

# AEM SPA Editor - Deep Dive - Part. 2 Angular

Artur Kudlacz

#AdobeRemix  
Jon Noorlander

# Promotion

- Introduction to the AEM SPA Editor  
<http://bit.ly/spa-editor-intro>
- Deep Dive - Part. 1 React  
<http://bit.ly/spa-deep-dive-react>
- We are working on the starter kit
- Open sourcing is under discussion



The Headless Horseman: a strange tale of Texas.  
"Hain't ye forgot to fetch yur head we' ye?"

# Agenda

- Project Structure
- AEM Page Component
- Content Services
- Model Manager
- SPA Initialization
- Frontend Component
- Dynamically instantiated component
- Container components
- App Routing and Model Router



# Project Structure

#AdobeRemix  
Jon Noorlander

# Project Structure

## Breakdown

- AEM area
  - Bundles
    - AEM Component **Sling Model**
    - **Servlets** (e.g. Server-Side Rendering)
  - Content
    - AEM Component HTML templates
    - Editable **Templates** and **Content Policies**
    - **Content Structure** and component data of the App
- Frontend area
  - Frontend components **mapped** to AEM component resource types



# Project Structure

Team and project composition

- **Single project** with modules
  - The frontend project is one of the modules
- **Two projects**
  - A project for the **AEM modules**
  - A project for the **frontend App**



# Project Structure

Client library packaging and delivering strategies

- Copy in the content directory (full-stack project structure type)  
[`aem-clientlib-generator`](#) (*npm*)
- Direct Upload (any project structure type)  
[`aemfed`](#) (*npm*)





