

Adobe CQ Help /

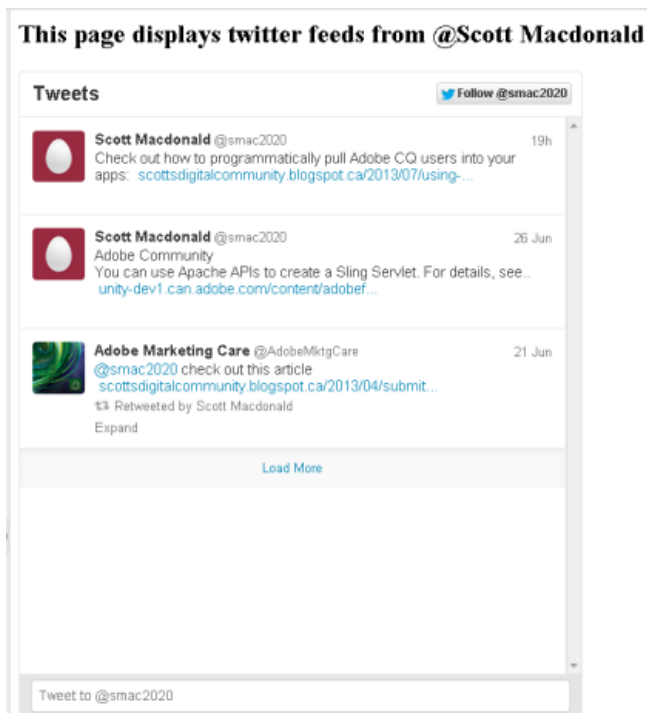
Creating Adobe CQ feed components that display social data

Article summary

| | |
|--------------------------------------|---|
| Summary | This article has been updated from the Twitter 1.0 API which has been deprecated to use Twitter widget code. This article discusses how to obtain Twitter widget code and how to use it within an Adobe CQ component. |
| Digital Marketing Solution(s) | Adobe Experience Manager (Adobe CQ) |
| Audience | Developer (intermediate) |
| Required Skills | JavaScript, JQuery, CSS |
| Tested On | Adobe CQ 5.5, Adobe CQ 5.6 |

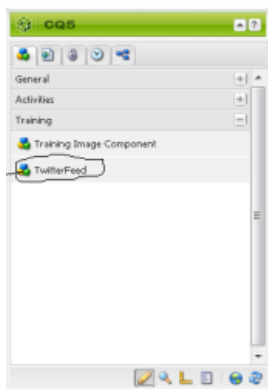
Introduction

You can create a custom Adobe CQ data feed component that retrieves external data and displays the data within a CQ web page. For example, a CQ component can retrieve data from an RSS feed to let your users view current forum data. Likewise, a CQ component can retrieve data from a social site like Twitter to keep your users updated with the latest social data, as shown here.



A Twitter feed component displaying tweets

This development article walks you through how to build a custom CQ twitter feed component and add it to the CQ sidebar.



The CQ sidekick containing a Twitter feed component

Once added to the CQ sidekick, a CQ author can drag and drop it onto a CQ page during design time. You create a Twitter feed component by integrating the Twitter API into Adobe CQ (shown in this development article).

Create a CQ component that integrates with the Twitter API by performing these steps:

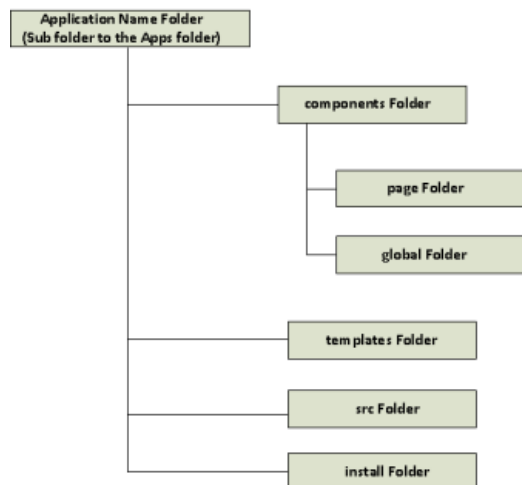
1. Create an Adobe CQ application folder structure.
2. Create a template on which the render component is based.
3. Create a render component that displays the CQ sidekick.
4. Get the Twitter widget code to use in your AEM application.
5. Create a site that contains a page that displays Twitter data.

Note: This development article discusses how to integrate the Twitter API to create a CQ component that displays Twitter feed data. You can integrate with other third party APIs such as Google Maps. For information, see [Integrating custom CQ widgets with third-party libraries](#).

Create a CQ application folder structure

[To the top](#)

Your first step is to create an CQ application folder structure that contains templates, components, and pages by using CRXDE Lite.



