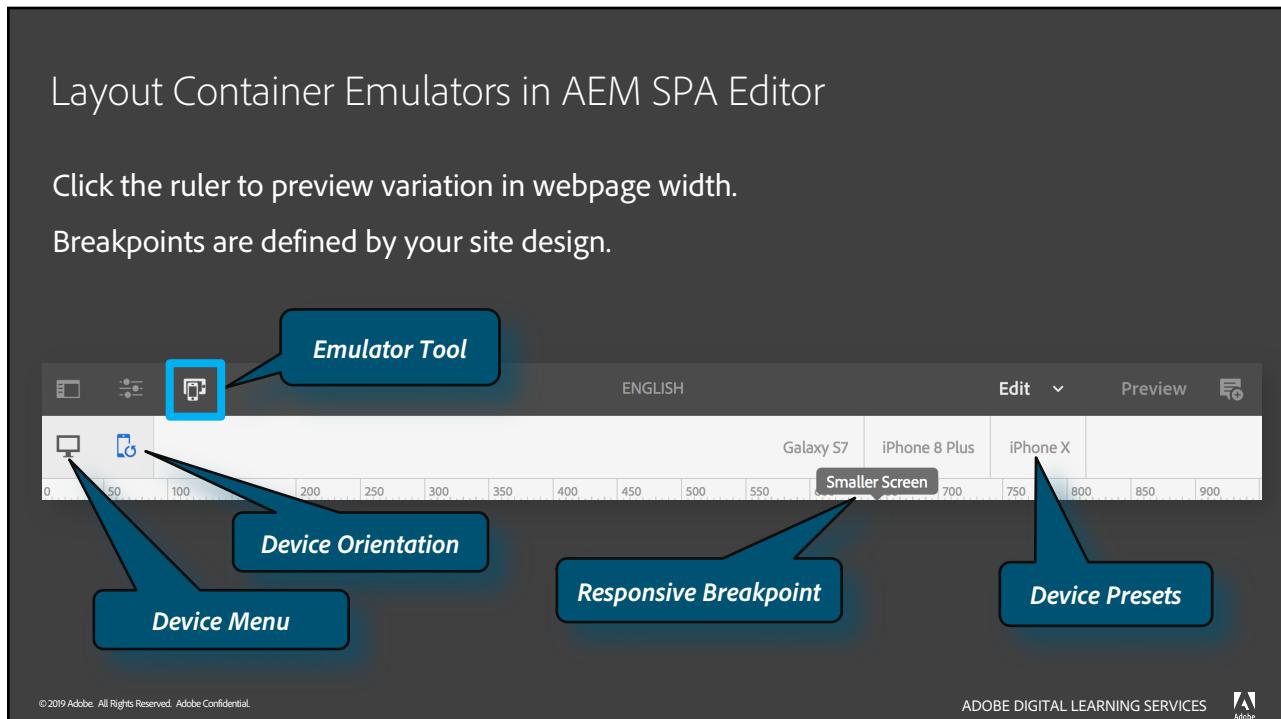
Responsive Grid client library

- category property = wknd-events.grid
- grid.less file included in the library, contains necessary CSS for Layout Mode.

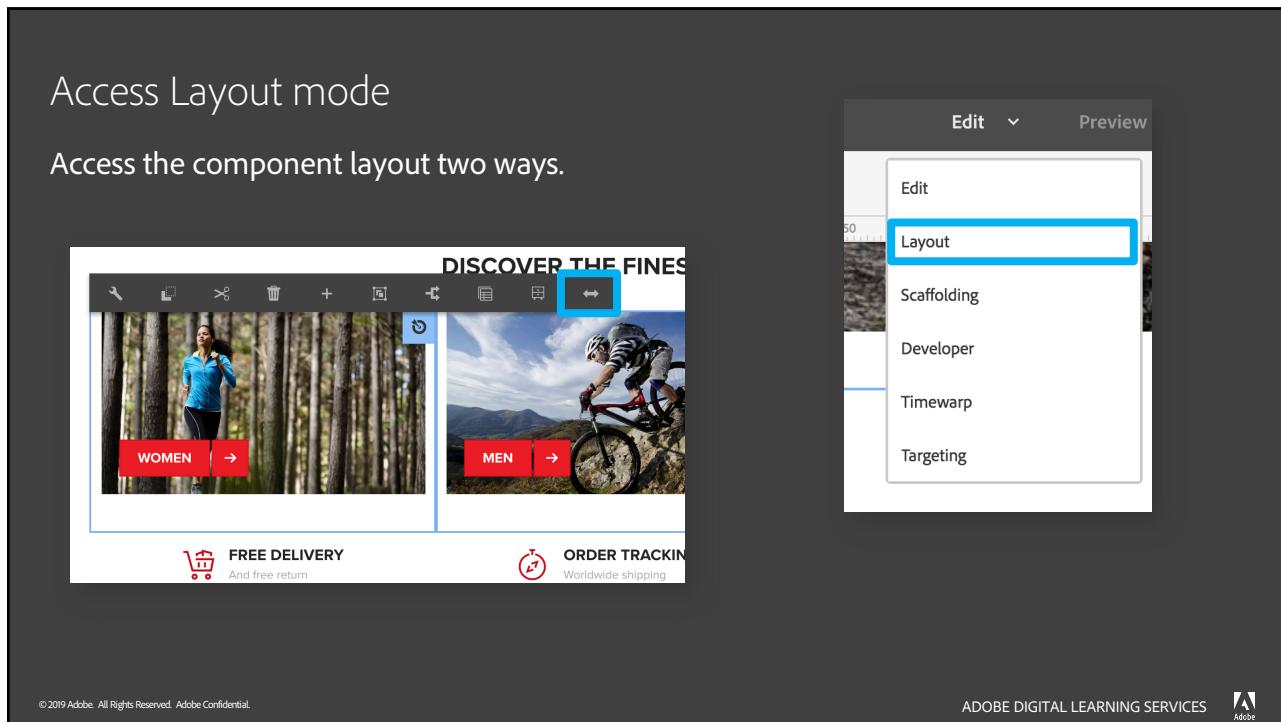
© 2019 Adobe. All Rights Reserved. Adobe Confidential.

