



adaptTo()

EUROPE'S LEADING AEM DEVELOPER CONFERENCE

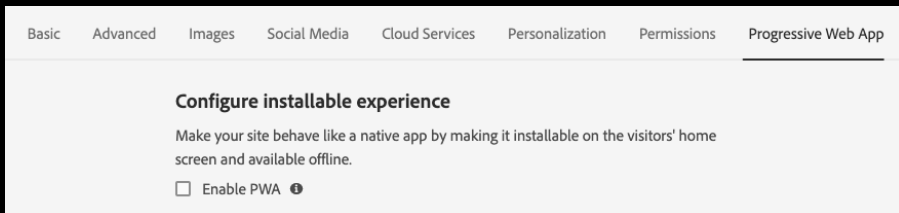
27th – 29th SEPTEMBER 2021

Future of AEM Core Components

Jean-Christophe Kautzmann, Adobe

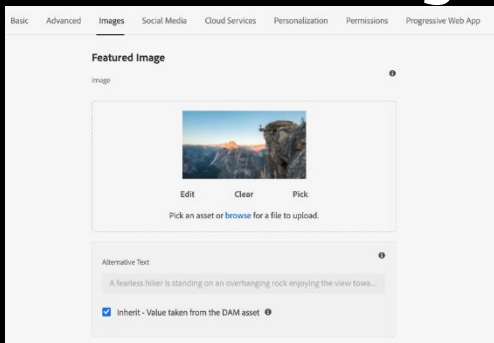
Features released over the last 12 months

- Link Handling
- Dynamic Media support (image)
image modifiers, image presets, and smart crops
- PWA support (page)



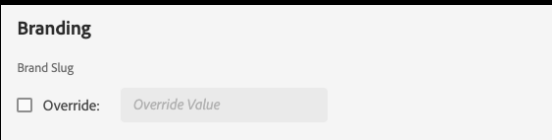
- Accessibility improvements

- **Featured Image (Tech Preview)**



- **Data Layer 2.0.2**
- **Brand Slug (page)**

```
<title>VW Model Lineup – Cars and SUVs | Volkswagen</title>
<title>News & Insights | CVS Health</title>
<title>Model 3 | Tesla</title>
```



Focus on the new link handler

- Contributed by the community (Stefan Seifert)
- Link API:
 - Validity: `link.isValid()`
 - URL: `link.getURL()`
 - HTML attributes: `link.getHtmlAttributes()`
 - Link reference: `link.getReference()`

```

/**
 * Check if the link defined for the component is valid.
 *
 * @return {@code true} if component has a valid link defined
 * @since com.adobe.cq.wcm.core.components.commons.link 1.0.0
 */
default boolean isValid() { return false; }

/**
 * The link URL, supports context path and escaping.
 *
 * @return Link URL or {@code null} if link is invalid
 */
@Nullable
default String getURL() { return null; }

/**
 * The mapped URL, which supports mapping and vanity path.
 * This usually is resource resolver mapping.
 *
 * @return Mapped link URL or {@code null} if link is invalid or no processing can be done
 */
@Nullable
default String getMappedURL() { return null; }

/**
 * The externalized URL which also contains the scheme and host information.
 * This is usually created with a {@link com.day.cq.commons.Externalizer} service
 * @return Full link URL or {@code null} if link is invalid or can't be externalized.
 */
@Nullable
default String getExternalizedURL() { return null; }

/**
 * Map with Attributes for HTML Anchor tag for this link.
 * This usually also contains the Link URL as <code>href</code> attribute,
 * but may contain additional attributes like <code>target</code> and others.
 *
 * @return {@link Map} with HTML-specific anchor attributes, or an empty map if link is invalid
 * @since com.adobe.cq.wcm.core.components.commons.link 1.0.0
 */
@NotNull
default Map<String, String> getHtmlAttributes() { return Collections.emptyMap(); }

/**
 * Returns the referenced WCM/DAM object.
 *
 * @return Target page or {@code null}
 * @since com.adobe.cq.wcm.core.components.commons.link 1.0.0
 */
@Nullable
default T getReference() { return null; }

```

The LinkHandler API

- The link handler can be adapted from the request

```
/**  
 * Simple implementation for resolving and validating links from model's resources.  
 * This is a Sling model that can be injected into other models using the @Self annotation.  
 */  
@Model(adaptables = SlingHttpServletRequest.class)  
public class LinkHandler {
```

- Usage:
 - `linkHandler.getLink(resource)`
 - `linkHandler.getLink(resource, linkURLPropertyName)`
 - `linkHandler.getLink(page)`
 - `linkHandler.getLink(linkURL, target)`
 - `linkHandler.getLink(linkURL, target, linkAccessibilityLabel, linkTitleAttribute)`