A CQ application folder structure

This list describes each application folder:

- **Application name:** Contains all of the resources that an application uses. The resources can be templates, pages, components, and so on.
- **components:** Contains components that your application uses.
- **page:** Contains page components. A page component is a script such as a JSP file.
- **global:** Contains global components that your application uses (common may be a substitute for global).
- **templates:** Contains templates that you can base page components on.
- **src:** Contains source code that comprises an OSGi component (this example does not use an

OSGi component).

- **install:** Contains a compiled OSGi bundles container.

To create an Adobe CQ application folder structure:

1. To view the CQ welcome page, enter the URL `http://[host name]:[port]` into a web browser. For example, `http://localhost:4502`.
2. Select CRXDE Lite (if you are using AEM 5.6, click Tools from the left menu).
3. Right-click the apps folder (or the parent folder), select Create, Create Folder.
4. Enter the folder name into the Create Folder dialog box. Enter *twitterFeed*.
5. Repeat steps 1-4 for each folder specified in the previous illustration.
6. Click the Save All button.

Note: You have to click the Save All button when working in CRXDE Lite for the changes to be made.

Create a template

[To the top](#)

You can create a template by using CRXDE Lite. A CQ template enables you to define a consistent style for the pages in your application. A template comprises of nodes that specify the page structure. For more information about templates, see <http://dev.day.com/docs/en/cq/current/developing/templates.html>.

To create a template, perform these tasks:

1. To view the CQ welcome page, enter the URL `http://[host name]:[port]` into a web browser. For example, `http://localhost:4502`.
2. Select CRXDE Lite (if you are using AEM 5.6, click Tools from the left menu).
3. Right-click the template folder (within your application), select Create, Create Template.
4. Enter the following information into the Create Template dialog box:
 - **Label:** The name of the template to create. Enter *templateTwitter*.
 - **Title:** The title that is assigned to the template.
 - **Description:** The description that is assigned to the template.
 - **Resource Type:** The component's path that is assigned to the template and copied to implementing pages. Enter *twitterFeed/components/page/templateTwitter*.
 - **Ranking:** The order (ascending) in which this template will appear in relation to other templates. Setting this value to 1 ensures that the template appears first in the list.
5. Add a path to Allowed Paths. Click on the plus sign and enter the following value: */content(/.*)?*.
6. Click Next for Allowed Parents.
7. Select OK on Allowed Children.

Create a render component that displays the CQ sidekick

[To the top](#)

Components are re-usable modules that implement specific application logic to render the content of your web site. You can think of a component as a collection of scripts (for example, JSPs, Java servlets, and so on) that completely realize a specific function. In order to realize this functionality, it is your responsibility as a CQ developer to create scripts that perform specific functionality. For more information about components, see <http://dev.day.com/docs/en/cq/current/developing/components.html>.

By default, a component has at least one default script, identical to the name of the component. To create a render component, perform these tasks:

1. To view the CQ welcome page, enter the URL `http://[host name]:[port]` into a web browser. For example, `http://localhost:4502`.
2. Select CRXDE Lite (if you are using AEM 5.6, click Tools from the left menu).
3. Right-click `/apps/twitterFeed/components/page`, then select Create, Create Component.
4. Enter the following information into the Create Component dialog box:
 - **Label:** The name of the component to create. Enter *templateTwitter*.
 - **Title:** The title that is assigned to the component.
 - **Description:** The description that is assigned to the template.
5. Select Next for Advanced Component Settings and Allowed Parents.
6. Select OK on Allowed Children.
7. Open the *templateTwitter.jsp* located at: `/apps/twitterFeed/components/page/templateTwitter/templateTwitter.jsp`.
8. Enter the following JSP code.

```

1  <%@include file="/libs/foundation/global.jsp" %>
2  <cq:include script="/libs/wcm/core/components/init/init.jsp"/>
3  <html>
4  <head>
5  <title>AEM Twitter Feed Page</title>
6  </head>
7  <body>
8  <h2>This page displays twitter feeds</h2>
9  <cq:include path="par" resourceType="foundation/components/parsys" />
10 </body>
11 </html>

```

This line of code:

```
<cq:include path="par" resourceType="foundation/components/parsys" />
```

enables the use of the Adobe CQ paragraph system. An author is able to drag and drop components from the CQ sidekick to the CQ web page. Later in this development article, you drag and drop the Twitter feed component onto a page based on this render component.

This line of code:

```
<cq:include script="/libs/wcm/core/components/init/init.jsp"/>
```

displays the CQ sidekick in the page during design time.