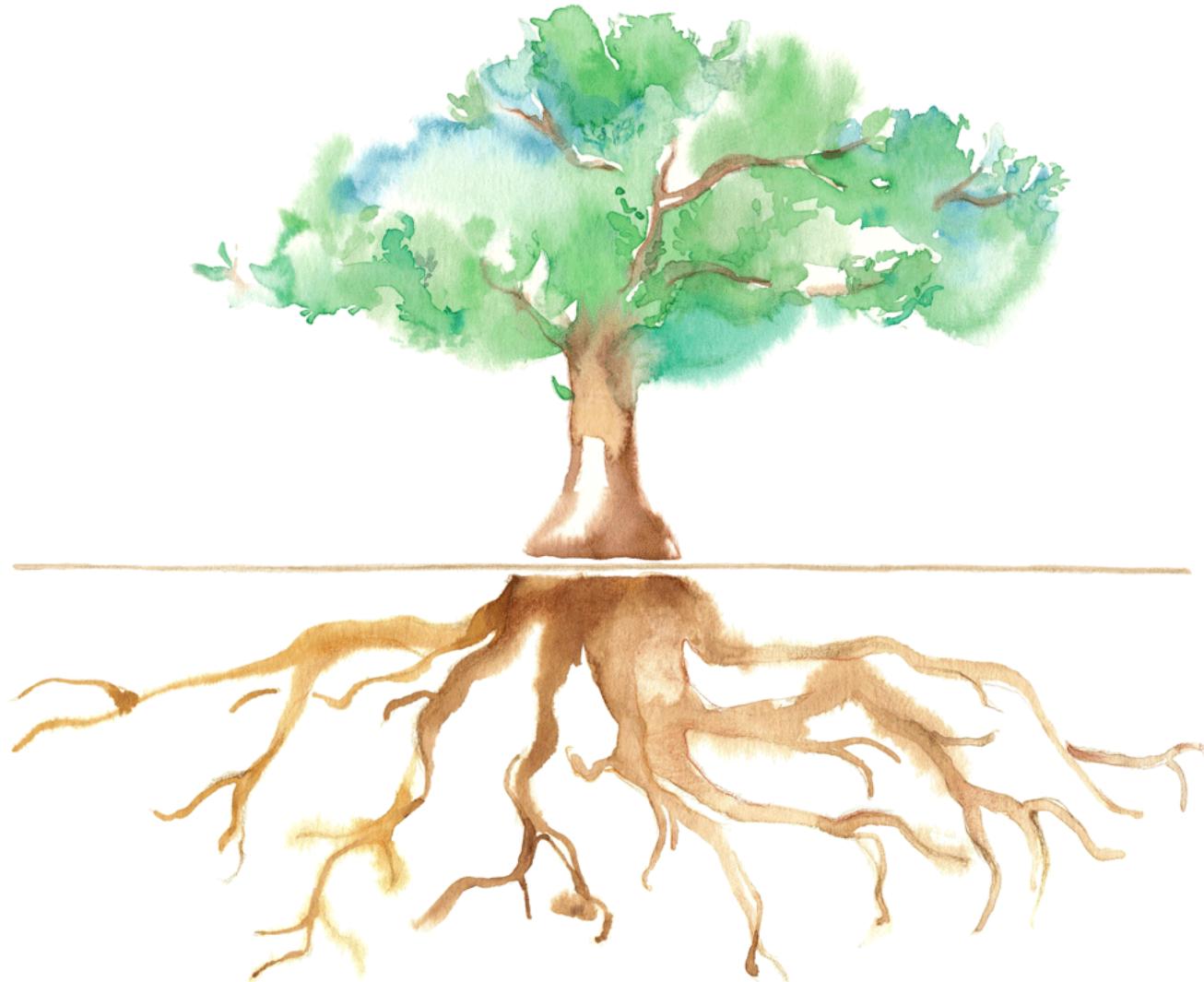
# AEM Page Component

#AdobeRemix  
Jon Noorlander

# AEM Page Component

## Root model

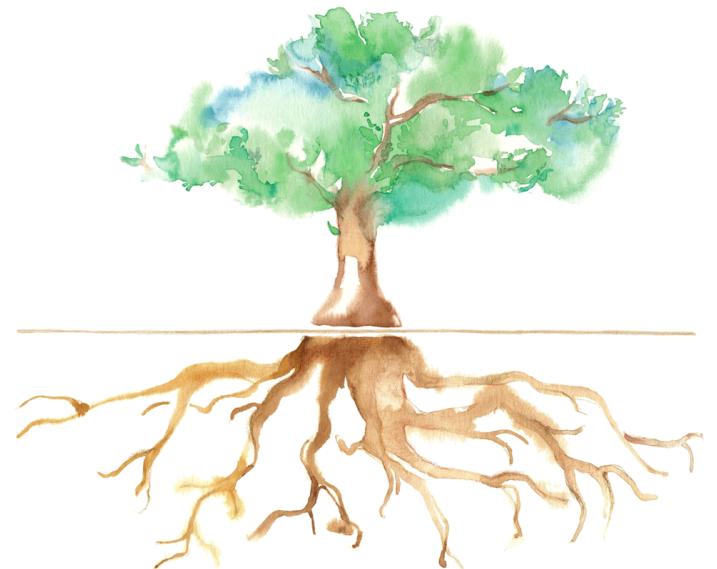
- Initial model of the SPA
  - shared content
  - **pre-loaded** page models
  - parts can be loaded **asynchronous**



# AEM Page Component

## HTML template

- Meta properties
  - Initialize the **PageModelManager**
    - **cq:datatype** – format in which to fetch the data and to update the page model (JSON)
    - **cq:wcmmode** – WCMMode **edit** or **preview** (author instance only)
    - **cq:pagemodel\_root\_url** – URL to the model of the App root (incl. selectors)



# AEM Page Component

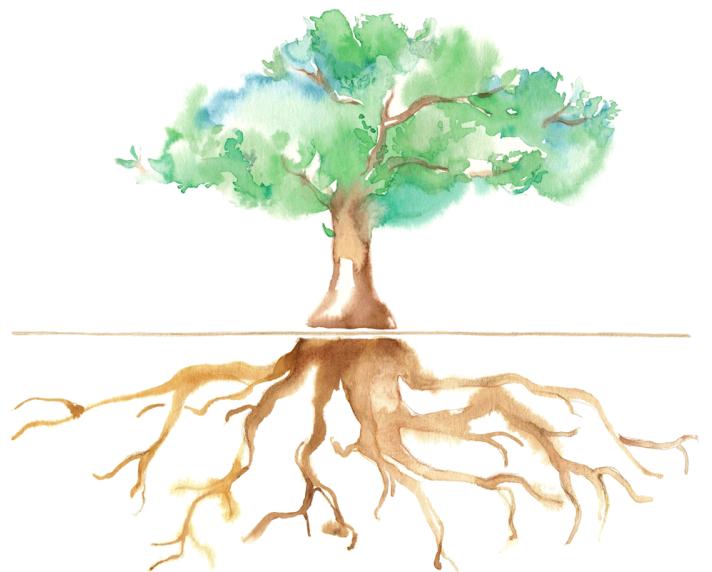
## HTML template

- Client library categories
  - Files: `customheaderlibs.html`, `customfooterlibs.html`

```
data-sly-call="${clientLib.js @ categories='cq.authoring.pagemodel.messaging'}"  
data-sly-call="${clientlib.<js|css|all> @ categories='my-spa-category'}"
```

