**Using Data Source in Controller Renderings – Sitecore**

[Data source](https://doc.sitecore.com/developers/sxa/17/sitecore-experience-accelerator/en/data-sources.html) for [Sitecore](https://www.sitecore.com) component provide data points to be used in the components, such as controller renderings or view renderings. Content author can easily edit those data points (e.g. header title, body text or CSS style) to customise components. It is one of the techniques to make Sitecore component customisable. Data source can be assigned to a component.

Another advantage of using data source is that it makes it easier to create AB test or personalised content. Out of the box, Sitecore provides the ability for a component to reference different data source items and swap according to the rules.

In fact, we will create an AB test based on this component in the next post - <a href=" <https://www.mydatahack.com/how-to-set-up-content-tests-with-data-source-sitecore>" target="\_blank">How to Set up Content Tests with Data Source</a>

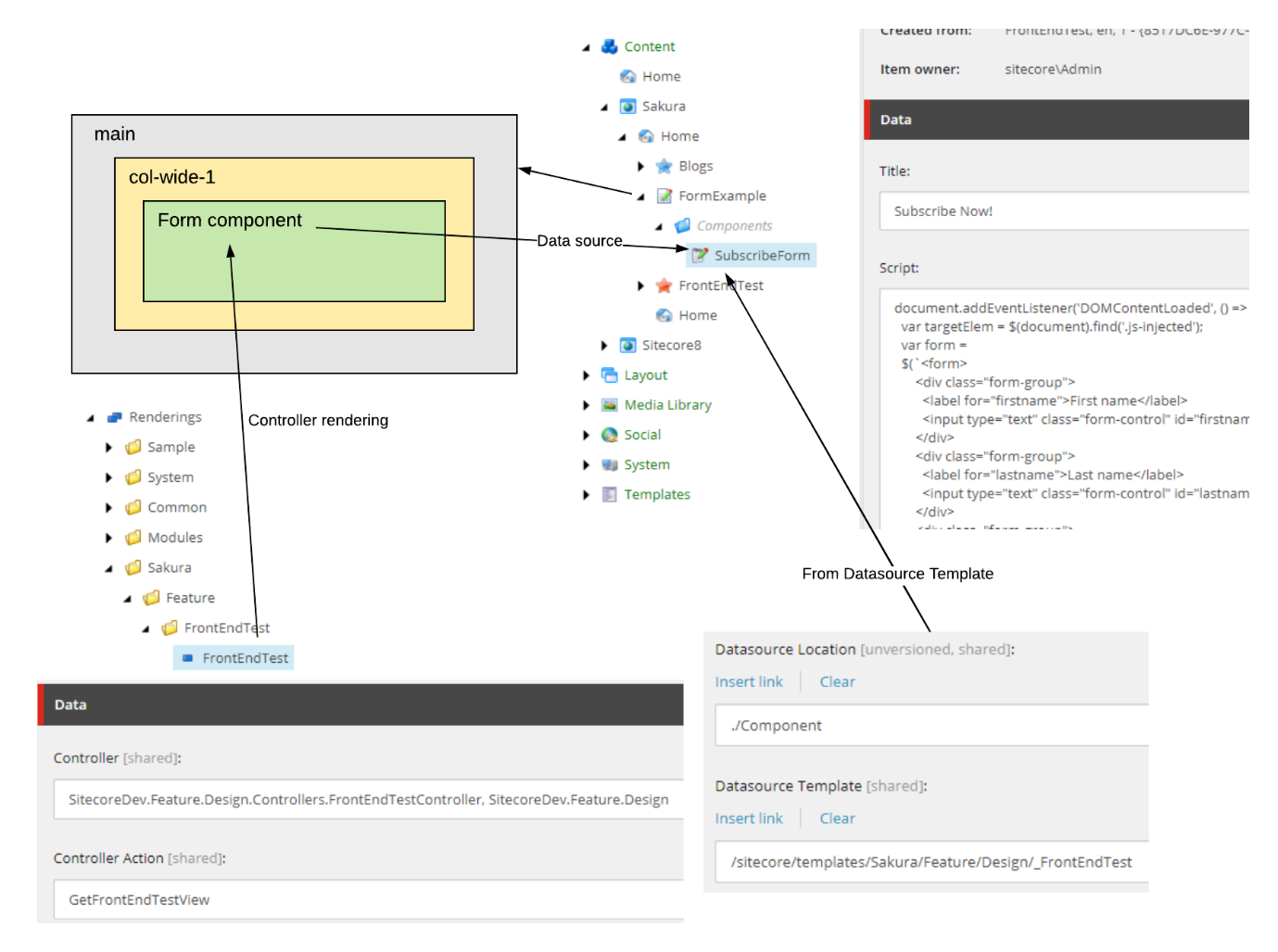
To use data source, the component should reference to a template with or without default values. Actual data points are stored as items. Therefore, the component needs to know the location (folder) of the items.

In this post, we are going to create a controller rendering to render form with data source items. The template for the data source has two fields, Title as single-line text and Script as multi-line text. A form gets injected by writing JavaScript in the multi-line text field.