143

ADOBEDIGITAL LEARNING SERVICES 



144



145

Modify component layout

Use blue drag circles to adjust component width on page.

Separate layouts of the components can be created per breakpoint.

DISCOVER THE FINEST GEAR

WOMEN → MEN → EQUIPMENT → ONLINE SUPPORT

FREE DELIVERY And free return ORDER TRACKING Worldwide shipping

© 2019 Adobe. All Rights Reserved. Adobe Confidential. ADOBE DIGITAL LEARNING SERVICES Adobe

146

Modify component layout

Float to new line.

- Force a component to a new line.

SHOP NOW → SHARE NOW →

© 2019 Adobe. All Rights Reserved. Adobe Confidential. ADOBE DIGITAL LEARNING SERVICES Adobe

147

Modify component layout

Hide component

- Hide a component for a specific screen.

Reveal and restore a hidden component with the layout container.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

ADOBEDIGITAL LEARNING SERVICES

148

Integrating Responsive Grid into SPA- /react-app/clientlib.config.js

```

1 module.exports = {
2   // default working directory (can be changed per 'cwd' in every asset option)
3   context: __dirname,
4   // path to the clientlib root folder (output)
5   clientlibRoot: "././ui.apps/src/main/content/jcr_root/apps/wknd-events/clientlibs",
6
7   libs: [
8     {
9       name: "react-app",
10      allowProxy: true,
11      categories: ["Wknd-Events.React"],
12      serializationFormat: "xml",
13      dependencies: ["wknd-events,grid"],
14      js: [
15        "build/static/**/*.js"
16      ],
17      css: [
18        "build/static/**/*.css"
19      ]
20    }
21  }
22}
23
24};

```

clientlib.config.js

WKND

Hello World

WKND

Hello World

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

149

ADOBEDIGITAL LEARNING SERVICES

Use Layout Container/Responsive Grid to Create Columns

Layout Container 1

Layout Container 2

Layout Container 3

Article Heading 1

SideBar Heading 2

WKND

Drag components here

Drag components here

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

150

ADOBEDIGITAL LEARNING SERVICES

Adobe

150

AEM Instance and React Development Server Render the Same Content

localhost:3000/content/wknd-events/react/home.html

localhost:4502/content/wknd-events/react/home.html?wcmmode=disabled

WKND

WKND

Article Heading 1

SideBar Heading 2

Article Heading 1

SideBar Heading 2

WKND

Majestic Rainbow

Majestic Rainbow

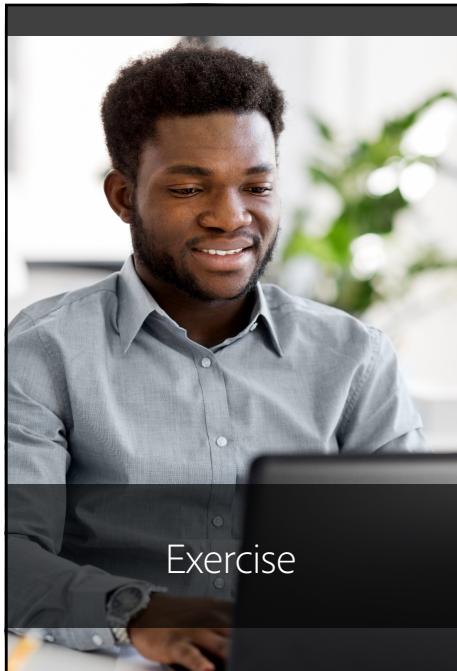
© 2019 Adobe. All Rights Reserved. Adobe Confidential.

151

ADOBEDIGITAL LEARNING SERVICES

Adobe

151



Integrate Responsive Grid

In this exercise, you will integrate AEM Responsive Grid CSS into your project.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 57 - 59

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-2.html#responsive-grid>

Note: Step-by-step instructions are available in the Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

152

ADOBEDIGITAL LEARNING SERVICES 

React Styleguidist



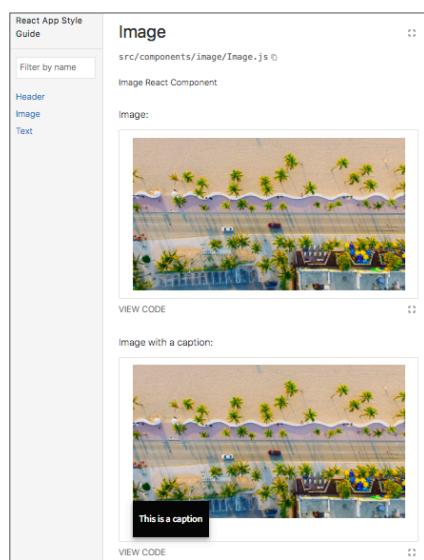
Style Guide generator.

Isolated React component development environment.

