

Use of the xID and TINFO client in an in-app browser

Some mobile applications, like Facebook, might use their own in-app browser to display web pages when a user clicks a link from within the app. This in-app browser can have limitations that a stand-alone mobile browser application might not have.

A known scenario occurs when a merchant wants to initiate an xID authentication in a popup window. An in-app browser may not open popups in a new window context, e.g. as a new tab. Instead, it may redirect to the popup's URL, away from the merchant initiating the authentication. When the xID authentication and potential consent handling via TINFO has finished, the merchant callback page, returned from the `redirect_uri` endpoint, may try to send a message back to the parent window to let it know the authentication has finished. If the in-app browser redirected to the authentication, this will fail, since the current window has no parent window.

To solve this, the merchant needs to support a redirect type login. One way to do this, is to check, on the callback page returned from the `redirect_uri`, whether the current window context equals the parent window context. If they are equal, there is no parent window, and the merchant can assume it needs to redirect the user to the page it wants to display after the xID/TINFO dialog.

Here is an example of this check in JavaScript:

```
var windowParent = window.opener || window.parent;
if (windowParent === window) {
  // The current window has no parent window
} else {
  // The current window has a parent window
}
```

If the current window has no parent, `window.opener` will be null. Also, `window.parent` will be a reference to the current window itself, since it has no parent. This means that if the current window has no parent, it will be equal to the `window.parent` property.

If the browser initiated the xID/TINFO dialog as a popup, the current window would not be equal to the `window.parent` property.