UPT UNIVERSIDADE PORTUCALENSE

CONTENT

1. Context
2. XML
    1. XML example
3. JSON
4. JSON vs. XML
5. AJAX
6. XMLHttpRequest
7. Classic model vs. AJAX model
8. Same-Origin Policy (SOP)
    1. DOM access
    2. AJAX Requests
    3. Data Storage
    4. Cookies
9. Circumventing Same-Origin Policy Restrictions: CORS/HTML5
    1. CORS: Requests with credentials
    2. CORS: Simple requests
    3. CORS: Preflighted requests

Context

- Client/server communication is based on HTTP protocol
- Needs a standard format to exchange information. Examples:
  - *Extensible Markup Language* (**XML**): https://www.w3schools.com/xml
  - *JavaScript Object Notation* (**JSON**): https://www.w3schools.com/js/js_json_intro.asp
- **API** (*Application Programming Interface*): flexible form of communicating with the web server. Example:
  - *Asynchronous JavaScript and XML* (**AJAX**): https://www.w3schools.com/js/js_ajax_intro.asp
    - *XMLHttpRequest* (**XHR**) https://www.w3schools.com/xml/xml_http.asp

XML

- Mark-up language, as HTML, but with no predefined tags
- W3C Standard.
- Example

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Note

To: Tove

From: Jani

Reminder

Don't forget me this weekend!

- Extensível

```
<note>
  <date>2015-09-01</date>
  <hour>08:30</hour>
  <to>Tove</to>
  <from>Jani</from>
  <body>Don't forget me this weekend!</body>
</note>
```

Note

To: Tove

From: Jani

Date: 2015-09-01 08:30

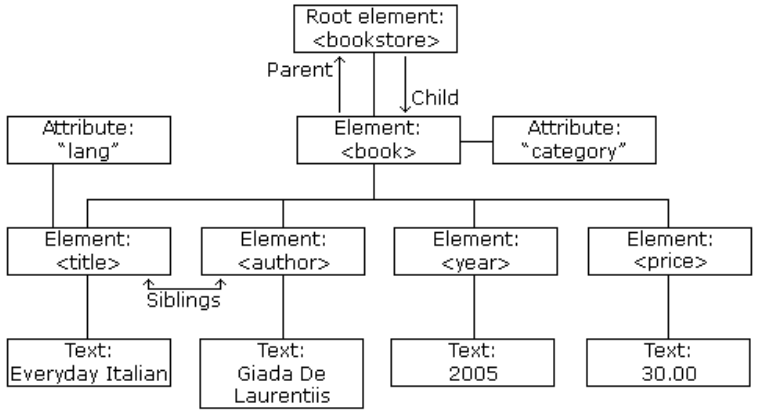
Don't forget me this weekend!

XML example: Books

Books.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>

  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>

  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>

  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>

  <book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>

</bookstore>
```

Document tree:



Title	Author
Everyday Italian	Giada De Laurentiis
Harry Potter	J K. Rowling
XQuery Kick Start	James McGovern
Learning XML	Erik T. Ray

JSON

- Can save the same information as XML

data1.json

```
let data = {
    "note" : {
        "to" : "Tove",
        "from" : "Jani",
        "heading" : "Reminder",
        "body" : "Don't forget me this weekend!"
    }
}
```

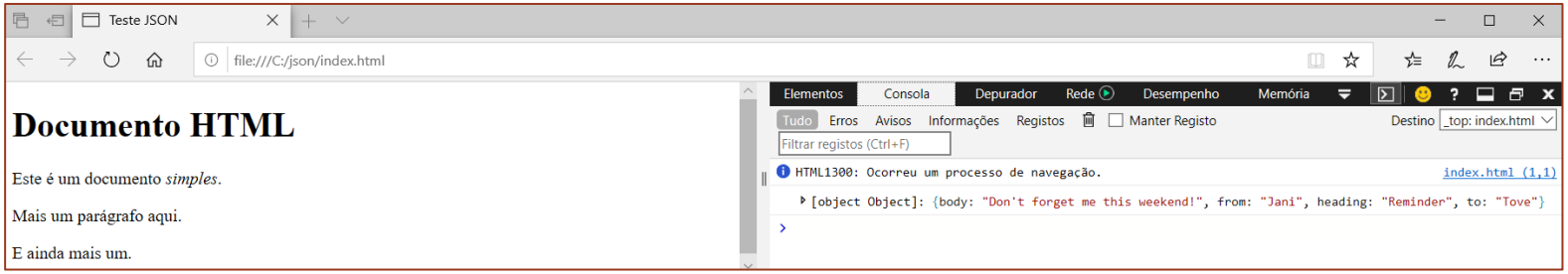
data2.json

```
let data = {
    "note1" : {
        "to" : "Tove",
        "from" : "Jani",
        "heading" : "Reminder",
        "body" : "Don't forget me this weekend!"
    },
    "note2" : {
        "to" : "Jani",
        "from" : "Tove",
        "heading" : "Reminded",
        "body" : "I won't forget!"
    }
}
```

- Easily and directly convertible to JavaScript

script1.js