Living style guide:

- Visual overview of components
- Categorization of components into logical groups
- Playground to try all possible variations of properties and content

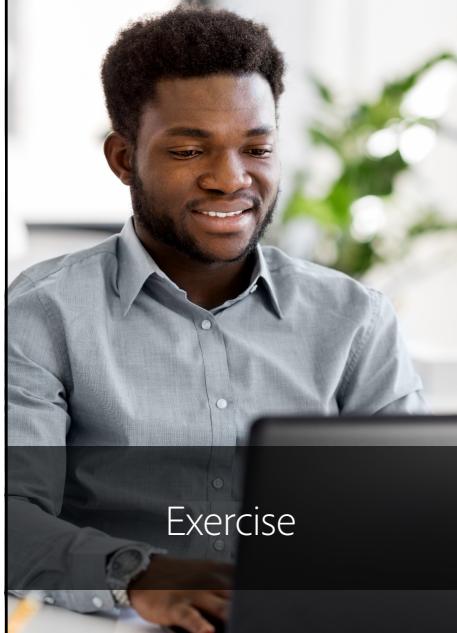
Generates component documentation.



© 2019 Adobe. All Rights Reserved. Adobe Confidential.

153

ADOBEDIGITAL LEARNING SERVICES 



Integrate Styleguidist

In this exercise, you will use Styleguidist as it combines a style guide and documentation into a single tool.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 60 - 62

Resources:

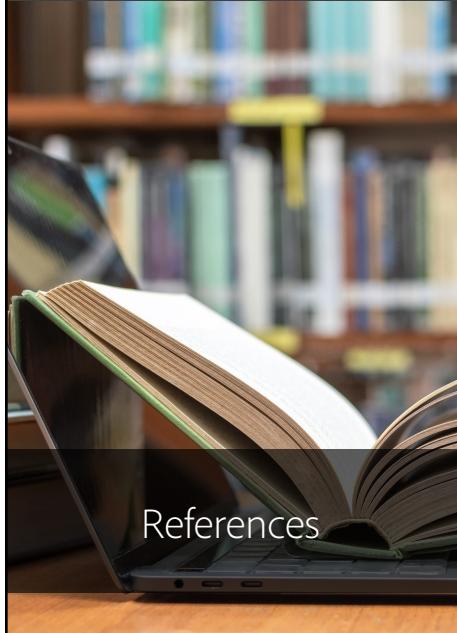
- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-2.html#styleguidist>

Note: Step-by-step instructions are available in the Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

154

ADOBEDIGITAL LEARNING SERVICES 



References

Hands-on Activity

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-2.html>

React and Sass

- <https://github.com/facebook/create-react-app/blob/master/packages/react-scripts/template/README.md#adding-a-sass-stylesheet>

Proxy API Requests in Development

- <https://github.com/facebook/create-react-app/blob/master/packages/react-scripts/template/README.md#proxying-api-requests-in-development>

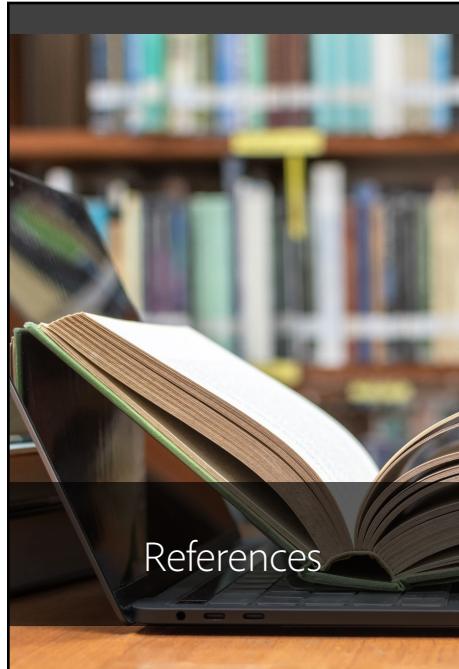
AEM Layout Mode

- <https://helpx.adobe.com/experience-manager/6-4/sites/authoring/using/responsive-layout.html>

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

155

ADOBEDIGITAL LEARNING SERVICES 



References

React Environment Variables

- <https://github.com/facebook/create-react-app/blob/master/packages/react-scripts/template/README.md#adding-custom-environment-variables>

