Prefill Lookups Using Secure Parameters

Download the PDF of this article.

In this Article

Related Articles

Overview

You can use secure parameters with our Salesforce Prefill Connectors to ensure the security of your prefilled data. This feature is currently available for Essentials plans and above.

Note: We recommend testing with the unsafe query parameter before moving on to the secure parameter.

Secure prefilling is done by adding a "signature" to the end of your prefilled link. To learn more about prefilled links, check out this help document. You create this signature in one of two ways:

1. You can create a signature inside your Salesforce Prefill Connector and manually add it to your prefilled link.
2. You can dynamically create the signature inside of Salesforce and then add it to the prefilled link using Apex code.

The first option can be done entirely inside FormAssembly without any custom code. However, you can only sign a single set of search values with this method. This means that your prefilled link would not be able to change based on the respondent.

If you would like your prefilled link to be able to change for each respondent, you will need to dynamically create the signature with the second option inside Salesforce. This requires some technical knowledge and familiarity with custom Apex coding & Visualforce, and **our Support Team is unable to assist with writing and editing custom code.**

Requirements

- The Salesforce and ExactTarget Prefill Connectors only allow query-string parameters that are authenticated with a base64-encoded HMAC-SHA256 signature.
- If your query requires a dynamic parameter (i.e., a parameter that is not known at setup time, and changes for each respondent), you can pass the desired value to the form by appending the parameter to the URL. Please note that we **do not provide support for dynamically signed secure parameters**.

- **Note that the parameter must be signed** to prevent a respondent from arbitrarily changing the parameter and accessing other records in your database.

Static (unchanging) Secure Parameters

As noted above, you have an option if you'd like to use secure parameters but do not know how to write custom code. If your prefilled link will remain the same for every respondent, this is a great option!

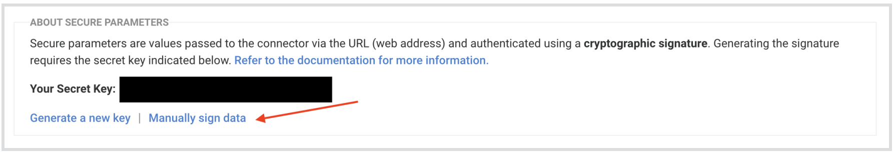
1. Obtain the Shared Secret Key

This key is set on a form-by-form basis, and displayed in the setup screen of the connector. If no key has been set yet, one can be generated automatically by pressing the **generate key** button. Take note of the value of the secret key, you'll need it to generate the signature.

Keep this key safe, as making it public defeats the purpose of securing your query parameters. If you believe that your secret key has been compromised, you can generate a new one at any time.

2. Generate the Signature

FormAssembly offers a quick tool to generate a signature if the parameters to be signed are already known (click "**Manually Sign Data**").



ABOUT SECURE PARAMETERS

Secure parameters are values passed to the connector via the URL (web address) and authenticated using a **cryptographic signature**. Generating the signature requires the secret key indicated below. Refer to the documentation for more information.

Your Secret Key: [redacted]

Generate a new key | Manually sign data

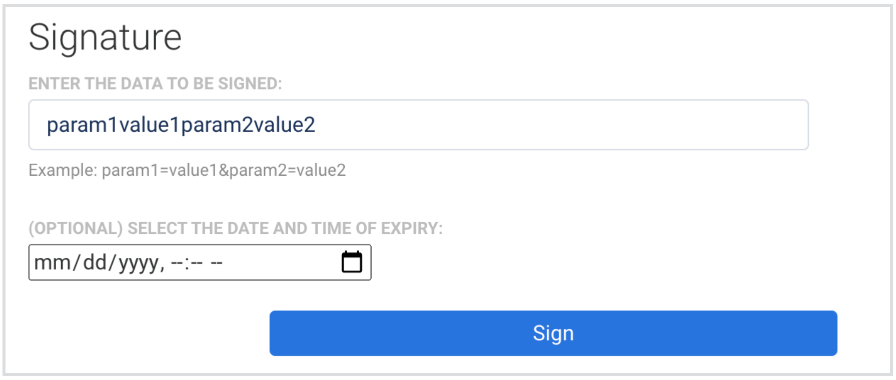
To prepare your data for signature, you must follow this convention:

- Include all parameters present in the query string, in the order they appear.
- Include the parameter names and values, but remove the `&` and `=` separators.
- Do not include any other part of the URL, e.g., domain, path, etc.
- Sign plain text only (not URL-encoded).