```
let obj = JSON.parse('{"to" : "Tove","from" : "Jani","heading" : "Reminder","body" : "Don\'t forget me this weekend!"}')
console.log(obj)
```



- Verify document structure: https://jsoneditoronline.org/

JSON vs. XML

JSON

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

XML

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

Similarities

- Self-describing
- Hierarchical
- Parsing in multiple languages
- Allow using XMLHttpRequest

Differences

- JSON has no end-tag
- JSON is shorter
- JSON is faster reading/writing
- JSON allows using arrays

AJAX

Asynchronous JavaScript and XML

- Set of technologies used for asynchronous communication
- Includes XHR
- Allows data to be sent (by XHR) between the browser and the server
    - No need for page refresh
- Requests triggered via JavaScript
- Problem: different browsers may implement AJAX in different ways

XMLHttpRequest

- May be used to request data from a server

- Allows:
  - Updating the page without needing to refresh
  - Exchange (request / receive / send) data with a server after the page is loaded

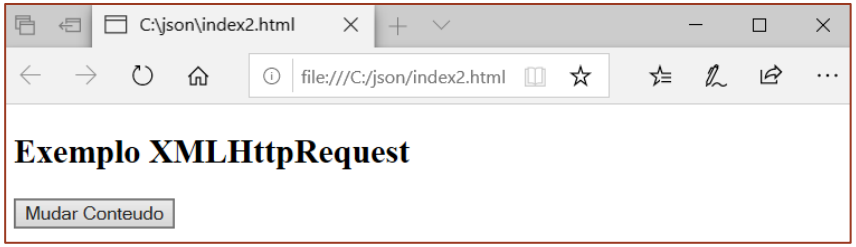
- Example:

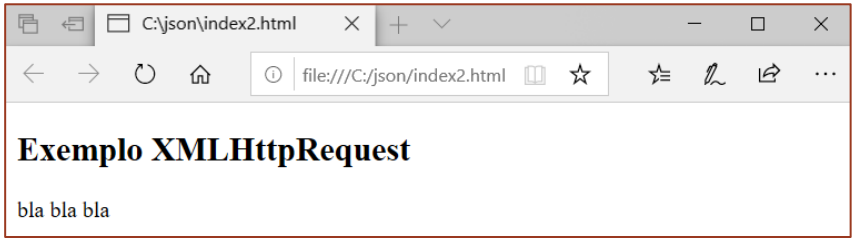
index2.html

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

<h2>Exemplo XMLHttpRequest</h2>

<div id="demo">
<button type="button" onclick="loadXMLDoc()">Mudar Conteudo</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", "xmlhttp_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```

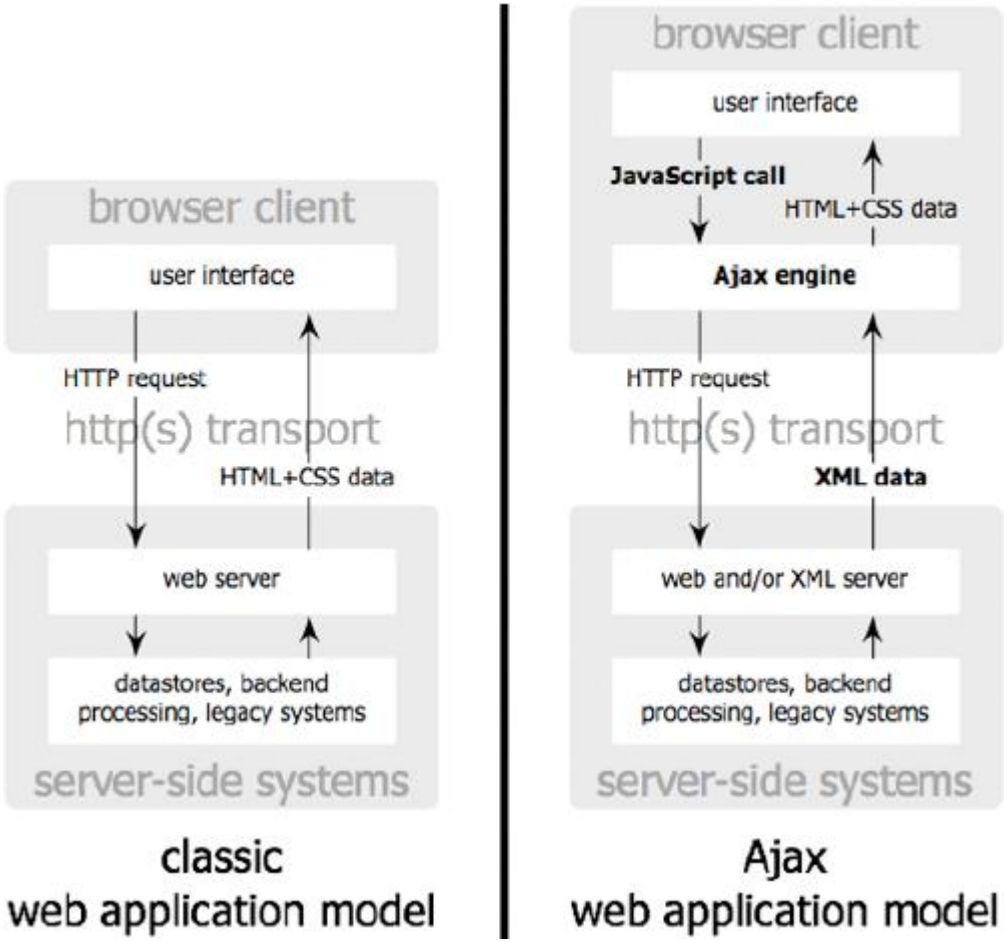
xmlhttp_info.txt