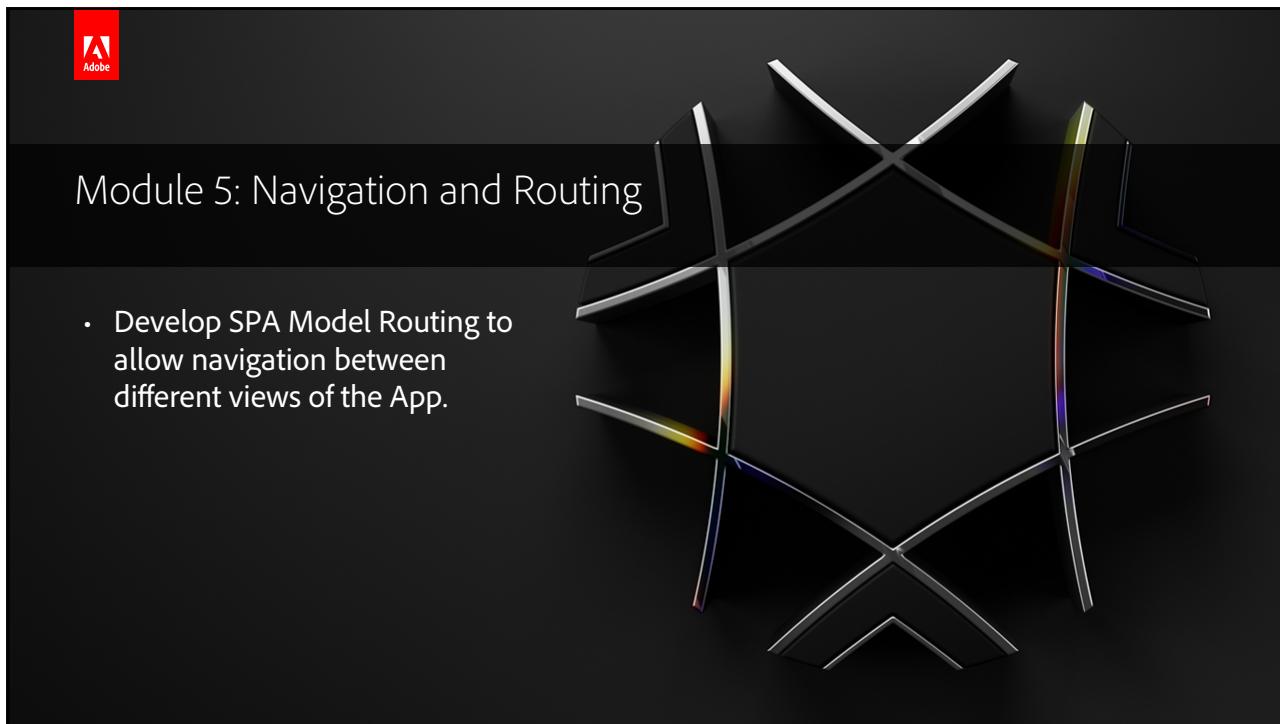
Styleguidist

- <https://react-styleguidist.js.org/>
- <https://github.com/styleguidist>
- <https://react-styleguidist.js.org/docs/getting-started.html>

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

156

ADOBEDIGITAL LEARNING SERVICES 



Module 5: Navigation and Routing

- Develop SPA Model Routing to allow navigation between different views of the App.

Routing in a Single Page Application

Routing enables navigating the app through links.

We want the behaviors expected in normal web applications:

- Browser should change the URL when you navigate to a different screen.
- Deep linking should work
 - A bookmark for a page in the website, should render that page, not the website/application home page
- Browser back and forward buttons should work as expected.

Routing links your application navigation with the navigation features offered by the browser:

- address bar
- navigation buttons

ModelRouter

When enabled, it encapsulates the HTML5 History API functions.

- **pushState**
- **replaceState**

Internally triggers the PageModelManager to load the model that corresponds to the specified URL.

- Guarantee a given fragment of model is pre-fetched and accessible

Notifies the registered Front-end component that the model has been modified.

- cq-pagemode-route-changed event

Manual vs Automatic Model Routing

Enabled ModelRouter automates the fetching of fragments of the model.

ModelRouter can be **disabled** or configured to ignore paths using meta properties.

- Enabled by default
- Disable `<meta property="cq:pagemode1_router" content="disable"\>`

Front-end developers can implement their own model routing layer by requesting that the **PageModelManager** load any given fragment of the model using the **getData()** function.

Meta Properties

cq:wcmode: WCM mode of the editors (for example, page, template).

cq:pagemode1_root_url: URL of the root model of the App.

cq:pagemode1_router: Enable or disable the ModelRouter of the PageModelManager library.

cq:pagemode1_route_filters: Comma-separated list or regular expressions to provide routes that the ModelRouter must ignore.

Sling Model JSON Exported Structure Configuration

When routing is enabled, assume that the JSON export of the SPA contains the different routes of the application.

- Due to the JSON export of the AEM navigation component

JSON output of the AEM navigation component can be configured in the SPA's root page content policy through the following two properties:

- **structureDepth**: Number corresponding to the depth of the tree exported
- **structurePatterns**: Regex or array of regexes corresponding to the page to be exported

This can be shown in the SPA sample content in:

`/conf/wknd-events/settings/wcm/policies/wknd-events/components/structure/app/default`

SPA Routing Approach

Map different views within the SPA to respective pages within AEM.

- Easy to manage multiple parts of the application
- Content authors can edit individual views

```

1 // psuedo code in JSX to represent the SPA
2
3 <BrowserRouter>
4   <App>
5     <Route path="/content/wknd-events/react/home">
6       <WkndPage cqPath="/content/wknd-events/react/home" />
7     </Route>
8     <Route path="/content/wknd-events/react/home/child1">
9       <WkndPage cqPath="/content/wknd-events/react/home/child1" />
10    </Route>
11    <Route path="/content/wknd-events/react/home/child2">
12      <WkndPage cqPath="/content/wknd-events/react/home/child2" />
13    </Route>
14  </App>
15 </BrowserRouter>

```

<Route /> elements for each page shown in the JSX pseudo code

Each AEM Page represented in the App will be wrapped in a <Route> with a path of the AEM page.

SPA Routing Approach (cont.)

Review:

- App is driven-based on a JSON model provided by AEM.
- JSON model uses HierarchyPage Sling Model to include the content of multiple AEM pages with a single request.

Benefits:

- Allows for most of the content of the App to be loaded on the initial page load.
- Should minimize subsequent server-side requests as a user navigates the app.

React Router

Standard routing library for React.

- Collection of navigational components.
- Provide and manage different views of the application.

Keeps UI in sync with the URL.

Simple API with powerful features built in:

- lazy code loading
- dynamic route matching
- location transition handling

React Router Modules

Library of 3 Modules:

- **react-router**: core package for the router.
- **react-router-dom**: DOM bindings for React Router, router components for the web app development environment.*
- **react-router-native**: native bindings for React Router, router components for an app development environment using React Native.

*We will be using **react-router-dom** because this app will be deployed to the web.

