Using FormAssembly's Publishing API with Salesforce

Download the PDF of this article.

In this Article

Related Articles

Introduction

The FormAssembly Publishing API allows FormAssembly users to embed a fully functional form on their site which allows form interactions to appear as if they are taking place within the user's own domain.

The Publishing API can be used within Salesforce's VisualForce system for the same effect, either internally to Salesforce organization, or externally using a Portal. The steps below outline the process for completing this integration within Salesforce.

> **Note:** You must have knowledge of VisualForce to complete the steps below. **We do not provide troubleshooting support for VisualForce.** The guide below is meant to help you through this process, but all VisualForce coding must be done on your end.

Add FormAssembly as a Remote Site to Salesforce

To make a "callout" or "remote access call" to FormAssembly from VisualForce code, you must first add FormAssembly's two domains to your Salesforce organization's approved remote sites. Otherwise, your organization's VisualForce code will refuse to connect to FormAssembly.

You can do this by going to Salesforce to **Setup → Security Controls → Remote Site** and adding:

```
http://app.formassembly.com
https://app.formassembly.com
```

For more details, please see this Salesforce doc: Apex Web Services and Callouts

---

Create Your Apex Class

In Salesforce, go to **Setup App Setup → Develop → Apex Classes** and select **New**. Give the class the name

`FormAssemblyFormController`, and paste the following code into the interface:

```
public class FormAssemblyForm {
    public HTTPResponse res {get; set;}
    public String resBody {get; set;}
    public String endpoint {get; set;}

    public FormAssemblyForm() {
        // Grab the query string from the url
        String queryString = getQueryString();
        PageReference pageRef = ApexPages.currentPage();
        HttpRequest req = new HttpRequest();
        req.setMethod('GET');

        if(pageRef.getParameters().get('tfa_next') ==null){
            // Replace 'FORM_ID' with your form's ID number
            endpoint = 'https://app.formassembly.com/rest/forms/view/123457?' + queryString;
        } else {
            endpoint = 'https://app.formassembly.net/rest' + pageRef.getParameters().get('tfa_next');
        }

        req.setEndpoint(endpoint);
        Http http = new Http();

        try {
            // Execute web service call here
            res = http.send(req);
            resBody = res.getBody();
        } catch(System.CalloutException e) {
            // Exception handling goes here....
            System.debug(e);
        }
    }

    /**
     * Grabs the query string from the url of the current page
     * @returns String queryString
     */
    private String getQueryString() {
        String queryString = '';

        Map<String, String> parameters = ApexPages.currentPage().getParameters();
        Set<String> keys = parameters.keySet();

        for (String k: keys) {
            queryString += k + '=';
            queryString += parameters.get(k) + '&';
        }

    return queryString;
    }
}
```

Note that in the above code, the value `FORM_ID` should be replaced by the form ID for your FormAssembly form, such as `123456` as retrieved from your form's **Publish** tab inside FormAssembly.

Create Your Salesforce Page

In Salesforce, go to **Setup > App Setup > Develop > Pages** and select **New**. Give the page the name `FormAssemblyPage`, and paste the following code into the interface:

```
<apex:page controller="FormAssemblyFormController">
  <apex:outputText value="{!resBody}" escape="false" />
</apex:page>
```

Save, and your form will now be available at the location:

```
https://xxxx.salesforce.com/apex/FormAssemblyPage
```

It will also keep users on the same page location as they complete the form or save & resume responses.