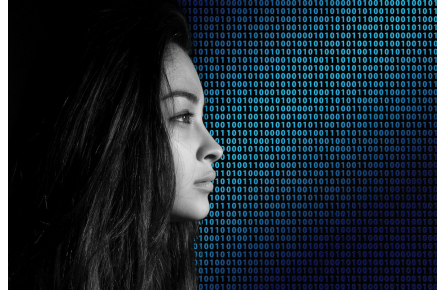


ALLAN THRAEN | 4 years ago | PDF | [Tips and Tricks](#) [Integrations](#) [CMS](#) [Optimizely \(Episerver\)](#)

# EPISERVER FORMS AND PARDOT FORM HANDLERS



**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!**



## Want to get more out of Episerver Forms?

We are experts in optimizing and extending Episerver forms. Reach out and let's talk.

[REACH OUT](#)

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.

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).

[Tips and Tricks](#) [Integrations](#) [CMS](#) [Optimizely \(Episerver\)](#)

**CodeArt ApS**

Teknikerbyen 5, 2830 Virum, Denmark

**Email:** info@codeart.dk

**Phone:** +45 26 13 66 96

**CVR:** 39680688