Routing Components - /src/util/RouteHelper.js

```

1 import React, {Component} from 'react';
2 import {Route} from 'react-router-dom';
3 import { withRouter } from 'react-router';
4
5 /* @ */
6 /* @ */
7 export const withRouter = (WrappedComponent, extension) => {
8   return class CompositeRoute extends Component {
9     render() {
10       let routePath = this.props.cgiPath;
11       if (!routePath) {
12         return <wrappedComponent {...this.props}/>;
13     }
14     extension = extension || 'html';
15     // Context path + route path + extension
16     return <Route key={ routePath } path={`${(.*).+ ${routePath} .+ ${extension}}` render={ (routeProps) => {
17       return <wrappedComponent {...this.props} {...routeProps}/>;
18     } } />;
19   };
20 }
21
22 /*-
23 * ScrollToTop component will scroll the window on every navigation.
24 * wrapped in in 'withRouter' to have access to router's props.
25 */
26 class ScrollToTop extends Component {
27   componentDidUpdate(prevProps) {
28     if (this.props.location !== prevProps.location) {
29       window.scrollTo(0, 0);
30     }
31   }
32   render() {
33     return this.props.children
34   }
35 }
36
37 export default withRouter(ScrollToTop);

```

withRoute can wrap any React component. Primarily used to wrap Page component to provide routing between pages

Import {Route} and {withRouter}

Scroll Restoration: Scroll to top of page on

BrowserRouter

Uses the HTML5 history API to keep the App UI in sync with the URL.

Allows easy deep linking to specific views of the App.

- *ScrollToTop* component from *RouteHelper.js* is included.

Wrapping the App - */src/index.js*

```

1 // src/index.js
2
3 import React from 'react';
4 import ReactDOM from 'react-dom';
5 import { ModelManager, Constants } from '@adobe/cq-spa-page-model-manager';
6 import './index.scss';
7 import App from './App';
8 //include Mapped Components
9 import './components/MappedComponents';
10 //wrap the app
11 import {BrowserRouter} from 'react-router-dom';
12 import ScrollToTop from './utils/RouteHelper';
13
14 function render(model) {
15   ReactDOM.render(
16     <BrowserRouter>
17       <ScrollToTop>
18         <ReactQuicksilver children={ model[Constants.CHILDREN_PROP] } cqItems={ model[Constants.ITEMS_PROP] } cqItemsOrder={ model[Constants.ITEMS_ORDER_PROP] } cqPath={ ModelManager.rootPath } locationPathname={ window.location.pathname } />
19       </ScrollToTop>
20     </BrowserRouter>
21     document.getElementById('root'));
22   }
23
24 ModelManager.initialize({ path: process.env.REACT_APP_PAGE_MODEL_PATH }).then(render);

```

wrap the App with
BrowserRouter and ScrollToTop

Wrap WkndPage component - */src/components/page/Page.js*

```

1  /* Page.js
2
3  Import withRoute specific implementation of Page
4  and map it to the AEM resource type
5
6  */
7
8  import {Page, MapTo, withComponentMappingContext} from "@adobe/cq-react-editable-components";
9  import {withRoute} from '../../utils/RouteHelper';
10
11 require('./Page.css');
12 // This component is a variant of a React Page component mapped to the "structure/page" resource type
13 // No functionality is changed other than to add an app specific CSS class
14 class WkndPage extends Page {
15
16   get containerProps() {
17     let attrs = super.containerProps;
18     attrs.className = (attrs.className || '') + ' WkndPage ' + (this.props.className || '');
19     return attrs;
20   }
21 }
22
23 MapTo('wknd-events/components/structure/page')(withComponentMappingContext(withRoute(WkndPage)));

```

170

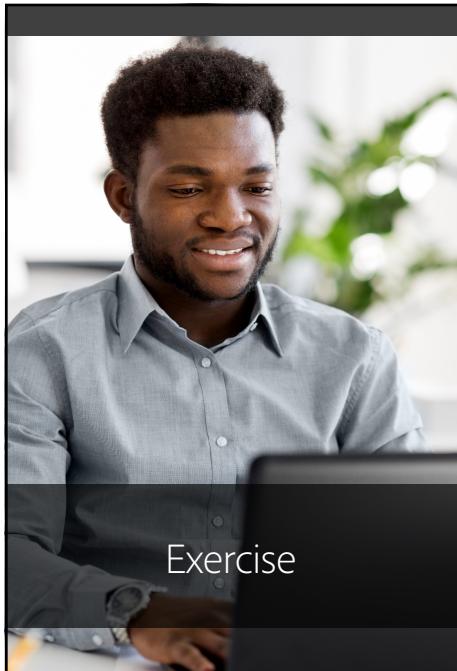
Routing Review

AEM resource **wknd-events/components/structure/page**.

- Represents an AEM Page.
- Will be mapped to the React component **WkndPage**.

Addition of **withRoute** causes all pages to be wrapped as a **Route**.

- Navigate to any page.



Install React Router

In this exercise, you will install and use react-router-dom. Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Pages 67 – 69

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-3.html#react-router>

Note: Step-by-step instructions are available in the Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

172 ADOBE DIGITAL LEARNING SERVICES 

172

Working with Links - /src/components/list/List.js

```

1 import React, {Component} from 'react';
2 import {MapTo} from '@adobe/cq-react-editable-components';
3 import {List} from 'react-router-dom';
4 require('./list.scss');
5
6 //EditConfig to be used by the AEM editor
7 const ListEditConfig = {
8   emptyLabel: 'List',
9   isEmpty: function(props) {
10     return !props || !props.items || props.items.length < 1;
11   }
12 };

```

Map React List component to AEM Core List component and EditConfig for AEM Editor

```

44 /**
45  * List renders the list content
46 */
47 export default class List extends Component {
48   render() {
49     return (
50       <div className="list">
51         <ol className="list-wrapper">
52           {this.props.items && this.props.items.map((listItem, index) => {
53             return <Listitem path={listItem.path} url={listItem.url} title={listItem.title} date={listItem.lastModified}>/>
54           })
55         </ol>
56       </div>
57     );
58   }
59 }
60
61 MapTo("wknd-events/components/content/list")(List, ListEditConfig);

