Third-party-cookies - Browser cookies and session handling

Introduction and caused issue

Current web browsers are more and more going to block so called third party cookies to increase privacy of the internet user. However, a lot of shop implementations rely on a session handling where the sessionId is stored in such a cookie.

By blocking these cookies the merchant's shop looses the information (e.g. SessionId) when the consumer has been redirected to the payment pages and is returning back to the shop after the payment has been completed.

Possible solutions

parameter "Custom"

You can use the parameter "Custom" to pass any customized parameter (like sessionId or more) to and returns your "Custom"-values when consumer returns to your shop.

The parameter "Custom" is not encrypted. Several parameters can be concatenated separated by "|" in the request and are returned by "&" for easy handling in the response.

Sample for request: `Custom=sessionId=123|customerId=456`

Sample for response: `sessionId=123&customerId=456`

Additional redirect after consumer returns in your shop

After a successful payment the consumer is redirected to the URL "URLSuccess" that you provided in the payment request.

With the first redirect the web browser ignores the stored cookie – because that redirect was initiated by a third party – and the sessionId is lost.

Once you initiate a second redirect within your shop just after the consumer has been redirected the cookie will be loaded – because this redirect has been initiated by the original site.

Changing the cookie definition

Upgrading the cookie definition to explicitly allow third-party-cookies. Please consider browser compatibility when using this option.

A cookie is normally created with this information:

```
Set-Cookie: sessionId=<your-sessionId>; Domain=<your-domain>; Path=/; HttpOnly; Secure
```

Add the attribute `Secure; SameSite=None` (`SameSite=None` is only working together with `Secure`) when creating the cookie containing your sessionId:

```
Set-Cookie: sessionId=<your-sessionId>; Domain=<your-domain>; Path=/; HttpOnly; Secure; SameSite=None
```

So, please ensure that these attributes are set, meaning:

Attribute	Description
sessionId	Key and value you would like to store within the cookie, e.g. sessionId, sessionid, id, SESSIONID, ...
Domain	Best practice: Ensures that the web browser will only read cookie values stored by this domain (e.g. shop.merchant.com)
Path	Best practice: This path must exist in the URL – otherwise the browser won't send the cookie
HttpOnly	Best practice: Ensures that JavaScript can not access the cookie
Secure	Best practice: The cookie will only be sent to the server when request is done via https – ensuring that confidential information is sent unencrypted via http.

SameSite	**New:** This attribute disables the third-party-cookie blocking so the information will be available after the consumer returns to your shop. Please note that this attributes only works if `Secure` is used, too.

Affected implementations

- Credit card payment form "paySSL.aspx"
- Hosted payment page "paymentPage.aspx"