Customizing the link handling

- Customize the handling of links: the **PathProcessor**
- Customize how the links are sanitized, mapped and externalized
- Customize the HTML attributes

```

@ConsumerType
public interface PathProcessor {

    /**
     * Checks if the given path should be handled by this processor.
     * @param path the path which should be processed
     * @param request the current request
     * @return {@code true} if the processor should handle the given path, otherwise {@code false} and the next path processor is applied by
     * the {@link com.adobe.cq.wcm.core.components.internal.link.LinkHandler} if present
     */
    boolean accepts(@NotNull String path, @NotNull SlingHttpServletRequest request);

    /**
     * Sanitizes the given path by doing proper escaping and prepends the context path if needed.
     * @param path the path which needs to be sanitized
     * @param request the current request
     * @return the escaped absolute path with optional context path information
     */
    @NotNull String sanitize(@NotNull String path, @NotNull SlingHttpServletRequest request);

    /**
     * Applies mappings to the given path. Usually this is done with the {@link ResourceResolver#map(String)} method.
     * @param path the path which should be mapped
     * @param request the current request
     * @return the mapped path
     */
    @NotNull String map(@NotNull String path, @NotNull SlingHttpServletRequest request);

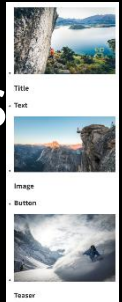
    /**
     * Externalizes the given path.
     * @param path the resource path
     * @param request the current request
     * @return the external link of the given path
     */
    @NotNull String externalize(@NotNull String path, @NotNull SlingHttpServletRequest request);

    /**
     * Processes the HTML attributes for the {@link com.adobe.cq.wcm.core.components.internal.link.LinkHandler}
     * @param path the path of the linked resource
     * @param htmlAttributes the origin HTML attributes of the link
     * @return a map of the processed HTML attributes for the link
     */
    default @Nullable Map<String, String> processHtmlAttributes(@NotNull String path, @Nullable Map<String, String> htmlAttributes) {
        return htmlAttributes;
    }
}

```

Currently in the making

- Capability to define a featured image at page level, with the image and the teaser components inheriting from this image by default (see upcoming demo).
- Extended list component, rendering the items as teasers.



Currently in the making

- Modernized and simplified image markup with **srcset** attribute
- Native lazy loading with **loading="lazy"** that no longer relies on component specific javascript.

```
▼ <div id="image-f9bb0ec349" data-cmp-data-layer="{\"image-f9bb0ec349\":{\"@type\":\"core-components-examples/components/image\",\"repo:modifyDate\":\"2021-09-01T09:53:31Z\"}}\" class=\"cmp-image cq-dd-image\" itemscope itemtype=\"http://schema.org/ImageObject\">  
  <img srcset=\"/content/core-components-examples/library/page-authoring/title/jcr_content/root/responsivegrid/image.coreimg.82.100.jpeg/1630490011893.jpeg 100w, /content/core-components-examples/library/page-authoring/title/jcr_content/root/responsivegrid/image.coreimg.82.200.jpeg/1630490011893.jpeg 200w, /content/core-components-examples/library/page-authoring/title/jcr_content/root/responsivegrid/image.coreimg.82.500.jpeg/1630490011893.jpeg 500w\" src=\"/content/core-components-examples/library/page-authoring/title/jcr_content/root/responsivegrid/image.coreimg.jpeg/1630490011893.jpeg\" loading=\"lazy\" class=\"cmp-image__image\" itemprop=\"contentUrl\" alt = $0  
</div>
```


Currently in the making

- Improved SEO capability for the page component

```
public String getCanonicalLink() {...}
public Map<Locale, String> getAlternateLanguageLinks() {...}
public List<String> getRobotsTags() {...}
```

- More accessibility improvements after a detailed accessibility review of all the components.
- More consistent building blocks across components on new version: HTML, JSON, data layer, dialogs, etc

What's next?

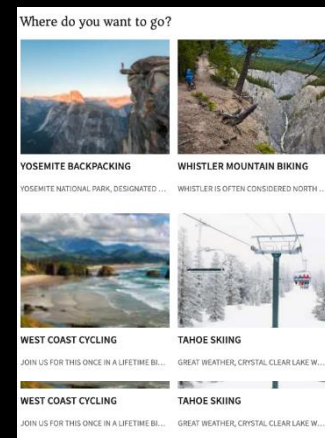
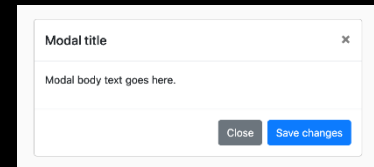
- Release 3.0.0 with new versions for the page, image, teaser and list components.
- Optimize page and image HTTP cache control headers for improved page performance.

What's next?

- New table of content component.
- New modal component.
- Modernized grid component.

ON THIS PAGE

- Usage
- Responsive Features
- Dynamic Media Support
- Version and Compatibility
- SVG Support
- Security
- Sample Component Output
- Technical Details
- Configure Dialog
- Asset Tab
- Metadata Tab
- Edit Dialog
- Design Dialog
- Main Tab
- [Features Tab](#)
- Styles Tab
- Adaptive Image Servlet
- Adobe Client Data Layer



- Github repo:
<https://github.com/adobe/aem-core-wcm-components>
- Discussions:
<https://github.com/adobe/aem-core-wcm-components/discussions>
- Road map, dev guide, style guide:
<https://github.com/adobe/aem-core-wcm-components/wiki>

Demo

Q & A



Thank you!