```

List expects an array named `items`. Each element in the array will be passed to `Listitem` for rendering.

```

36 /**
37  * ListItem renders the individual items in the list
38 */
39 class Listitem extends Component {
40   get date() {
41     if(this.props.date) {
42       return null;
43     }
44     let date = new Date(this.props.date);
45     return date.toLocaleDateString('en-US');
46   }
47   render() {
48     if(!this.props.path || !this.props.title || !this.props.url) {
49       return null;
50     }
51     return (
52       <li className="ListItem" key={this.props.path}>
53         <Link className="Listitem-link" to={this.props.url}>{this.props.title}</Link>
54         <span className="Listitem-date">{this.date}</span>
55       </li>
56     );
57   }
58 }

```

`Listitem` expects `title`, `url`, `path`, and `date` as `props`

<code><Link /></code> from `react-router-dom` provides look and feel of a standard HTML link, but uses React Router to navigate to a different part of the App without a page refresh

173

Routing and Linking

The screenshot shows two browser windows side-by-side. Both have a yellow header bar with the text 'WKND'.

Left Window (Home Page):

- URL: `localhost:4502/editor.html/content/wknd-events/react/home.html`
- Content: A 'Home' page featuring a blue biplane on a snowy mountain background. Below it is a list of articles:

 - First Article** (published 12/8/2019)
 - Second Article** (published 12/5/2019)

Right Window (Article Preview):

- URL: `localhost:4502/editor.html/content/wknd-events/react/home.html`
- Content: A preview of the 'First Article' page, which also features a blue biplane on a snowy mountain background.

A blue callout bubble points from the right window to the left window, containing the text: "In Preview mode, URL did not change when following the link".

174

Routing and Linking (cont.)

The screenshot shows two browser windows side-by-side. The left window has a yellow header bar with the text 'WKND'.

Left Window (Home Page):

- URL: `localhost:4502/editor.html/content/wknd-events/react/home.html`
- Content: A 'Home' page featuring a blue biplane on a snowy mountain background. Below it is a list of articles:

 - First Article** (published 12/8/2019)
 - Second Article** (published 12/5/2019)

Right Window (Article Page):

- URL: `localhost:4502/content/wknd-events/react/home/first-article.html`
- Content: The 'First Article' page, which also features a blue biplane on a snowy mountain background.

A blue callout bubble points from the right window to the left window, containing the text: "Outside of Editor, URL changes when following link and browser back arrow works".

175

Header Component - /src/components/header/Header.js

```

1 import React, { Component } from 'react';
2 import { Link } from "react-router-dom";
3 import { withRouter } from 'react-router';
4 // import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
5 import './Header.scss';
6 require('../utils/Icons');
7
8 class Header extends Component {
9
10    get homeLink() {
11        let currLocation;
12        currLocation = this.props.location.pathname;
13        currLocation = currLocation.substr(0, currLocation.length - 5);
14
15        if (this.props.navigationRoot && currLocation !== this.props.navigationRoot) {
16            return (<Back>);
17        }
18
19        return null;
20    }
21
22    render() {
23        return (
24            <Header>
25                <Header-wrapper>
26                    <Header-title>WKND<span className="Header-title--inverse">_</span></Header-title>
27                    <Header-tools>
28                        {this.homeLink}
29                </Header-tools>
30            </Header>
31        );
32    }
33
34}
35
36
37 export default withRouter(Header);

```

Add 'Back' option to Header if not on the home page

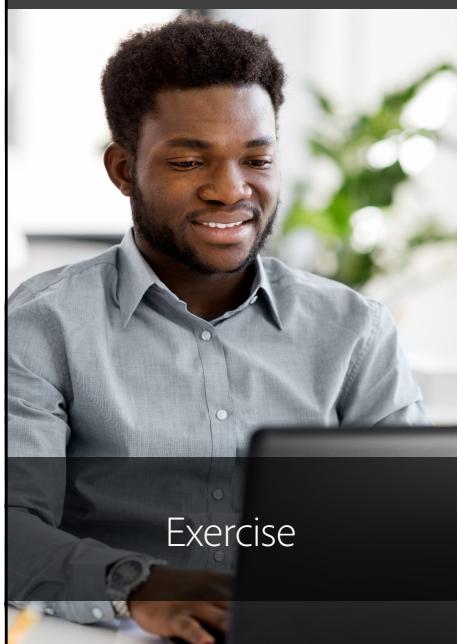
Export the Header wrapped in withRouter.

176

Header Component Back Option for Child Pages



177



Exercise

List Component

In this exercise, you will implement a React component to display a list of links.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-3.html#list-component>

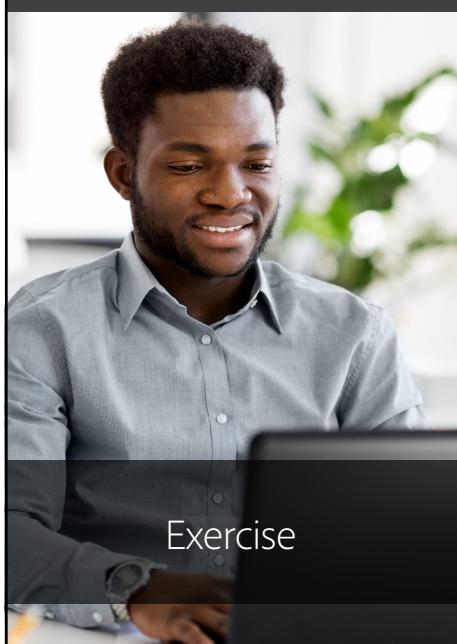
Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

178

ADOBEDIGITAL LEARNING SERVICES





Exercise

Update Header Component

In this exercise, you will add a Back button as part of the Header component.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-3.html#header-component>

Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

179

ADOBEDIGITAL LEARNING SERVICES



Font Awesome

Font Awesome provides an official React component to easily integrate the icon set into a React application.

