

## Wirecard CEE Integration Documentation



**Created: 2019-12-08 19:29**

# Initialization of Wirecard Data Storage

## Definition

The Wirecard data storage is used for storing sensitive payment data of your consumer during the payment process of your online shop. By using the functionality (based on JavaScript) you are able to retrieve, send and store e.g. credit card numbers of your consumer without using your web server. This ensures that your online shop and your web server do not need to be PCI-compliant.

## Time of initialization

You can initialize the Wirecard data storage at the beginning of the payment process in your online shop. After initialization, the data storage session for a specific consumer is valid for 30 minutes after the last access from your online shop to the data storage. After 30 minutes the session becomes invalid and a new data storage has to be initialized.

The Wirecard data storage is only required for the following payment methods which manage sensitive payment data of your consumer:

- Credit Card
- Credit Card - Mail Order and Telephone Order
- Maestro SecureCode
- SEPA Direct Debit
- paybox
- giropay

For all other supported payment methods you do not need to store data in the data storage before starting the payment process itself.

## Initializing the Wirecard data storage

To initialize the Wirecard data storage, send a server-to-server request from your web server to a specific URL at the Wirecard Checkout Server with specific parameters in the POST data.

The URL for the server-to-server initialization is:

`https://checkout.wirecard.com/seamless/dataStorage/init`

Please be aware that it is sometimes necessary to enable server-to-server requests in the

configuration of your web server. This issue arises typically on provider managed web servers with PHP.

Please also configure your firewall settings for sending data from your server to checkout.wirecard.com (185.60.56.34:443).

For a proper request you have to set a correct HTTP header. Therefore you need to set the following HTTP header elements within your request:

HTTP header parameter	Description
Host	Domain name of server. Has to be set to the following value: checkout.wirecard.com
User-Agent	User agent string of client.
Content-Type	MIME type of the body. Has to be set to the following value: application/x-www-form-urlencoded
Content-Length	Length of body in bytes.
Connection	Type of connection. Has to be set to the following value: close

Please be aware that an incorrect setting of the header parameters results in an HTTP 403 error message of the Wirecard Checkout Server.

## Computing the fingerprint

The fingerprint is computed by concatenating all request parameters without any dividers in between and using the secret as cryptographic key for the hashing function. If you do not use the optional parameters shopId and javascriptScriptVersion you have to omit them in your fingerprint string.

Please be aware that the concatenation of the request parameters and the secret has to be done in the following order:

1. customerId
2. shopId
3. orderId
4. returnUrl
5. language
6. javascriptScriptVersion
7. secret

After concatenating all values to a single string create an HMAC-SHA-512 hash with your secret as cryptographic key. The result is the fingerprint which you add as a request parameter to the server-to-server call.

The Wirecard Checkout Server is thus able to check whether the received parameters are manipulated by a 3rd party. Therefore it is essential to keep your secret safe!

## Required request parameters

To initialize the Wirecard data storage you have to set all required parameters to their corresponding values you need within your online shop. If one or more of these required parameters are missing you will get an error message.

Parameter	Data type	Short description
customerId	Alphanumeric with a fixed length of 7.	Unique ID of merchant.
orderId	Alphanumeric	Unique reference to the order of your consumer.
returnUrl	Alphanumeric	Return URL for outdated browsers.
language	Alphabetic with a fixed length of 2.	Language for returned texts and error messages.
requestFingerprint	Alphanumeric with a fixed length of 128.	Computed fingerprint of the parameter values and the secret.

### returnUrl

The parameter `returnUrl` is used for browsers who are not capable of fully supporting CORS (Cross Origin Resource Sharing). In that case the communication between the HTML page and the Wirecard Checkout Server will be done within an iframe where the anonymized payment data are returned to JavaScript objects. This return URL is called by the browser of your consumer. You can find an example for the return page written in PHP within the example code.

## Optional request parameters

Parameter	Data type	Short description
shopId	Alphanumeric with a variable length of 16.	Unique ID of your online shop.
javascriptScriptVersion	Alphanumeric	Version number of JavaScript.

### javascriptScriptVersion

This parameter defines the version of the used JavaScript for managing the JavaScript based communication between your HTML page and the Wirecard data storage. If this parameter is not set we will deliver the JavaScript script version that is defined within your setup. You only need to use this parameter if you have at least two online shops which require different versions of the JavaScript script version. Our support teams will inform you if this is necessary.

## Format of return values

After you send the data storage initiation request as a server-to-server request from your web server to the Wirecard Checkout Server you will get the result of the initiation as key-value pairs returned in the content of the response.

## Returned response parameters

For a successful initialization of the Wirecard data storage you will get the following parameters returned:

Parameter	Data type	Short description
storageId	Alphanumeric with a fixed length of 32.	Unique reference of the data storage for a consumer.
javascriptUrl	Alphanumeric	URL to a JavaScript resource which have to be included for using the storage operations of the data storage.

For example, a successful initiation of the Wirecard data storage would return:

```
storageId=73171af0b8990b9ef2d11b2070f54ad3&javascriptUrl=https://checkout.wirecard.com/seamless/dataStorage/js/D200001/qmore/73171af0b8990b9ef2d11b2070f54ad3/dataStorage.js
```

If the initialization did not succeed you will get parameters describing the error:

Parameter	Data type	Short description
errors	Numeric	Number of errors occurred.
error.{n}.errorCode	Numeric with a fixed length of 5.	Numeric error code which you should log for later use.
error.{n}.message	Alphanumeric with special characters.	Error message in English.
error.{n}.consumerMessage	Alphanumeric with special characters.	Error message in localized language for your consumer.

For further details, see [Error Codes](#).

For example a possible error would look like:

```
error.1.errorCode=11500&error.1.message=CUSTOMERID+is+missing.&error.2.errorCode=11009&error.2.message=Language+is+missing.&error.3.errorCode=15300&error.3.message=ORDERIDENT+has+an+invalid+length.&error.4.errorCode=11301&error.4.message=RETURNURL+is+missing.&error.5.errorCode=11506&error.5.message=REQUIREFINGERPRINT+is+missing.&errors=5
```

Please be aware that we are not able to return a consumerMessage if your configuration is not valid.

## Next step

- Storing sensitive payment data in Wirecard data storage