- Body of the page
  - File: `body.html`

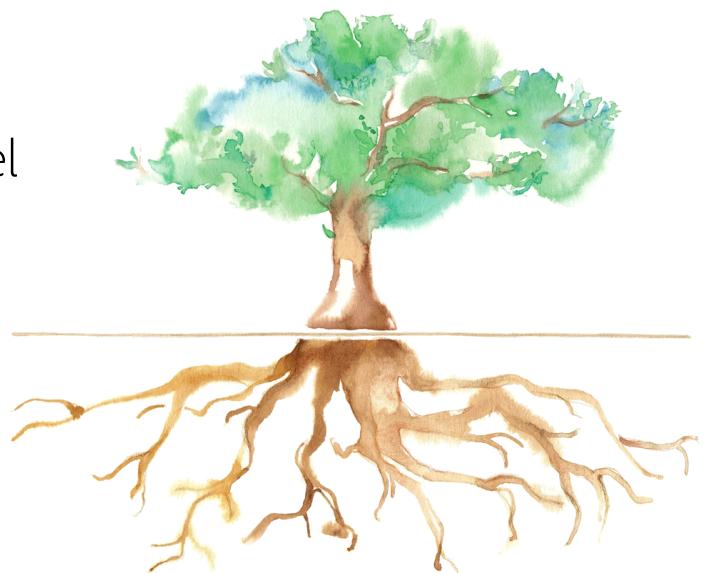
```
<app-root></app-root>
```



# AEM Page Component

## Hierarchy Page

- Initial JSON representation of the App
- Instant and asynchronous model
- Flat list of child pages (root model)
- Content Policy properties
  - **isRoot** – identifies the page as the **root** of an App
  - **structureDepth** – max depth of root model
  - **structurePatterns** – list of **regexp** to filter out pages from the root model





# Content Services

#AdobeRemix  
Jon Noorlander

# Content Services

## Sling Model Exporter

- Selector: **model**
- Must be mentioned **first** when using **multiple** selectors
- Default name
- Also used by the **Core Components**

```
cq:page model_root_url = /content/my-site/my-page.model.json
```

# Content Services

- JSON representation of the content
  - Leverage the Sling Model Exporter
- Meta fields (reserved)
  - **:type** – Resource type of AEM component
  - **:items** – Map of child components
  - **:itemsOrder** – Ordered list of the item keys
  - **:children** – Map of child pages
  - **:path** – Path to the resource

```
{  
  :path: "/content/we-retail-journal/angular",  
  :hierarchyType: "page",  
  - :children: {  
    + /content/we-retail-journal/angular/home: {...},  
    + /content/we-retail-journal/angular/blog: {...},  
    + /content/we-retail-journal/angular/blog/aboutus: {...}  
  },  
  title: "We.Retail Journal",  
  :type: "we-retail-journal/angular/components/structure/app",  
  - :items: {  
    + root: {...},  
    + image: {...}  
  },  
  - :itemsOrder: [  
    "root",  
    "image"  
  ]  
}
```

# Content Services

## Sling Model

- **Mandatory** for the SPA SDK
- Schema of the component
- Business logic
- **export.json** API
  - **ComponentExporter**
  - **ContainerExporter**
  - **HierarchyNodeExporter**
- Core components
  - Sling models
  - Uses the **export.json** API



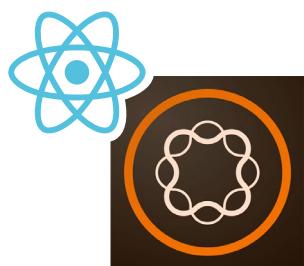
# Model Manager

#AdobeRemix  
Jon Noorlander

# Model Manager

## ModelManager

- **Stores** the model
- **Caches** the model
- Allows to **asynchronously load** fragments of model
- Returns the **immutable** model of a component
- Is updated by the Page Editor
- Can **force reload** the model of a component



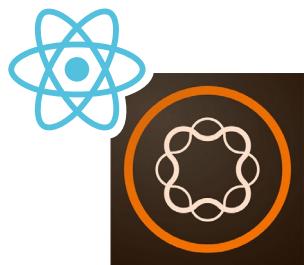


# SPA Initialization

#AdobeRemix  
Jon Noorlander

# SPA Initialization

- Explore the frontend project
- Project bootstrap
  - Initialize the **ModelManager** (`app.component.ts`)
    - Meta properties
    - URL of the root model
    - **ModelClient**
  - Instantiate the components



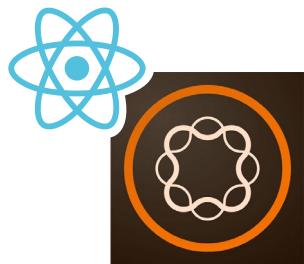
# SPA Initialization

Dynamic instantiation of the complete page

```
@Component({  
    selector: 'app-main',  
    template: `<aem-page class="structure-page"  
        [attr.data-cq-page-path]="path" [cqPath]="path"  
        [cqItems]="items" [cqItemsOrder]="itemsOrder"></aem-page>`  
})  
export class MainContentComponent { ... }
```

Why do we have to set so many fields?