- Popular icon set
- Open source
- Provides an official React Component that integrates icon set into a React application
- Install with **npm**

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

180



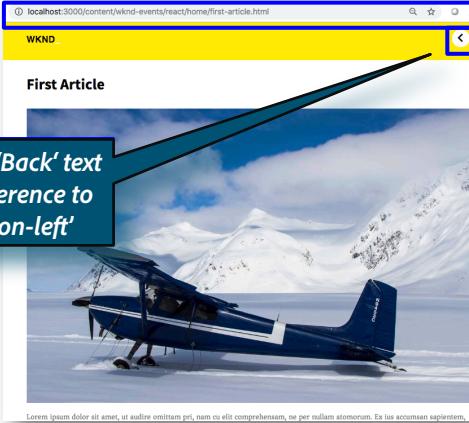
180

Use Font Awesome as Back Option - /src/components/header/Header.js

```

1 import React, { Component } from 'react';
2 import { Link } from 'react-router-dom';
3 import { withRouter } from 'react-router';
4 import { FontAwesomeIcon } from 'fortawesome/react-fontawesome';
5 import './Header.scss';
6 require('../utils/icons');
7
8 class Header extends Component {
9   get homeLink() {
10     let currLocation;
11     currLocation = this.props.location.pathname;
12     currLocation = currLocation.substr(0, currLocation.length - 5);
13   }
14   if (this.props.navigationRoot && currLocation !== this.props.navigationRoot) {
15     return (
16       <FontAwesomeIcon icon="chevron-left" />
17     );
18   }
19   return null;
20 }
21
22 render() {
23   return (
24     <header className="Header">
25       <div className="Header-wrapper">
26         <h1 className="Header-title">WKND<span className="Header-title--inverse">_</span></h1>
27         <div className="Header-tools">
28           { this.homeLink }
29         </div>
30       </div>
31     </header>
32   );
33 }
34
35 }
36
37 export default withRouter(Header);

```



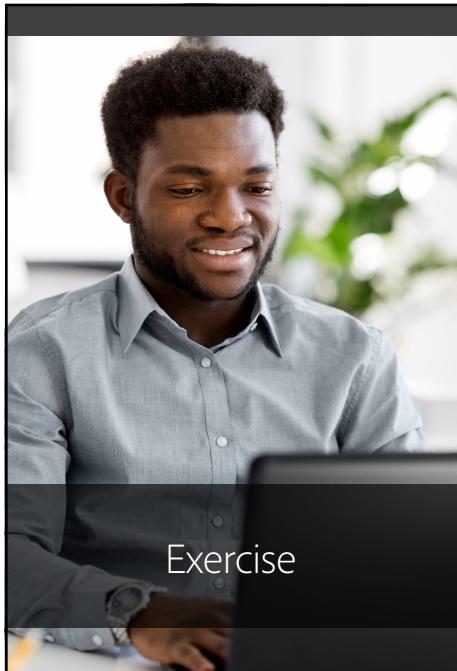
Replace 'Back' text with reference to 'chevron-left'

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

181

ADobe DIGITAL LEARNING SERVICES 

181



Add Font Awesome Icons

In this exercise, you will integrate a few of the icons into your project.

Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Resources:

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-3.html#font-awesome>

Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

182

ADOBEDIGITAL LEARNING SERVICES 

Redirects - */src/App.js*

```

1 import React from 'react';
2 import { Page, withModel, EditorContext, Utils } from '@adobe/cq-react-editable-components';
3 import { Redirect } from 'react-router';
4 import Header from './components/header/Header';
5
6 /**
7  * Returns a model path from the given URL
8  * @param {string} url - Path from which to extract a model path
9  * @return {string|undefined}
10 */
11
12 function getModelPath(url) {
13   if (!url) {
14     return;
15   }
16
17   let dotIndex = url.indexOf(".");
18   return url.substr(0, dotIndex > -1 ? dotIndex : url.length);
19 }
20
21 /**
22  * Should the App redirect to the home page
23  * @param {string} modelUrl - Path of the current model
24  * @return {boolean}
25 */
26
27 function canRedirectHome(modelUrl, pathname) {
28   if (!pathname) {
29     return false;
30   }
31   const currentUrl = getModelPath(pathname);
32   // 1. if a model url has been provided
33   // 2. if the current URL is located under the content
34   // 3. if app root model path equals the current URL
35   return modelUrl && modelUrl.indexOf('/content/') > -1 && (modelUrl === currentUrl || modelUrl.endsWith(currentUrl));
36 }

```

getModelPath() - strip URL extension and return path

canRedirectHome() - determine if app should redirect to Home view

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

183

ADOBEDIGITAL LEARNING SERVICES 

Redirects - `/src/App.js`. (cont.)

```

37 // This component is the application entry point
38 class App extends Page {
39
40     get redirect() {
41         const modelRootPath = this.props.cqPath;
42         const locationPathname = this.props.location.pathname;
43
44         if (canRedirectHome(modelRootPath, locationPathname)) {
45             // Redirect to the home url
46             return <Redirect to={ modelRootPath + '/home.html' }/>;
47         }
48         return null;
49     }
50
51     render() {
52         return (
53             <div className="App">
54                 <Header navigationRoot="/content/wknd-events/react/home" />
55                 <EditorContext.Provider value={ Utils.isInEditor() }>
56                     { this.redirect }
57                     { this.childComponents }
58                     { this.childPages }
59                 </EditorContext.Provider>
60             </div>
61         );
62     }
63 }
64
65 export default withModel(App);

