



JavaScript - Consuming Services

Internet Services Architectures

Michał Wójcik



JavaScript - scripting programming language:

- commonly used with web technologies,
- weak typing,
- interpreted dynamically,
- allows to manipulate DOM objects tree,
- usually executed on client side in the browser.



Different methods for adding JavaScript scripts to the HTML page:

- place it inside `<script>` tag in `<head>` or `<body>` sections,
- load external file using `<script src=""></script>`.

It's recommended to separate logic (scripts) from view (HTML document).



```
<html>
  <head>
    <script src="script.js"></script>

    <script>
      <!-- Methods declaration. -->
    </script>
  </head>

  <body>
    <script>
      <!-- Script to be executed. -->
    </script>
  </body>
</html>
```

JavaScript allows to **modify** DOM tree elements:

- fetch element using:
 - `document.getElementById("id");`
 - `document.getElementsByTagName("name");`
 - `document.getElementsByClassName("name");`
- change element content with `element.innerHTML = "value";`
- change element attribute with `element.attribute = "value";`
- change CSS styles with `element.style.key = "value";`

```
<p id="test"/><!-- empty paragraph -->

<script>
  let test = document.getElementById("test");
  test.innerHTML = "Hello World!";
  test.style.color = "red";
  test.align = "right";
</script>
```

Using the `innerHTML` can be used for Cross-site Scripting (XSS).

JavaScript allows for dynamic modification and creation of DOM elements:

- `document.createElement("tag name")` - create new element,
- `element.appendChild(el)` - add element to another one.

```
<div id="container"></div>
```

```
let container = document.getElementById("container");  
  
let span = document.createElement("span");  
let text = document.createTextNode("woof");  
  
span.appendChild(text);  
container.appendChild(span);
```

JavaScript allows executing specified functions as reaction to DOM tree **events**:

- loading particular element,
- change of element content,
- clicking on element (e.g.: buttons),
- ...

```
<element event="JS code"/>
```




Selected events:

- **onchange** - element change,
- **onclick** - clicking element,
- **onmouseover** - cursor moved over the element,
- **onmouseout** - cursor moved out the element,
- **onkeydown** - pressing keyboard button,
- **onload** - loading document.



```
<html>
  <head>
  </head>

  <body>
    <script>
      window.onload = function (e) {
        alert('woof');
      }
    </script>
  </body>
</html>
```



AJAX - Asynchronous JavaScript and XML:

- updating page without reloading whole content,
- downloading data from server,
- sending data to server.

```
let xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        let object = JSON.parse(this.responseText);
    }
};

xhttp.open("GET", "http://localhost:8080/api/wolves", true);
xhttp.send();
```



```
let xhttp = new XMLHttpRequest();  
let wolf = {name: "green", age: 9};  
  
xhttp.open("POST", "http://localhost:8090/api/wolves", true);  
xhttp.setRequestHeader("Content-type", "application/json");  
xhttp.send(JSON.stringify(wolf));
```



```
let xhttp = new XMLHttpRequest();  
  
xhttp.open("DELETE", "http://localhost:8090/api/wolves/0", true);  
xhttp.send();
```

SOP - Same Origin Policy:

- origin (protocol, host and port) defines web application,
- different applications can not exchange content:
 - images linking,
 - sending forms to different server;
- secures against XSS (Cross-site scripting) and CSRF (Cross-site request forgery) attacks,
- default browser behavior,
- blocks AJAX requests between applications.

CORS - Cross-Origin Resource Sharing:

- different applications can exchange content,
- defines using headers on the server side,
- allows for:
 - JavaScript applications to communicate with API on different servers,
 - SSO (Single Sign On) systems for a number of applications.



- Mozilla, *JavaScript*, <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- w3schools, *CSS Tutorial*, <https://www.w3schools.com/js/>.