```
bla bla bla
```





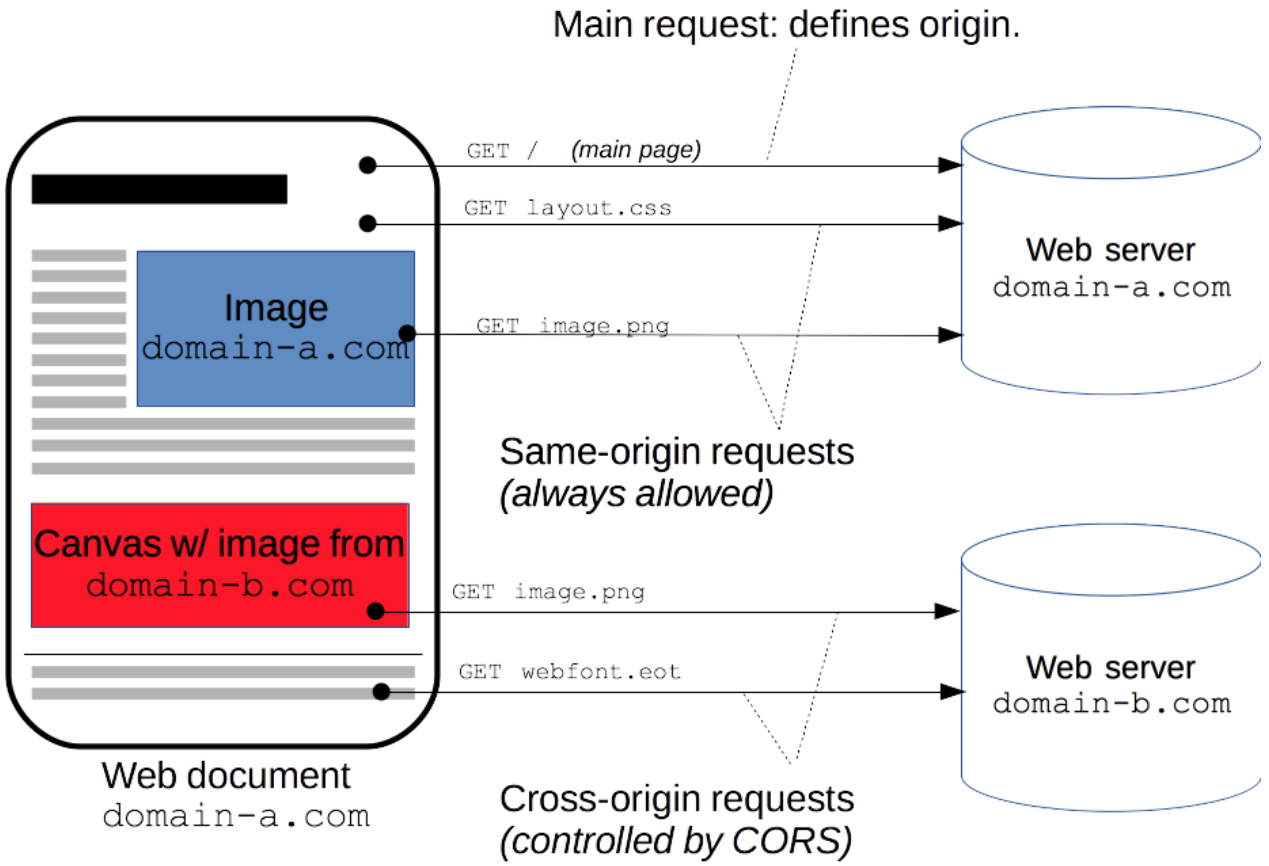
Classic model vs. AJAX model



Garrett, J. J. (2005). Ajax: A new approach to web applications.

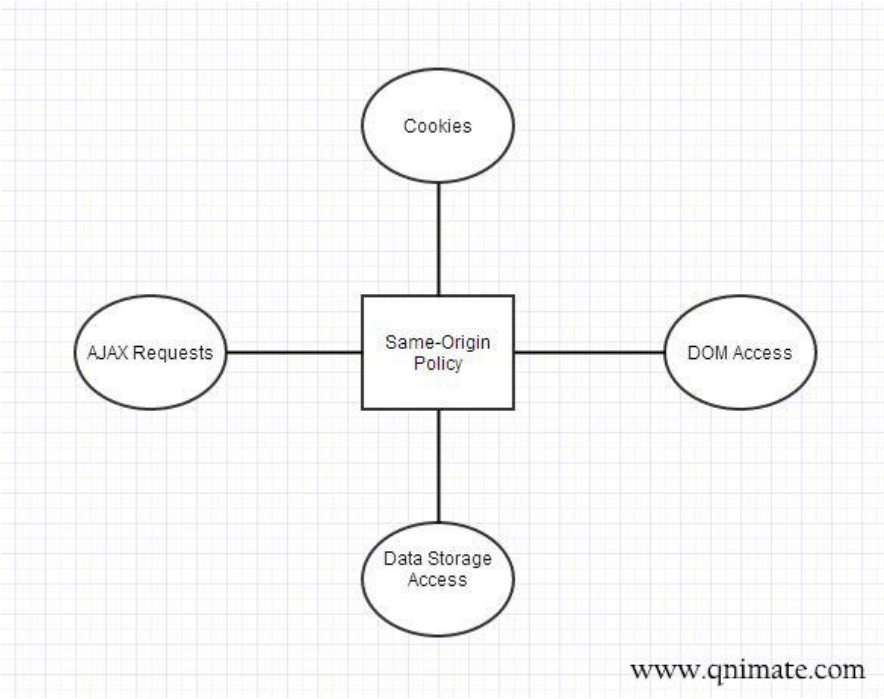
Same-Origin Policy (SOP)

Same origin: same protocol, domain and port

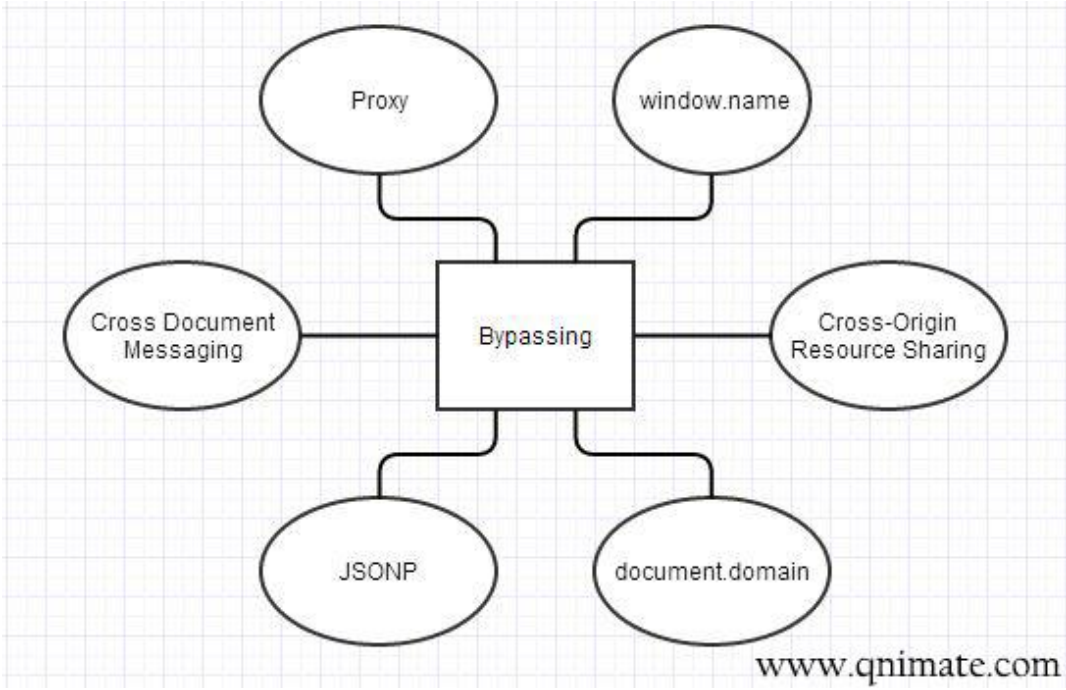


SOP

Restrictions



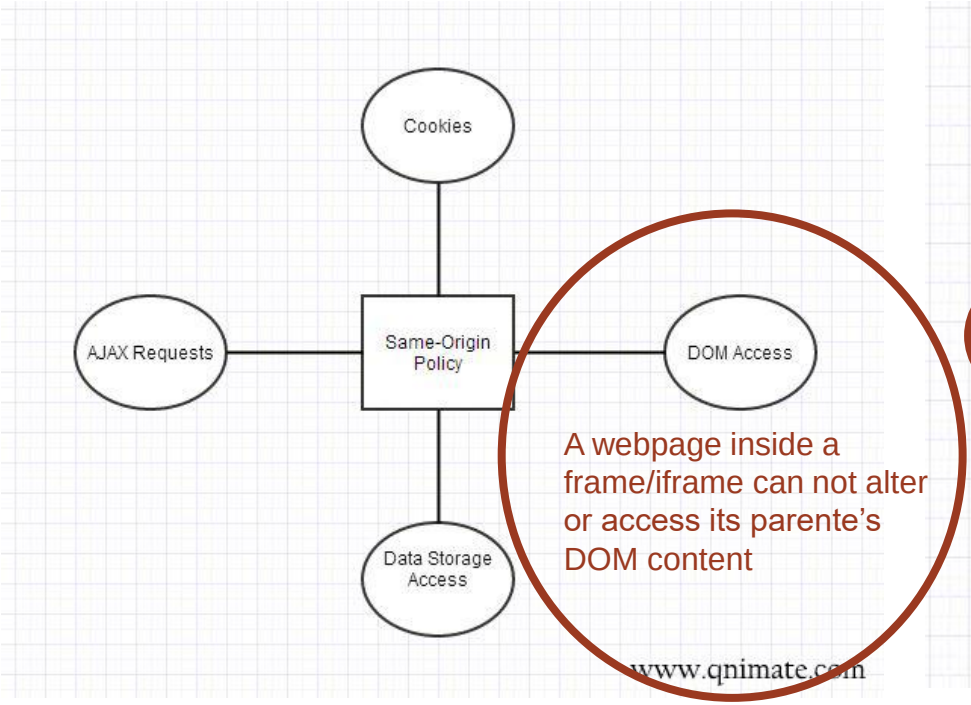
Circumventing restrictions



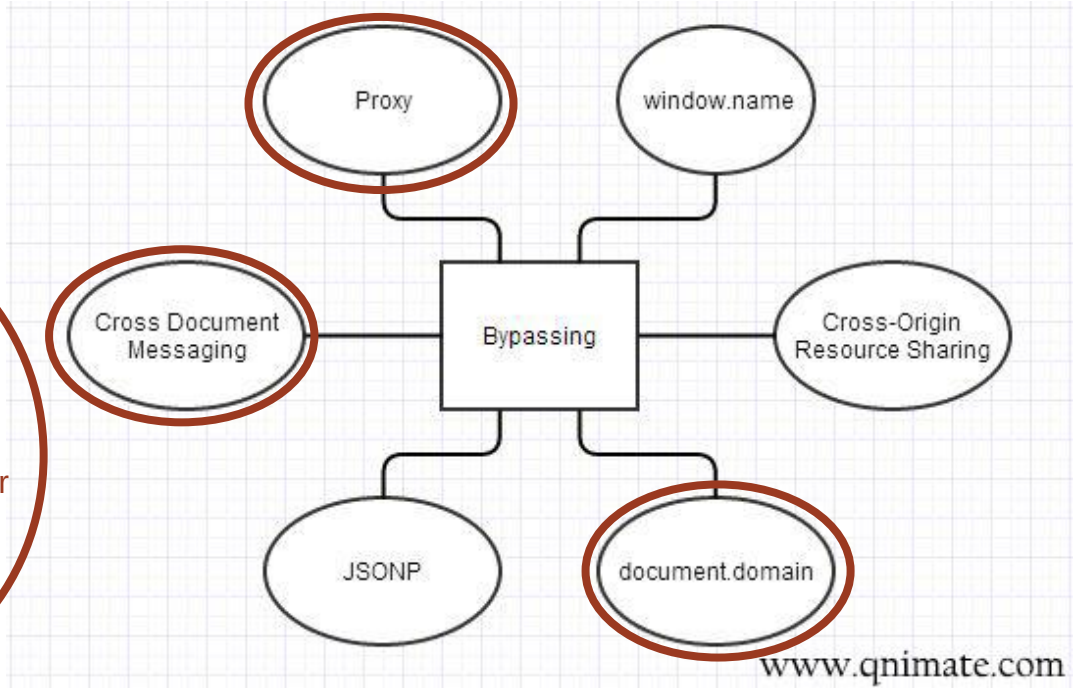
More information: http://qnimate.com/same-origin-policy-in-nutshell/

SOP: DOM access

Restrictions

