Wirecard CEE Integration Documentation

() -

Created: 2019-11-20 21:14

PCI DSS SAQ A compliance for Wirecard Checkout Seamless

As of January 1st, 2018, PCI DSS v3.2.1 is mandatory for all e-commerce merchants and defines some new and stringent requirements for the handling of sensitive cardholder data in your online shop. Following this standard, a merchant's web site(s) is not allowed to itself handle sensitive credit card specific data.

For merchants, any efforts to become compliant with the new standard PCI DSS v3.2.1 and the corresponding Self-Assessment Questionnaire SAQ A-EP applicable for Wirecard Checkout Seamless, which might be required by your acquirer, will be rather costly and time-consuming.

In this respect, Wirecard, as PCI DSS certified third-party payment processor, has developed the “PCI DSS SAQ A Compliance” feature for merchants who use Wirecard Checkout Seamless but are not themselves PCI-certified according to PCI DSS v3.2.1 based on SAQ A-EP.

In other words, as a merchant or integrator you may decide whether you integrate Wirecard Checkout Seamless **without** the “PCI DSS SAQ A Compliance” feature requiring PCI DSS v3.2.1 certification and compliance with SAQ A-EP, or **with** the “PCI DSS SAQ A Compliance” feature which was developed especially for Wirecard Checkout Seamless and ensures compliance with PCI DSS v3.2.1 and is based on the less stringent SAQ A.

Please note that when using “PCI DSS SAQ A Compliance” you will of course still be able to enjoy all features and benefits of Wirecard Checkout Seamless without any restrictions.

For more information and details read PCI DSS v3.2.1 and SAQ A-EP.

Implementation

When using Wirecard Checkout Seamless, sensitive payment data are typically entered by your consumers in your HTML forms in your online shop and directly transferred to Wirecard via JavaScript to be stored in the Wirecard data storage. Although sensitive data are never transferred to your online shop, compliance with the new PCI DSS v3.2.1 and corresponding SAQ A-EP is required.

For “PCI DSS SAQ A Compliance” credit card relevant data are no longer transferred to Wirecard via JavaScript but are entered directly in a web form delivered by Wirecard. For this purpose an iframe is displayed containing the relevant credit card input fields where the content of this iframe is directly delivered by Wirecard.

Once your consumer has entered the financial data into the web form within the iframe, without noticing that it is actually hosted by Wirecard, he is redirected to your online shop. The whole process is designed seamlessly and allows for customization of the relevant iframe input fields via CSS.

Parameters to use

When initializing the Wirecard data storage for the feature “PCI DSS SAQ A Compliance” the following **request parameter**, which is optional for conventional Wirecard Checkout Seamless, **becomes required** to allow activation of the “PCI DSS SAQ A Compliance” feature:

Parameter	Data type	Short description
javascriptScriptVersion	Alphanumeric with a fixed length of 4.	The value `pci3` must be set for this parameter to allow the input of credit card related data in the displayed iframe.

The following **request parameters are optional parameters** used to display or hide credit card related input fields for payment methods credit card, credit card - mail order and telephone order as well as for Maestro SecureCode:

Parameter	Data type	Short description
iframeCssUrl	URL	URL to a CSS file on your server to perform customizations of the iframe input fields.
creditcardPanPlaceholder	Alphanumeric with a variable length of 32.	Placeholder text for the credit card number field.
creditcardCvcPlaceholder	Alphanumeric with a variable length of 32.	Placeholder text for the credit card verification number field.
creditcardCardholderNamePlaceholder	Alphanumeric with a variable length of 32.	Placeholder text for the credit card holder field.
creditcardIssueNumberPlaceholder	Alphanumeric with a variable length of 32.	Placeholder text for the credit card issue number.
creditcardShowExpirationDatePlaceholder	Boolean	Display placeholder text for the credit card expiration date.
creditcardShowIssueDatePlaceholder	Boolean	Display placeholder text for the credit card issue date.
creditcardShowCardholderNameField	Boolean	Display field containing card holder name.
creditcardShowCvcField	Boolean	Display CVC field.
creditcardShowIssueDateField	Boolean	Display field containing the card issue date.
creditcardShowIssueNumberField	Boolean	Display field containing the card issue number.

