SOP WEB SECURITY

Deliverable for this lab: briefly explain the following:

- ***explain the error you get in exercise 1.1***
- ***explain what you had to do to fix the problem of exercise 1.2***

1.1 SOP in the DOM

This exercise shows a practical example of how the DOM of another origin cannot be accessed.

Create a HTML page that loads 'https://www.orange.be' in an iframe, and that tries to read the content of the iframe. For example:

```
<!DOCTYPE html>
<html>
<body>

<h2>SOP example 2</h2>

<iframe id="frame" src=" https://www.orange.be/"></iframe>

<script>
    window.onload = function() {
        setTimeout(function(){
            document.getElementById(frame).contentWindow.innerHTML;
        },5000)
    };
</script>

</body>
</html>
```

Deploy this HTML page on Netlify and browse to it.

Open developer tools. Re-issue the following command in the console of the developer tools: "document.getElementById('frame').contentWindow.innerHTML;" Understand the error that is thrown.

1.2 SOP in an XMLHTTPRequest and CORS

This exercise shows one practical implication of the Same Origin Policy when using the XMLHttpRequest API..

Create a HTML file containing the following contents:

```
<!DOCTYPE html>
<html>
<body>

<h2>Using the XMLHttpRequest Object</h2>

<div id="demo">
<button type="button" onclick="loadXMLDoc()">Get Content</button>
</div>

<script>
function loadXMLDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "REPLACE_WITH_LINK_TO_YOUR_API", true);
  xhttp.send();
}
</script>

</body>
</html>
```

Make sure to replace "REPLACE_WITH_LINK_TO_YOUR_API” with the link to your API, for example https://api-websec.herokuapp.com/ .

Deploy this HTML page on Netlify and browse to it.

Open developer tools. Why does it fail? Fix it so that it works.