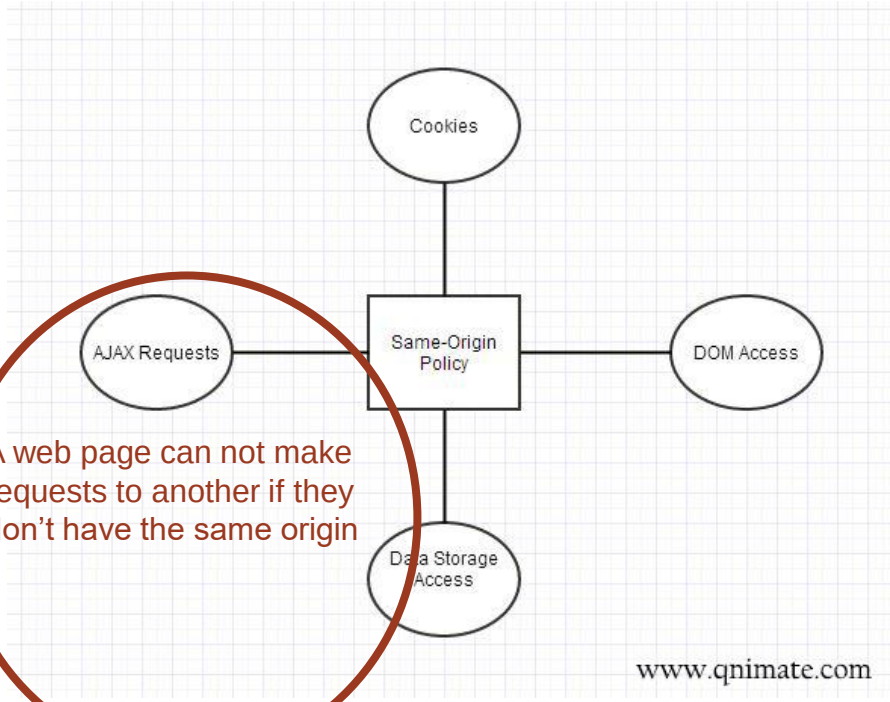


Circumventing restrictions



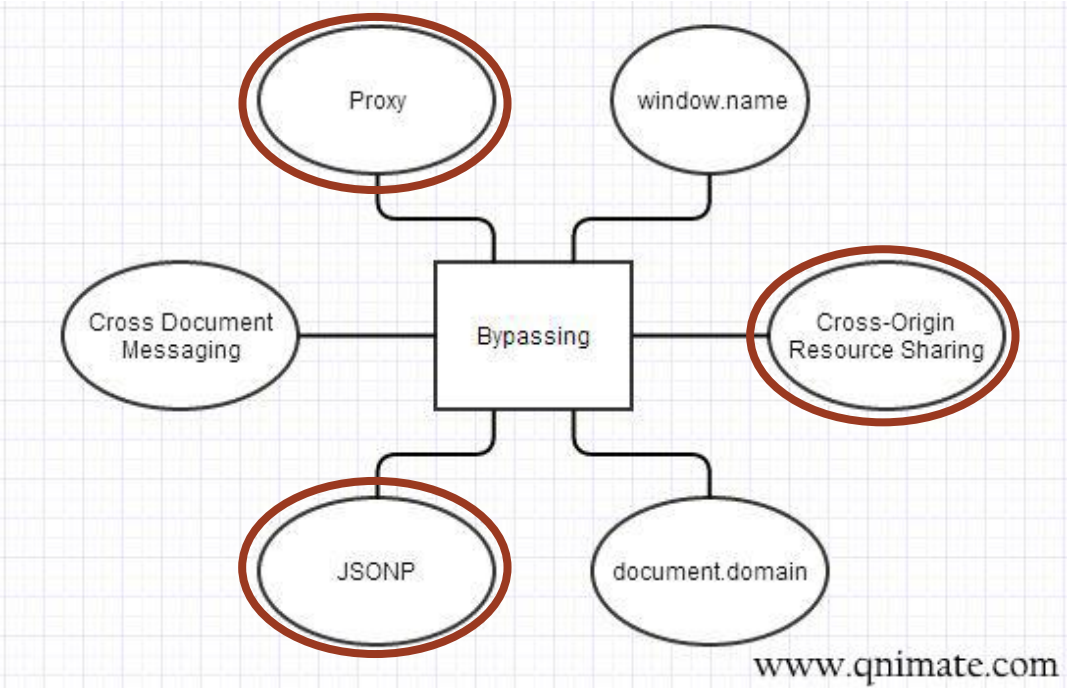
SOP: AJAX Requests

Restrictions



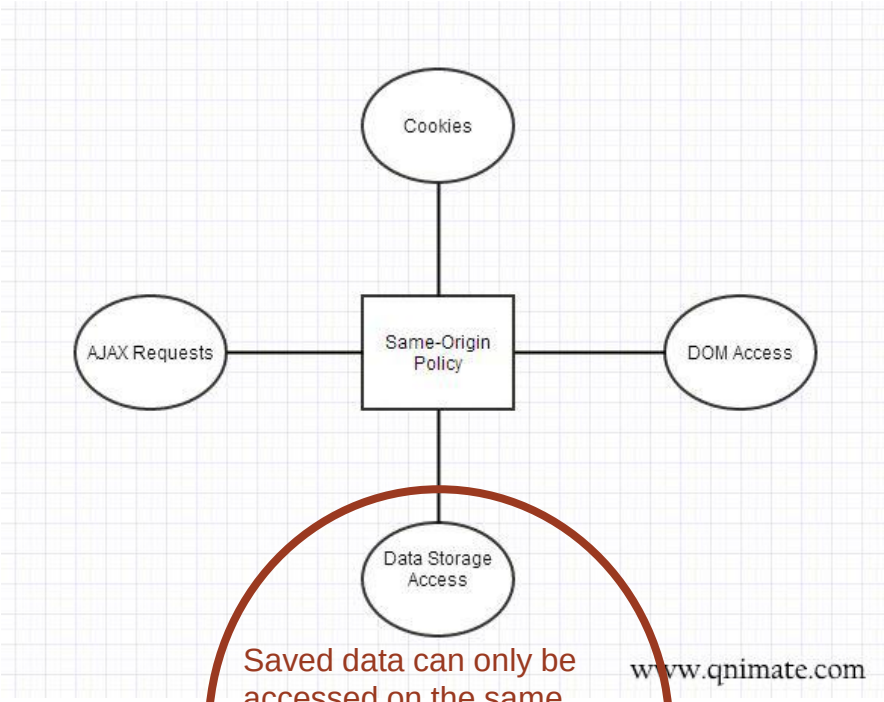
A web page can not make requests to another if they don't have the same origin

Circumventing restrictions



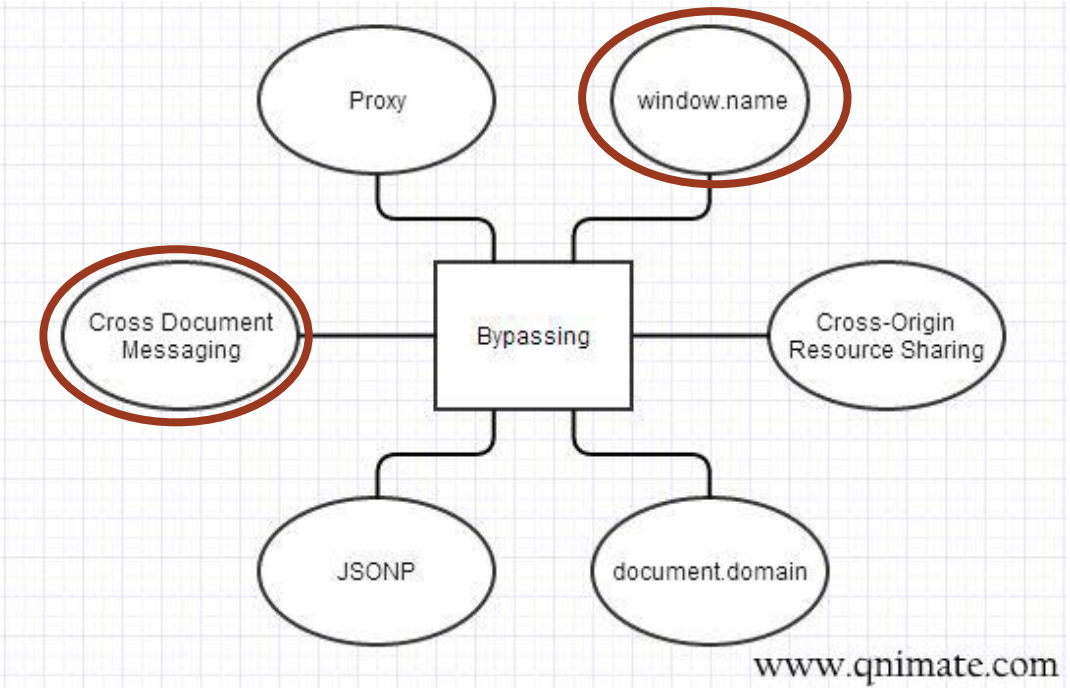
SOP: Data Storage

Restrictions



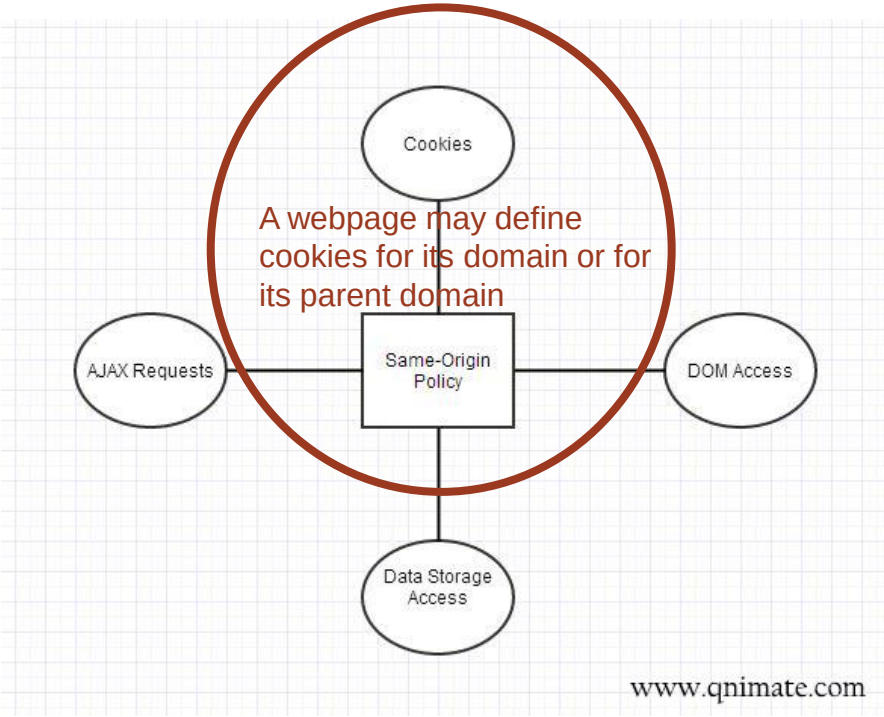
Saved data can only be accessed on the same origin from where they were saved.

Circumventing restrictions

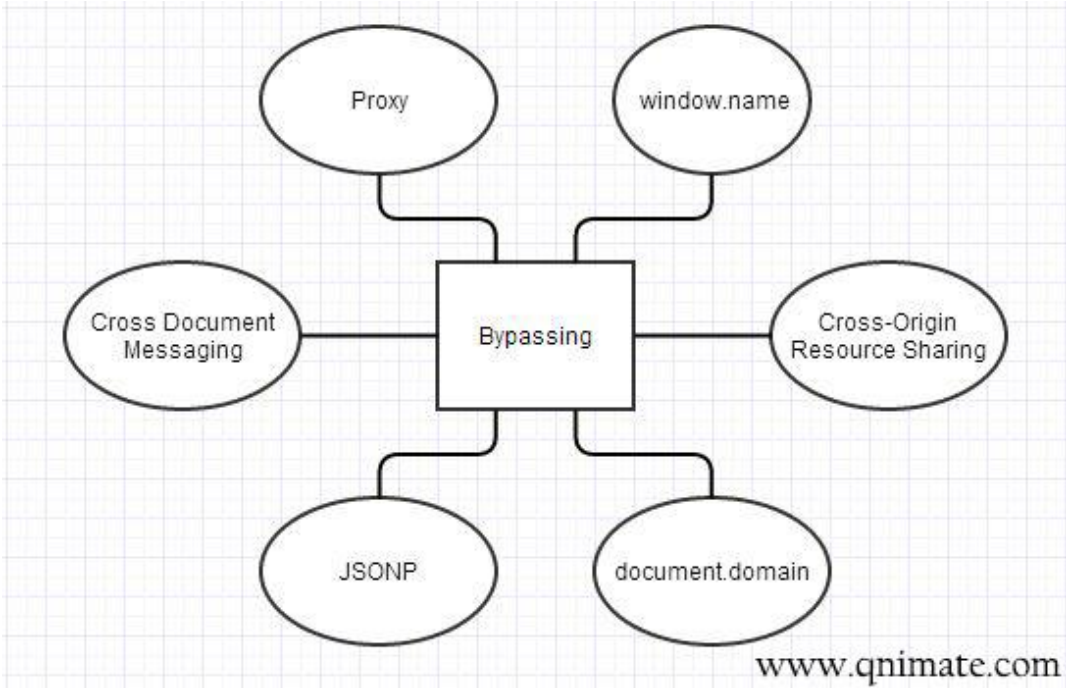