Example:

For this query string: `param1=value1&m2=value2`

The data to sign is: `param1value1param2value2`



Signature

ENTER THE DATA TO BE SIGNED:

param1value1param2value2

Example: param1=value1&param2=value2

(OPTIONAL) SELECT THE DATE AND TIME OF EXPIRY:

mm/dd/yyyy, --:-- --

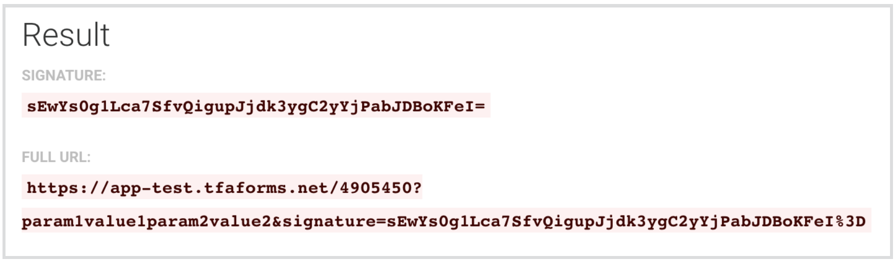
3. Add an Optional Expiry Parameter

You can also set your secure prefilling links to expire at a certain point in time by adding an expiration timestamp to the prefilling link. This optional feature helps to improve the security of prefilling by preventing links from being reused if found in a browser history or old email.

4. Click "Sign"

5. Retrieve the Prefilled Link

You will now see a "Result" section appear. This section will contain your newly generated prefilled link. You can use this link to prefill your form with the parameters and values that you have signed. This way, your respondents won't be able to change the values or parameters to try and prefill the form differently. Only these parameters and values will work with this signature.



Result

SIGNATURE:

```
sEwYs0g1Lca7SfvQigupJjdk3ygC2yYjPabJDBoKFeI=
```

FULL URL:

```
https://app-test.tfaforms.net/4905450?
param1value1param2value2&signature=sEwYs0g1Lca7SfvQigupJjdk3ygC2yYjPabJDBoKFeI%3D
```

Dynamic Secure Parameters

While static parameters are useful when sending out one link to all of your respondents, it's more likely that your parameter values change dynamically. In this case, you must compute the signature automatically using the standard HMAC-SHA256 algorithm and the secret key obtained from your connector. The resulting signature must then be base64- and URL-encoded.

Note: FormAssembly does not provide support for dynamically signing secure parameters.

This process must be done inside Salesforce. For example, you could use Apex code and VisualForce pages to open a prefilled link with secure parameters when a button is clicked on a record. You could also send out emails using a process in Salesforce which would then utilize Apex code to create a prefilled link with secure parameters and add it to the email.

Because this needs to occur inside of Salesforce, and every single instance of Salesforce is different, we're not able to provide examples for every single use case. Instead, we've provided some basic Apex code to get started with below.

Apex Code Sample for Visualforce:

```
// encoding this query string: recordid=9876
String data = 'recordid9876'; // strip '&' and '=' from string.
Blob mac = Crypto.generateMac('HMacSHA256', Blob.valueOf(data), Blob.valueOf('secret_ke
y'));
String sig = EncodingUtil.urlEncode(EncodingUtil.base64Encode(mac), 'UTF-8');
String url = 'https://tfaforms.com/12345?recordid=9876&signature=' + sig;
```

Apex Code Sample with Optional Expire Parameter

When working with an expire parameter, there are two important things to keep in mind:

1. **The expire parameter is optional.** If it is set, it must be included in the list of parameters signed with HMAC.

2. The expire value must be a unix timestamp, set in the future. The timestamp indicates when the link should expire. Depending on your use case, it could be within a few minutes, hours or days.

```
// encoding this query string: recordid=9876
String data = 'recordid9876'; // strip '&' and '=' from string.
// current timestamp
Integer time = DateTime.now().getTime() / 1000;
// add 5 minutes (300 seconds)
time = time + 300;
// add the expiration parameter.
data = data+"expire"+String.valueof(time);
// sign
Blob mac = Crypto.generateMac('HMacSHA256', Blob.valueOf(data), Blob.valueOf('secret_ke
y'));
String sig = EncodingUtil.urlEncode(EncodingUtil.base64Encode(mac), 'UTF-8');
String url = 'https://tfaforms.com/12345?recordid=9876&signature=' + sig;
```

Append the Parameter(s) and the Signature to the URL

See more on building a prefilling link.

Example:

Let's say your form is hosted at:

```
https://tfaforms.com/12345
```

You have the value `9876` you want to send as the parameter `acctnum` of the record that you want to use to prefill your form. To pass these values to the form, you modify the address like this:

```
https://tfaforms.com/12345?acctnum=9876
```

To add the signature, compute its correct value and append it to the URL as the `signature` parameter.

```
https://tfaforms.com/12345?acctnum=9876&signature=g%2BRg6RkStnFYysLJKcgaWxhqwYH
SJc%2BuY%3D
```

Note:

- The example provided above does not include the optional expiry parameter. If using the optional expiry parameter, you will need to append it to your URL.
- It is not possible to prefill by simply referencing the signature. The prefilled link must also include the parameter(s) -- "acctnum" in the example above. Without the parameter(s), prefilling will not be successful.