

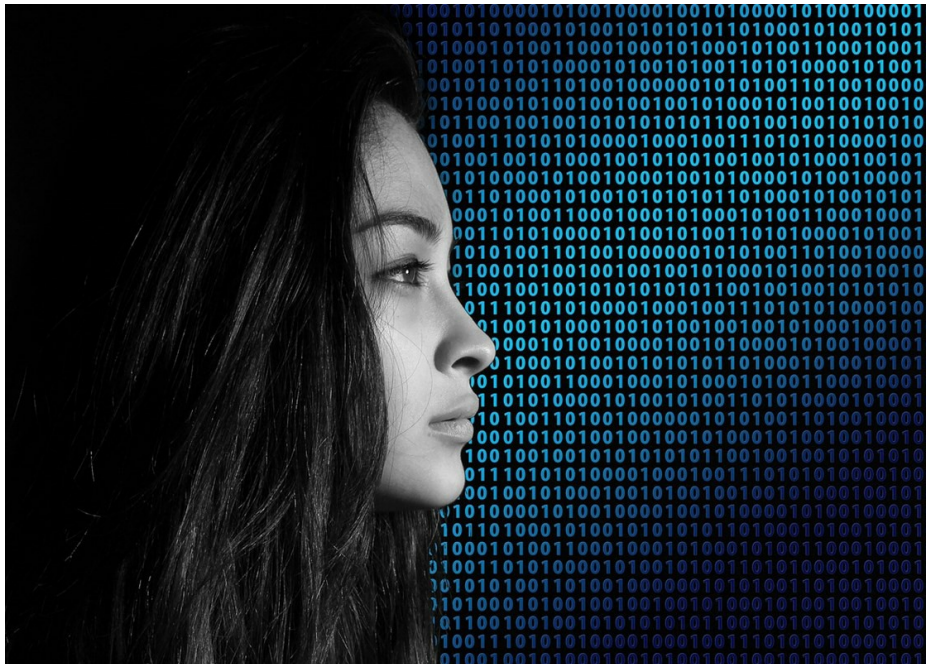
[← Blog posts](#)

Episerver Forms and Pardot Form Handlers

TIPS AND TRICKS INTEGRATIONS CMS OPTIMIZEZY (EPISERVER)



Allan Thraen | 2 Years Ago | PDF |



Episerver comes with a wide range of connectors that allows you to connect Marketing Automation systems to Episerver through multiple integration points. However in some cases you might want to hook directly into the Marketing Automation systems form handler. In the case of SalesForce Pardot it's very easy to do!

One of the most important integration points for any Marketing Automation integration is of course Episerver Forms, where the integration typically allows you to map form fields to profile fields in the Marketing Automation system. That way you can do cool stuff like 'Progressive Profiling' and build up your profiles as they fill in the forms. Or you can start to prefill forms with profile information to make your visitors more likely to sign in through them.

However, sometimes the out-of-the-box connector doesn't do exactly what you had hoped for, and leaves you wanting more. For example - in the case of SalesForce Pardot (and I'm sure also some of the other leading Marketing Automation systems) the connector has no problem in creating a new prospect after a form submission - but it has difficulty handling if the prospect already exist - and will typically end up creating a duplicate profile in the system. At the same time, while it might correctly map and record profile fields like First Name, Last Name and Email it doesn't actually track the "Form submitted event" - which could make it hard later to query & list all prospects who signed up for a webinar, for instance.

Luckily Pardot actually comes with a nice way to handle this: Form handlers. Just as you could create a form in Pardot, you can also create a "Form Handler" which basically can take input from any kind of external form, remap the fields to pardot and perform certain actions upon submission. They can be defined easily in Pardot's UI and after you have defined a form handler you basically get a URL you can post your forms to.

The screenshot shows the configuration page for a Pardot Form Handler. At the top, there is a search bar and the Pardot logo. The main configuration area includes the following fields:

- Name:** AllanTestPardotHandler
- Folder:** /Uncategorized/test-folder-allan
- Campaign:** [Redacted]
- Endpoint URL:** http://go.pardot.com/[Redacted] with radio buttons for **http** and **https**.
- Success Location:** Referring Uri
- Error Location:** Referring Uri
- Tags:** [Empty]
- Created At:** Aug 7, 2019 2:05 PM
- Updated At:** Aug 13, 2019 3:47 PM
- Created By:** [Redacted]
- Updated By:** [Redacted]

Below the configuration fields, there is a section for **Completion Actions** with a button labeled "Add to list" and a dropdown menu showing "Monthly Newsletter".

At the bottom, there is a section for **Form Field Mappings** with a table:

External Field Name	Field
email	Default Field: Email (r)
firstname	Default Field: First Name

So, now all we need to do is essentially make sure that the Episerver Form sends a post request to the Form Handler url with all the submitted data.

"Super", I thought - I'll just make a quick Form Submission Actor that does that if the form contains a secret field with the form handler url. And sure - that was easily done (with only a few quirks around form field name mapping in two places). But then I wondered - do I even need to make a code change? Couldn't I just setup an Azure function, post webhook json data to that, convert it and send it on? Of course. But it can be done even simpler (as I quickly realized). Episerver Forms webhooks will **either** post form data as json **or as a regular form post!**

So - the simply and elegant solution is nothing more than to add your Pardot Form Handler url in the Episerver Form web hooks list - and make sure the Pardot field mappings are accurate (case sensitive).

The screenshot shows a configuration interface for a webhook. It includes a dropdown menu for the trigger, currently set to "Trigger webhook after form submission". Below this is a "Webhook URL" field containing "http://p.pardot.com" followed by a redacted area. To the right, there is a "POST data in JSON format" checkbox which is currently unchecked.

- TIPS AND TRICKS
- INTEGRATIONS
- CMS
- OPTIMIZEZY (EPISERVER)

Post Comments (0)

Salesforce Pardot