SOP: Cookies

Restrictions



Circumventing restrictions



Circumvent Same-Origin Policy restrictions: CORS/HTML5

- **CORS** (*Cross-Origin Resource Sharing*) – Sharing resources wtith different origins

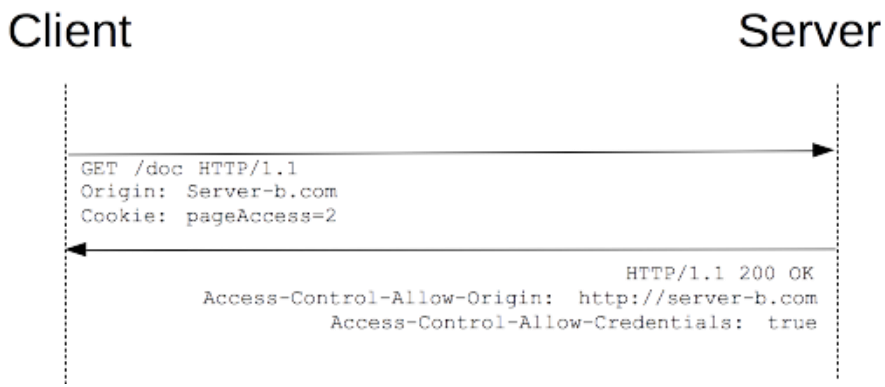
https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

- Use additional HTTP headers to inform the browser that it must allow a web application to be executed in an origin with permission to access resources in a different origin
- For security reasons, browsers block cross-origin HTTP requests made from scripts.