Get the Twitter widget code to use in your AEM application

[To the top](#)

To successfully embed an Adobe CQ widget that displays Twitter data into your AEM application, first you must get the Twitter widget code from Twitter. The Twitter widget code is simply JavaScript code that you can embed into a CQ component. The following JavaScript code represents Twitter widget code.

```

<a class="twitter-timeline" href="https://twitter.com/smac2020" data-
widget-id="359395392489652226">Tweets by @smac2020</a>
<script>!function(d,s,id){var js,fjs=d.getElementsByTagName(s)
[0],p=/^http:/.test(d.location)?'http':'https';if(!d.getElementById(id))
{js=d.createElement(s);js.id=id;js.src=p+"://platform.twitter.com/widgets.js";fjs.pare
(document,"script","twitter-wjs");</script>

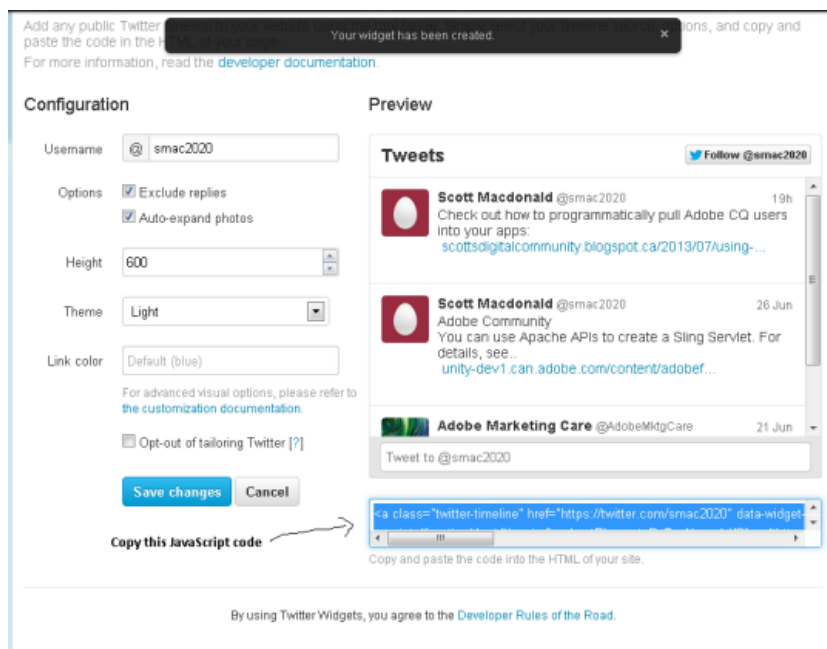
```

To get Twitter widget code, perform these steps:

1. Go to www.twitter.com and login to the account on which to base your component.
2. Go to your settings page.
3. Go to "Widgets" on the left hand side.
4. Click the "Create New" button.
5. Feel free to check "exclude replies" if you do not want replies in results.
6. Click the "Create widget" button.
7. On the right hand side, notice that there is a Preview of your twitter component. Next notice that there is JavaScript code under the Preview section (see the illustration following these steps).
8. Copy the JavaScript code. This will be copied into a CQ component later in this development article.

Note: Replace the Twitter widget code located in this article with your own Twitter widget code.

The following illustration shows the Twitter widget panel and the JavaScript code that you need to copy and paste into an Adobe CQ component.

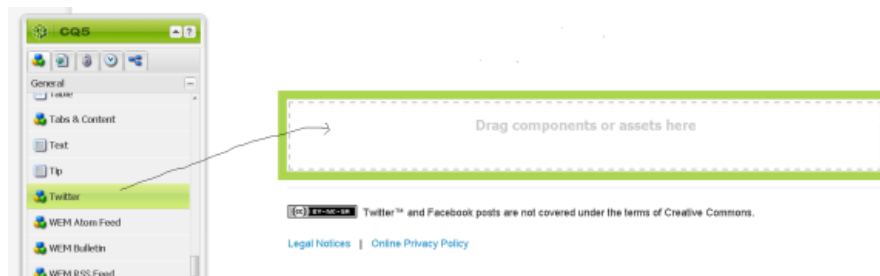


The Twitter Widget view

Create a site that contains a page that displays Twitter data

[To the top](#)

The final task is to create a CQ site that lets an author drag the Twitter feed component onto a page during design time, as shown in the following illustration.