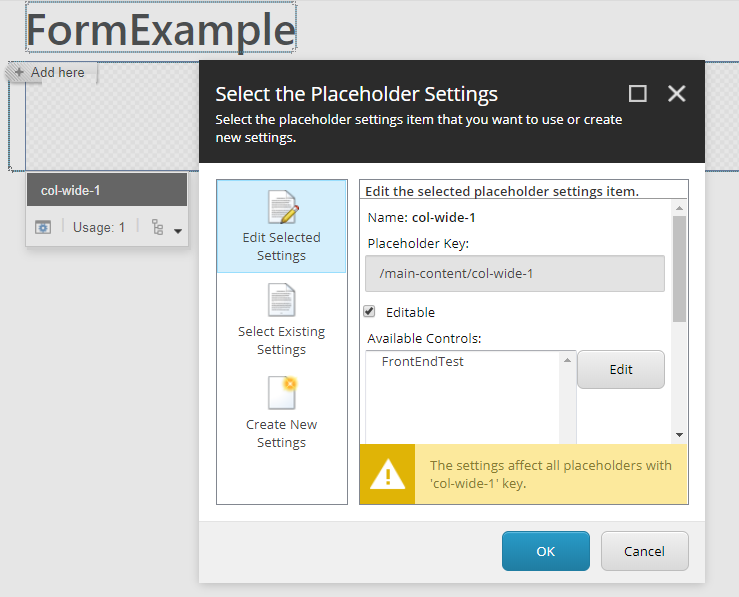
First, we need to create a template for the data source. Then, create a controller rendering that reference data source location and data template.

Once the page is created, it will have a folder template with data source.

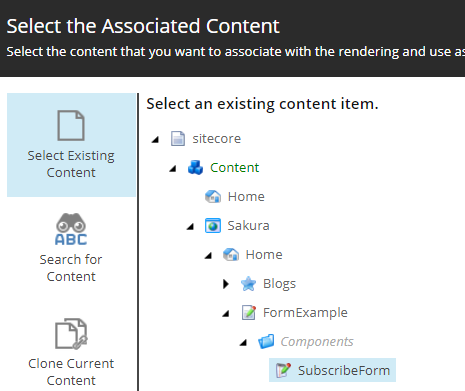
Below is the summary diagram of how everything puts together.



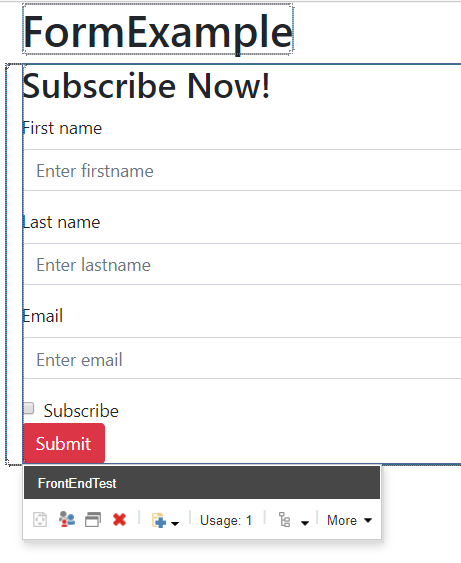
In experience editor, we can add form component to the placeholder, col-wide-1. Make sure that the placeholder includes FrontEndTest as an available control.



As the component has data source, you can select an associated content once you insert the component. To insert associated content, you can click database plus icon on the component.



Form will be rendered.



Let’s check out the code.

**FrontEndTest.cshtml**

This is the razor view.

<script src="https://gist.github.com/mydatahack/42b298fe5a2e277ce883637461abfe6b.js"></script>

**FrontEndTest.cs**

Obviously, we need a model to pass to the view.

<script src="https://gist.github.com/mydatahack/7a6c4c8f7e7eba46b365455ba57f5b52.js"></script>

**FrontEndTestRepository.cs**

Controller gets the data source ID from the current rendering context. Repository access database to retrieve the value based on a data source id.

<script src="https://gist.github.com/mydatahack/dd0683407d9a9f9be2a11848df159a8a.js"></script>

**FrontEndTestController.cs**

This should be fairly straight-forward. RenderingContext.Current.Rendering.DataSource will return the data source Id in string. It will be cast as ID object in the repository to call GetItem().

<script src="https://gist.github.com/mydatahack/2a18e301838b1c775c2ba67817b3e6d2.js"></script>

**Scripts.js**

Finally, this is the script in the multinet text field which injects form into the page.

<script src="https://gist.github.com/mydatahack/a6e2fc852fcd0c192b296c834780c039.js"></script>

<strong><u>Next steps</u></strong>

Let’s use this component to create an AB test. Creating a content test in Sitecore is easy when the component is set up correctly. Check this out: <a href=" <https://www.mydatahack.com/how-to-set-up-content-tests-with-data-source-sitecore>" target="\_blank">How to Set up Content Tests with Data Source</a>.

Before creating the test, we should create the custom goal can be triggerd by AJAX call – check out this post for custom goal creation: : <a href=" <https://www.mydatahack.com/how-to-configure-custom-goals-and-trigger-them-by-ajax-call-sitecore/>" target="\_blank">How to Set up Content Tests with Data Source</a>.