Please note that these fields must not be included in the fingerprint calculation.

Iframe

As regards the storing of payment data in the Wirecard data storage when using “PCI DSS SAQ A Compliance”, the credit card specific input fields are displayed in an iframe which is hosted by Wirecard. In order to show this iframe, the Wirecard data storage provides a specific function which is used to load the iframe into a certain section of your HTML page:

For Credit Card

```
<div id="creditcardDataIframe"></div>

<script type="text/javascript">
  var wd = new WirecardCEE_DataStorage();
  wd.buildIframeCreditCard('creditcardDataIframe', '100%', '250px');
</script>
```

For Credit Card - Mail Order and Telephone Order

```
<div id="creditcardMotoDataIframe"></div>

<script type="text/javascript">
  var wd = new WirecardCEE_DataStorage();
  wd.buildIframeCreditCardMoto('creditcardMotoDataIframe', '100%', '250px');
</script>
```

For Maestro SecureCode

```
<div id="MaestroDataIframe"></div>

<script type="text/javascript">
  var wd = new WirecardCEE_DataStorage();
  wd.buildIframeMaestro('MaestroDataIframe', '100%', '250px');
</script>
```

The following 3 parameters may be set within the `buildIframe*` function:

Function	Parameter	Description
buildIframeCreditCard buildIframeCreditCardMoto buildIframeMaestro	Parameter 1	ID of HTML element into which the iframe will be loaded.
	Parameter 2	Defines iframe width.
	Parameter 3	Defines iframe height.

Verification

In order to immediately verify the correctness of entered credit card data and to proceed to the next step of the payment process, use for

Credit Card

```
dataStorage.storeCreditCardInformation(null, callbackFunction);
```

Credit Card - Mail Order and Telephone Order

```
dataStorage.storeCreditCardMotoInformation(null, callbackFunction);
```

Maestro SecureCode

```
dataStorage.storeMaestroInformation(null, callbackFunction);
```

Note that `null` is used as first parameter while `callbackFunction` is a JavaScript function which allows to handle the result of the storage operation.

However, this function is optional since credit card specific data are already stored during input when the consumer changes from one field to another while entering the payment data.

Please note that a result is only returned, if the consumer's browser supports *postMessages*. Otherwise, the returned value is `null` and a `readDataStorage` operation may be performed to verify the correctness of the payment data.

Browser minimum versions that support *postMessage* are Internet Explorer v8.0, Mozilla Firefox v3.0, Google Chrome v1.0, Safari v4.0, Opera 9.5; older Android versions, e.g. v 2.3, might not support *postMessage*.

Customization via CSS

Wirecard ensures a seamless integration of “PCI DSS SAQ A Compliance” and the possibility to customize the iframe input fields according to your needs by using the parameter `iframeCssUrl` (URL to a CSS file on your server to perform customizations of the iframe input fields).

To ensure that all elements within the checkout page are hosted by Wirecard systems and in order to provide for SAQ A compliance, this CSS file is received via your given URL in the request parameter `iframeCssUrl` from your server to the Wirecard CEE server. After successful validation by Wirecard, the CSS file is subsequently loaded by Wirecard (during each data storage init), parsed and delivered as inline CSS file within the iframe, i.e. the CSS file is not loaded from your server during presentation in the browser of your consumer.

Our system uses a URL-based caching mechanism for this `iframeCssUrl`. If you wish to disable this mechanism, please append the current timestamp as GET parameter, e.g. `https://www.servername.com/iframe.css?23472304897234789`.

Please note that due to security reasons neither url() functions nor @import directives (e.g. external fonts) are allowed in the CSS file. If they are used they will be ignored.
Never send any font changes within the CSS file. Security checks may last several minutes and lead to payment cancelation.

Definition of terms

We invite you to visit Securing your online shop for an outline of the basic concepts and relevant terminology used in this page and to consult our Glossary.