- For example, XMLHttpRequest follows the *same-origin policy* - **SOP**.
    - An application can only perform HTTP requests to the same origin from where it was loaded, unless the the other origin's response includes the proper CORS headers.

CORS: *Requests with credentials*

- By default, browsers don’t send credentials, unless explicitly specified.
- In this case, the credentials are sent in the form of cookies
- The server’s reply must include the origin in the Access-Control-Allow-Origin field, instead of the * wildcard



CORS: *Simple requests*

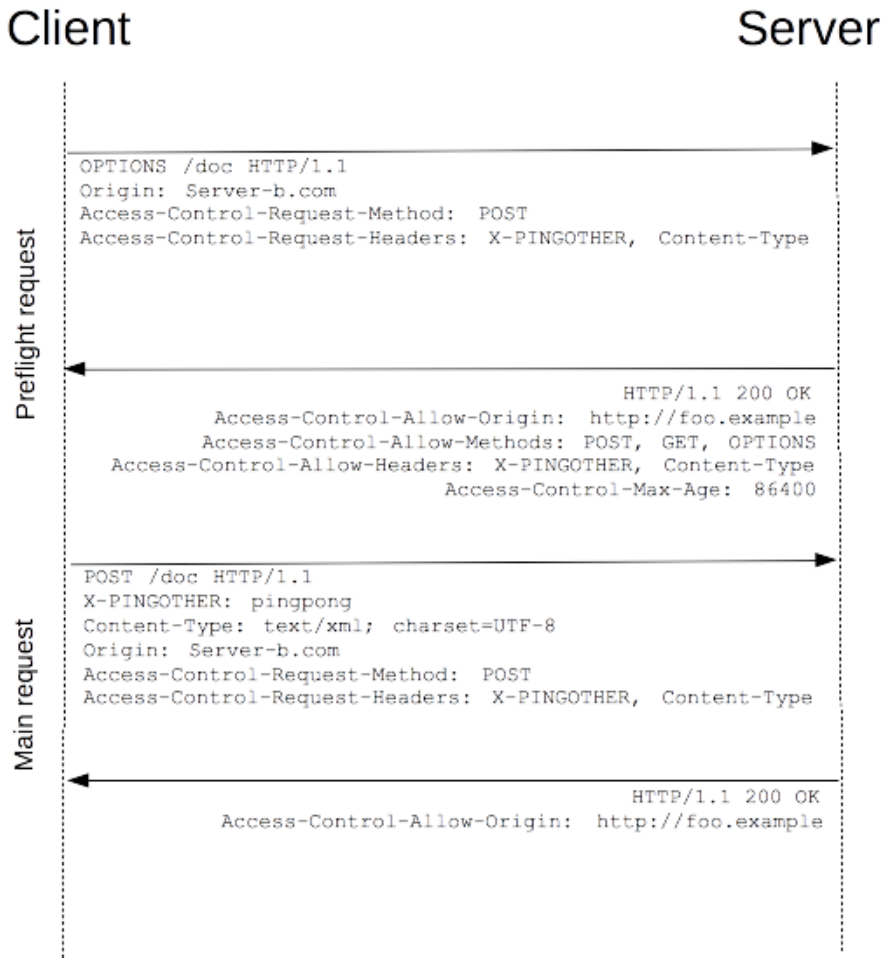
- Don’t trigger a CORS preflight
- Allowed **methods**: GET, HEAD, POST
- Besides the headers automatically defined by the agent, other headers may be manually defined: Accept, Accept-Language, Content-Language, Content-Type (see special requirements below), DPR, Downlink, Save-Data, Viewport-Width, Width
- Values allowed for the o **Content-Type**: application/x-www-form-urlencoded, multipart/form-data, text/plain
- No ***event listener*** is registered on any XMLHttpRequestUpload object used on the request (accessed with XMLHttpRequest.upload)
- No ReadableStream **object** is used on the request

CORS: *Preflighted requests*

- Send an HTTP request throught the OPTIONS method to obtain a resource in another domain, to determine if the resource is safe

- Uses one of the **methods**: PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH

- Don't respect what is presented for simple request in terms of **headers**, **Content-type**, ***event listeners*** and **objects**



UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.