A Twitter feed component located in the CQ sidekick being dragged to a CQ page

Create the Twitter feed component

Earlier in this article, you created the render page component on which a CQ author can drag and drop the Twitter feed component. In this section, you create the Twitter feed component and add it to the CQ sidekick. To create the Twitter feed component, perform these tasks:

1. Right-click `/apps/twitterFeed/component`, then select **New, Component**.
2. Enter the following values:
 - **Label:** Twitter
 - **Title:** Twitter
 - **Description:** Twitter Component
 - **Super Resource Type:** `foundation/components/parbase`
 - **Group:** General (this adds the Twitter feed component to the General category in the CQ sidekick)
 - **Allowed Parents:** `*/parsys`
3. Open the `Twitter.jsp` located at: `/apps/twitterFeed/components/page/Twitter/Twitter.jsp`.
4. Enter the following JSP code. Notice that Twitter widget JavaScript code is included in this CQ component (the Twitter widget code is located right after the `body` tag). Replace this code with the Twitter widget code that you obtained from Twitter.

```
1 <%@include file="/libs/foundation/global.jsp"%>
2 <cq:includeClientLib categories="twittersample" />
```

```

3  <!DOCTYPE html>
4  <html>
5    <head>
6      <title>AEM Twitter Feed Widget</title>
7    </head>
8  <body>
9
10     <a class="twitter-timeline" href="https://twitter.com/smac2020" data-widget-:
11     <script>!function(d,s,id){var js,fjs=d.getElementsByTagName(s)[0],p=/^http:/
12
13   </body>
14   </html>

```

5. Right-click the Twitter component and select New, Dialog. The Dialog wizard creates the initial structure for the component dialog box.

6. Define the following properties on the items parent of the tab1 node. Add this property that defines the widget reference:

- **Name:** xtype
- **Type:** String
- **Value:** = cqinclude

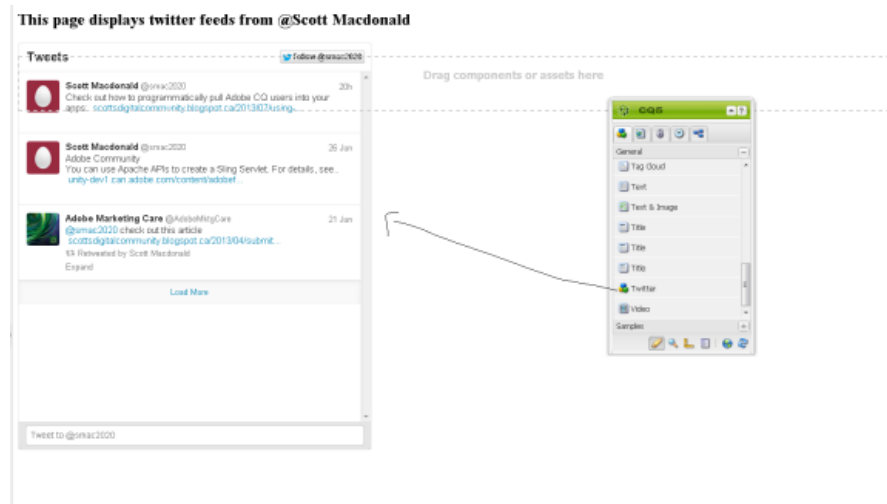
Add the property that defines the path to the reference:

- **Name:** path
- **Type:** String
- **Value:** /libs/foundation/components/image/dialog/items.infinity.json

Note: Delete the tab1 node.

Test the Twitter component

Create a site that contains a page that is based on the templateTwitter (the template created earlier in this development article). You will be able to drag and drop the Twitter component from the CQ sidekick to the page.



The Twitter CQ component being dragged from the CQ sidekick to a CQ page

Create a CQ web page that lets you test the Twitter component:

1. Go to the CQ welcome page at [http://\[host name\]:\[port\]](http://[host name]:[port]); for example, <http://localhost:4502>.
2. Select Websites. (If you are using AEM 5.6, click Tools from the menu on the left.)
3. From the left hand pane, select Websites.
4. Select New Page.
5. Specify the title of the page in the Title field.
6. Specify the name of the page in the Name field.
7. Select templateTwitter from the template list that appears. This value represents the template that is created in this development article. If you do not see it, then repeat the steps in this development article. For example, if you made a typing mistake when entering in path information, the template will not show up in the New Page dialog box.

8. Open the new page that you created by double-clicking it in the right pane. The new page opens in a web browser. You will see the CQ sidekick and the Twitter component under the General category. You can drag and drop the Twitter feed component onto the CQ page.

Note: If the CQ sidekick is empty, you can populate it by clicking the Design button located at the bottom of the sidekick. Next click the Edit button that appears and select General from the list of categories that appear. To view the full sidekick with the General category, click the down arrow icon that appears on the sidekick.



Twitter™ and Facebook posts are not covered under the terms of Creative Commons.

[Legal Notices](#) | [Online Privacy Policy](#)