**How to Configure Custom Goals and Trigger them by AJAX Call - Sitecore**

[Sitecore](https://www.sitecore.com/) has out of the box capability to record customers’ behaviours throughout their journey across the channels. We can customise the way to capture customers’ behaviours by registering our own custom events, failure events, goals and outcomes and use them to create personalisation. This out of the box analytics capability is where Sitecore really excels compared to other CMS.

[Goals](https://doc.sitecore.com/users/90/sitecore-experience-platform/en/goals.html) are typically set to a page or an action that is defined as a milestone within the sales funnel. For example, getting customer to a selection page is a milestone in the sales funnel where the end goal is conversion. Getting customer to add an item to a card is another milestone that can be captured as a goal.

They can be used for profiling and personalising content. If the customer visits more pages related to electronic goods as opposed to kitchen utensils, we can categorise them and create personalised experience on the site accordingly.

Another use of goal is to set up AB testing. We can use goal to track say which version of the form triggered submission.

In fact the goal set up here will be used for the next post about setting up content tests (<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>)

In this post, we will create a custom goal that can be triggered on form submit event by AJAX when the customer clicks a form submit button. Here are the things we need to do.

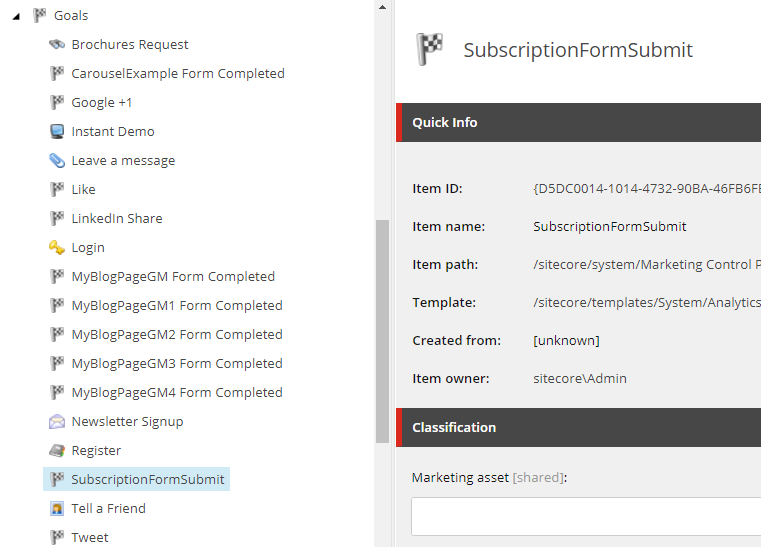
1. Create a new goal item and assign points.
2. Create a service class to register goal.
3. Create an API endpoint that can be used by JavaScript to call AJAX.
4. Create a page with a form.
5. Test the solution.

Let’s get started.

**Steps**

1. **Create a new goal item and assign points**

Go to the marketing control panel and add a new goal called SubscriptionFormSubmit. You can assign points that are associated with this goal. I added 11 points.



1. **Create a service class to register goal.**

In the foundation project under Customer, I created the GoalTriggerService, which takes a goal item ID as an argument. This class can be used wherever you want to register a goal.

Make sure that you have your Tracker enabled. It is a good idea to add VisitorIdentification tag in the layout, too. Otherwise, Sitecore treats every visitor as a bot. Tracker is enabled by default, but you should check the config file just in case. See how this can be done here, [Enabling tacking](https://doc.sitecore.com/developers/91/sitecore-experience-platform/en/enabling-tracking.html).

Here is the code to register goals.

<script src="https://gist.github.com/mydatahack/23855d677a572575df30b54a706523e0.js"></script>

1. **Create an API endpoint to trigger goal from front end**

In Sitecore, we can define a custom route by adding an MVC route, which allows AJAX call to go around the Sitecore render pipeline. This involves creating route map and add some config file. It is a simple thing to do. If you need further information, you should check out the documentation, [User MVC routing](https://doc.sitecore.com/developers/90/sitecore-experience-management/en/use-mvc-routing.html).

In the feature project called Subscription, first add RegisterHttpRoutes.cs in the root.

<script src="https://gist.github.com/mydatahack/dbcb6a53bd5c7224b0036a4bede40909.js"></script>

Now we need to register the custom route by adding the processer to the Initialise pipeline. Create a config file, Feature.Subscription.config in App\_Config/Include folder as below.

<script src="https://gist.github.com/mydatahack/e68ec4d646e6b08ca3b434ba441223d0.js"></script>

The controller is simple. It’s just passing the goal GUID to a GoalTriggerService method created in step 2. The goal can be triggerd by the query string by adding the goal name on the request url. However, registering goals by their GUID is a best practice as adding query parameter with goal name can become flaky.

<script src="https://gist.github.com/mydatahack/fef3a3f417245bd1806139736e8fb532.js"></script>

1. **Create a page with a form.**

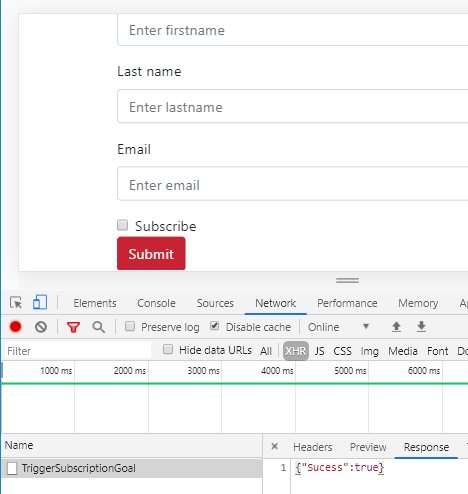
Let’s use the page created in the previous post, [Using Data Source in Controller Renderings](https://www.mydatahack.com/using-data-source-in-controller-renderings-sitecore/). We can update the JavaScript to add a click event listener to the submit button. Yes, it is a little out of ordinary to use a multiline text field to inject JavaScript. The production solution of course should be something a little bit more robust.

<script src="https://gist.github.com/mydatahack/a6cc727fc79785de92abadc342177c4d.js"></script>

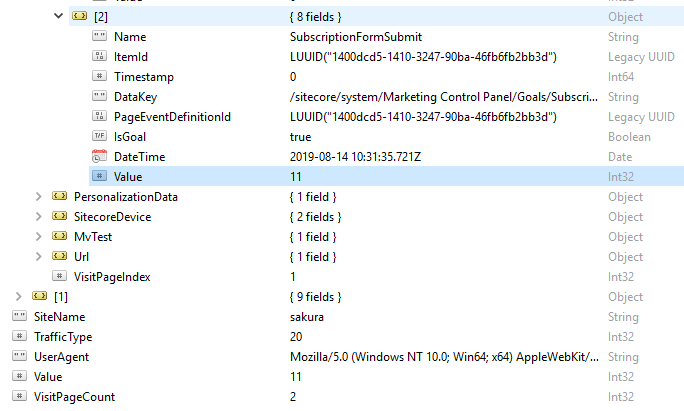
1. **Test the solution**

You cannot really test this outside of the page context as triggering a goal relies heavily on Sitecore Analytics Tracker. If you use curl to hit the endpoint, the goal registration would not be successful.

The best way to test this is to trigger it from the actual page. When you see the network tab, you can see the success message.



You can also check it in MongoDB. In the interaction collection, you can see the goal is triggered with the defined points, 11.



Yay!

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

Let’s use this goal to create an AB test. Creating a content test in Sitecore is easy. Proceed to : <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>)