```

redirect() - redirect to Home if it makes sense

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

184

ADOBEDIGITAL LEARNING SERVICES

184

Redirect using App.js

<http://localhost:3000/content/wknd-events/react.html>
redirects to
<http://localhost:3000/content/wknd-events/react/home.html>

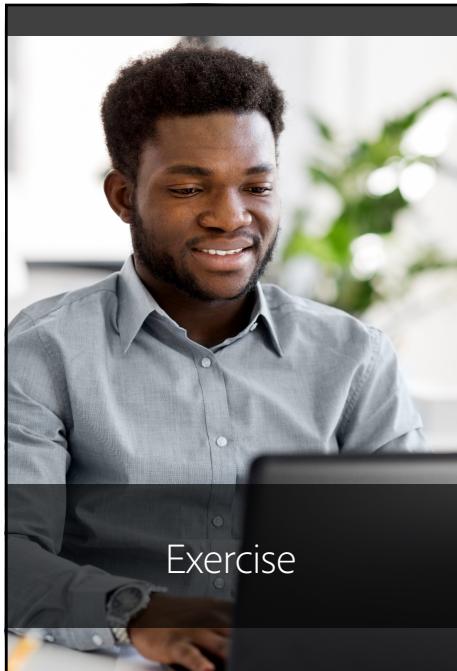
<http://localhost:4502/content/wknd-events/react.html>
redirects to
<http://localhost:4502/content/wknd-events/react/home.html>

The screenshot shows two separate browser windows. The top window has a blue header bar with the URL 'localhost:3000/content/wknd-events/react.html'. Below the header, there's a yellow header with 'WKND' and a main content area titled 'Home' with two articles: 'First Article' (1/28/2019) and 'Second Article' (1/29/2019). The bottom window has a blue header bar with the URL 'localhost:4502/content/wknd-events/react.html'. Below the header, there's a yellow header with 'WKND' and a main content area titled 'Home' with the same two articles. Both windows show a preview of the 'Second Article' content.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

185

ADOBEDIGITAL LEARNING SERVICES



Exercise

Add Redirect to Index.js
In this exercise, you will update Index.js to automatically redirect to the Home page view.
Your instructor will provide you with specific instructions for this exercise and answer any questions you may have.

Resources:

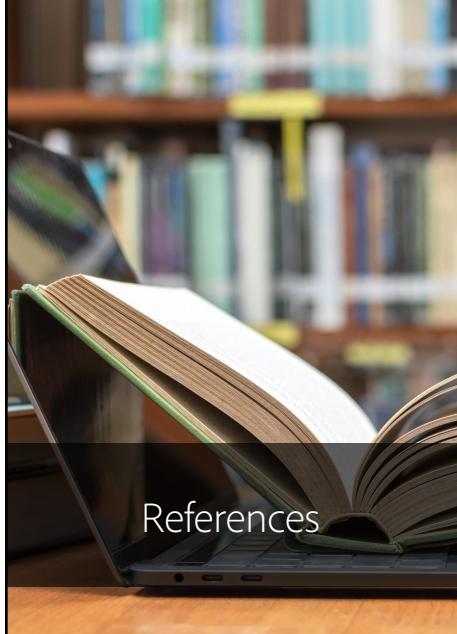
- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-3.html#redirect-app>

Note: The step-by-step instructions are available in your Student Guide.

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

186

ADOBEDIGITAL LEARNING SERVICES 



References

Hands-on Activity

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-3.html>

SPA Routing Approach

- <https://helpx.adobe.com/experience-manager/6-4/sites/developing/using/spa-routing.html>

React Router

- <https://www.npmjs.com/package/react-router>
- <https://reacttraining.com/react-router/web/example/basic>

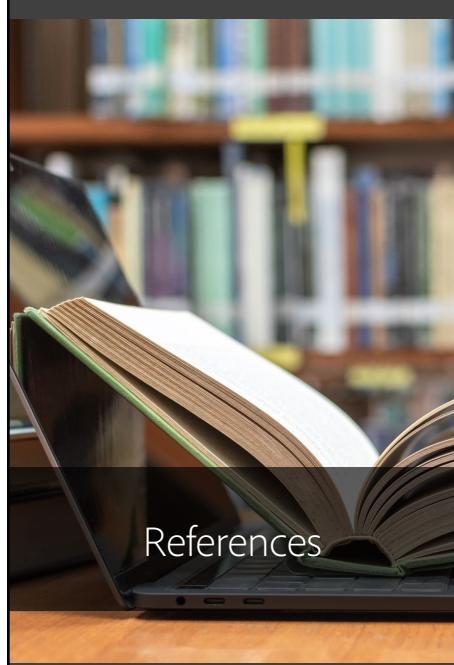
Hierarchy Page Implementation

- <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop/react/chapter-1.html#hierarchypagesling-model>

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

187

ADOBEDIGITAL LEARNING SERVICES 



Scroll Restoration

- <https://reacttraining.com/react-router/web/guides/scroll-restoration>

Troubleshooting

- <https://helpx.adobe.com/experience-manager/kt/platform-repository/using/local-aem-dev-environment-article-setup.html#troubleshooting>

© 2019 Adobe. All Rights Reserved. Adobe Confidential.

188

ADOBEDIGITAL LEARNING SERVICES



188



189