Additional Tasks

1. Grid Parsys
2. Hero Component
3. Breadcrumb Component
4. Search field
5. Events Planner

**Note**: For all tasks use Sightly & Sling models. Create dialogs for Classic & Touch UI.

Grid Parsys

Implement grid from parsys which would contain the same “Grid Parsys” and other components. Dialog would allow to set count of rows & columns.

**Note**: It is necessary to delete unused nodes after editing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | component |  | |  | component | | component | component |
| component |  | |  |  |  | | --- | --- | --- | |  | component |  | |

Hero Component

Implement component, which contains image, plain text, rich text and link. Dialog would have 3 tabs: 1 - for image (possible to drop or upload image, try disabled mode); 2 -plain text (simple text field), rich (rich text component with formatting, links & etc); 3 - link, link text (possibility to select path only from your site and selection where to open this link: blank, new tab... ).

|  |  |
| --- | --- |
| Image | header (plain text) |
| text (rich text with formatting & links) |
| link (just link) |

Breadcrumb Component

* Implement page structure with inheritance. Should have three types: master page, home page and content page. Master is the parent page for other types it contains common things like header, footer, libs etc. Home page is the root of your site and all other pages will be stored under it. Content page is just regular page with some data. Avoid creating unnecessary templates.
* Implement component for navigation. [Home -> Page1 -> Page2 -> … -> Parent of current page -> Current(not link) ] For ${Page%d} - ‘Navigation Title’ should be used if no then ‘Page Title’ -> Title. Pages are marked as “Hide in Navigation” should be skipped.

Search field

Create search field component included in header implemented in previous task. Component should have implementation of XPath, SQL2, Predicates. Corresponding engine is selected in dialog. The search should be performed on the child nodes.

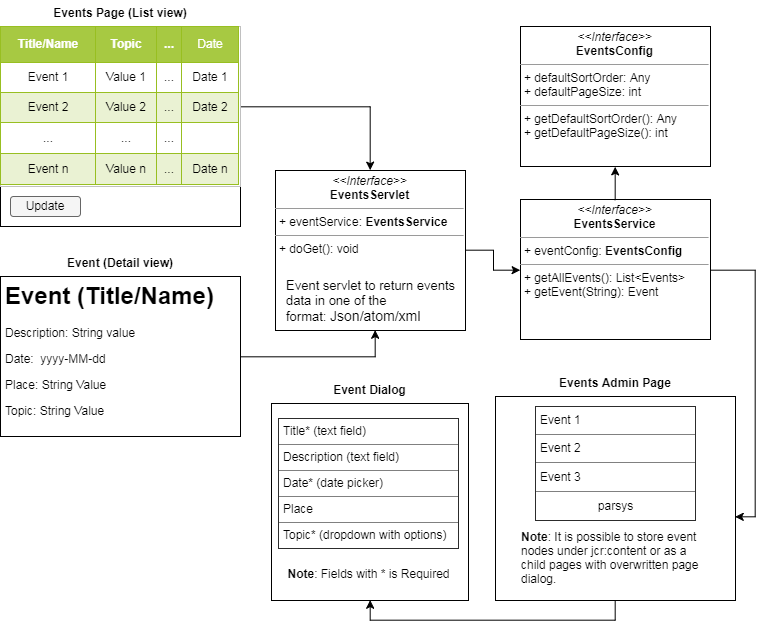
|  |  |
| --- | --- |
| Input field | Search |

|  |
| --- |
| table/list with results |

Events Planner

Create service to provide event management.

1. Event component (Date should be Date)
2. Event admin page
3. Service to provide data (should be configurable)
4. “Render” servlet and services to provide data in corresponding format (xml, json)
5. Pages to show events table
6. Add sorting to column headers (Event viewer).
7. Add pagination (Event viewer).



During training you may need to prepare maven project for AEM. The easiest way is to use Maven Archetype.

It is possible to use:

1. <https://kb.epam.com/display/EPMAEM/Maven+Archetype>
2. AEM archetype: mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate -DarchetypeGroupId=com.adobe.granite.archetypes -DarchetypeArtifactId=aem-project-archetype -DarchetypeVersion=11 -DarchetypeCatalog=https://repo.adobe.com/nexus/content/groups/public

**Note**: Artifact generated by this archetypes requires an unobfuscated uber jar from Adobe to be installed and available in local (or any other configured) Maven repository

Unobfuscated version of uber jar can be downloaded directly from Adobe website, DayCare access required, or requested form [EPAM Content Competency](mailto:OrgCompetencyContentServiceDesk@epam.com)

Uber jar should be installed into maven repo using following command (exmample for 6.2)

mvn install:install-file -Dfile=cq-quickstart-6.2.0-apis.jar -DgroupId=com.adobe.aem -DartifactId=uber-jar -Dversion=6.2.0 -Dclassifier=apis -Dpackaging=jar