- Done once when **bootstrapping** the App
- We already receive the model after the initialization of the **ModelManager**



# SPA Initialization

Static instantiation of a component

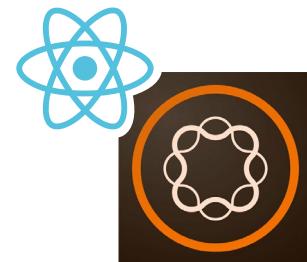
- **aemComponent** directive

```
<div [aemComponent] [cqItem]='model' [cqPath]='path'></div>
```

- Does not update the rendering if model is changed
- Need to manually get the model from model manager
- **aemModelProvider** directive

```
<div [aemModelProvider] [cqItem]='model' [cqPath]='path'></div>
```

- Updates the rendering if model changes
- We are considering enhancing that component would automatically grab the model





# Frontend Component

#AdobeRemix  
Jon Noorlander

# Frontend Component

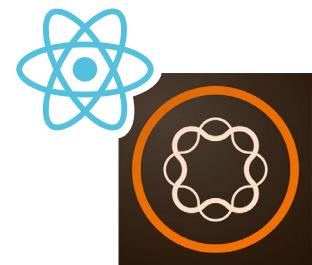
## Registering component

- **MapTo**
  - Maps an AEM component type to a Frontend component class
  - Stores the class in the **ComponentMapping**

## Configure the authoring experience

- **EditConfig**
  - **emptyLabel** – Label for the placeholder of an empty component
  - **isEmpty** – Is the (AEM) component empty

```
export MapTo('component/resource/type')(Component, EditConfig)
```





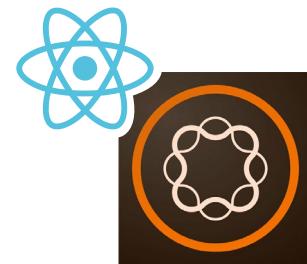
# Dynamically instantiated component

#AdobeRemix  
Jon Noorlander

# Dynamically instantiated component

## aemComponent directive

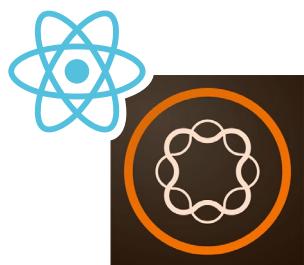
- Responsible of dynamically instantiated component
  - **cqItem** – model of the component to render
  - **cqPath** – path of the item to render
  - **itemName** – name of the item in the parent model
  - **itemAttrs** – extra attributes that will be rendered on the component
- Model fields are **transformed** to component inputs  
(**:items** to **cqItems**, ...)



# Dynamically instantiated component

## aemModelProvider directive

- Responsible of dynamically instantiated component
  - **cqItem** – model of the component to render
  - **cqPath** – path of the item to render
  - **itemName** – name of the item in the parent model





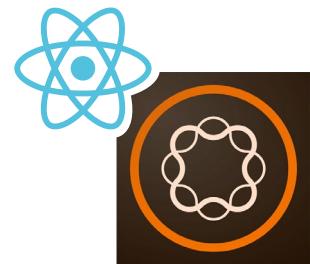
# Container components

#AdobeRemix  
Jon Noorlander

# Container components

## Components

- Extensible components
- **Container**
  - Iterates over the items and dynamically instantiated each of the components
- **ResponsiveGrid**
  - Provides access to the layout class names
- **Page**
  - Dynamic instantiation of **child** pages – iterates over **:children** field of the model
  - Also **auto-updates**, as it's extending **ResponsiveGrid**



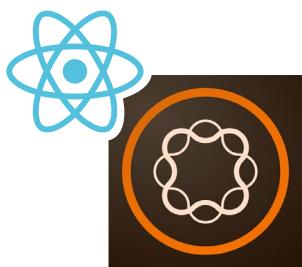


# App Routing and Model Router

#AdobeRemix  
Jon Noorlander

# App Routing and Model Router

- App **Routing** relates to the **navigation** the App
- The Model Router
  - **Pre-fetch** fragments of model
  - Listen for calls to the **History API**
  - **Enabled** by default
- In the current sample we are using static app routing
- In the near future we will switch to a dynamic routing strategy



# Resources

- Tutorials and presentations
  - <https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-spa-wknd-tutorial-develop.html>
  - <http://bit.ly/spa-editor-intro>
  - <http://bit.ly/spa-deep-dive-react>
- SPA Editor Overview  
<https://helpx.adobe.com/experience-manager/6-4/sites/developing/using/spa-overview.html>
- Sample – We retail Journal  
<https://github.com/adobe/aem-sample-we-retail-journal>
- Client library packaging
  - <https://www.npmjs.com/package/aem-clientlib-generator>
  - <https://www.npmjs.com/package/aemfed>
- Core components delegation pattern  
<https://github.com/Adobe-Marketing-Cloud/aem-core-wcm-components/wiki/Delegation-Pattern-for-Sling-Models>



**Adobe**