**How to Mock Rendering Parameters in Unit Test – Sitecore**

Now that we created a cool components with rendering parameters (see [Using Rendering Parameters to Inject Data into Hidden Input](https://www.mydatahack.com/using-rendering-parameters-to-inject-data-into-hidden-input-sitecore/)), we can write a unit test. Sitecore’s Rendering class has a publicly accessible string field called Parameters. This is the rendering parameter string get populated on page rendering. Mocking is relatively straight forward as all you need to do is to add the mock parameters to this field.

Here is the controller created in [the previous post](https://www.mydatahack.com/using-rendering-parameters-to-inject-data-into-hidden-input-sitecore/). We will write a unit test on this.

<script src="https://gist.github.com/mydatahack/2c97e1ffb505a3d48929693729ffa009.js"></script>

Here are the unit tests, covering scenarios for parameters present and absent. As you can see, the RenderingParameters class is instantiated with a query string with parameters, then added to the Parameters field in the Rendering object.

All you need to do is to create rendering context with the RenderingContext.EnterContext method and pass the mock rendering.

We are using [NSubstitue](https://nsubstitute.github.io/) to mock the repository through its interface. This should be straight forward. We are also using [FluentAssertions](https://fluentassertions.com) for assertions.

<script src="https://gist.github.com/mydatahack/ad414420ffb5dd8208d69b657b8cd6c0.js"></script>

Here is the repository created in [the previous post](https://www.mydatahack.com/using-rendering-parameters-to-inject-data-into-hidden-input-sitecore/).

<script src="https://gist.github.com/mydatahack/fa7d665605e4fdcf77cb1feb76aa39e1.js"></script>

Unit testing it is super easy. Just create the RenderingParameters object as in the controller test. Then, compare the results.

<script src="https://gist.github.com/mydatahack/6ddf2d433b546f70ddcb9e934774541e.js"></script>