



Web Engineering & Design 1

# AJAX

## AJAX Concepts

Tobias Blaser



**HSR**

HOCHSCHULE FÜR TECHNIK  
RAPPERSWIL

FHO Fachhochschule Ostschweiz

# Content

## AJAX

- AJAX Basics
- AJAX Native
- AJAX & jQuery
- Same Origin Policy
- AJAX & Forms

# Learning Goals

You are able to ...

- ... explain which requests are possible by AJAX and which not.
- ... list formats often transferred by AJAX.
- ... implement AJAX calls using the native XHR.
- ... use the browser developer tools to analyse AJAX requests and extract information about the request like request content, header fields, type.

# jQuery

You are able to ...

- ... implement AJAX calls using jQuery.
- ... rewrite a native AJAX call to jQuery und reversed.
- ... submit forms by AJAX.

# CORS

You are able to ...

- ... explain why the same-origin policy is necessary.
- ... list solutions to maxe cross-origin calls.

# Introduction

Home Ticket Shop Newsletter SBB Blog Kontakt Jobs Suche de fr it en 

Freizeit & Ferien Fahrplan Abos & Billette Bahnhof & Services Geschäftsreisen Konzern Login

**Fahrplan & Billettkauf**

Von: Rapperswil  
Nach: Rapperswil  
Via: Rapperswil West  
Rapperswil SG, EW  
Rapperswil BE, Post  
Rapperswil SG, Bildau  
Ab: Rapperswil SG, Bahnhof  
Rapperswil BE, Ziegelei  
Rapperswil SG, Zeughaus  
Rapperswil SG, Cityplatz

Zu zweit durchs Erlebnisland Schweiz.  
Jetzt online kaufen.

MITFAHRBILLET  
STATT 124.-  
CHF 37.-  
NUR CHF



[www.sbb.ch](http://www.sbb.ch)

Create your Google Account

One account is all you need  
One free account gets you into everything Google.



Name

Choose your username

Someone already has that username. Try another?  
Available: elooperdev378 elooper376 elooperd34

[accounts.google.com/SignUp](https://accounts.google.com/SignUp)

**OpenStreetMap** Edit History Export

Search Where am I? Go

**Query Features**

Nearby features

Enclosing features

Network Storage

Method	File	Domain	Type
GET	query?lat=47.22328&lon=8.81865&xhr=1	www.openstreetmap.org	html
POST	interpreter	overpass-api.de	json
POST	interpreter	overpass-api.de	json
GET	query?lat=47.22325&lon=8.81864&xhr=1	www.openstreetmap.org	html
POST	interpreter	overpass-api.de	
POST	interpreter	overpass-api.de	

**Query Features**  
Click on the map to find nearby features.

**Nearby features**  
Searching

**Enclosing features**  
Searching

[www.osm.org](http://www.osm.org)

# Motivation

- Enhanced user experience

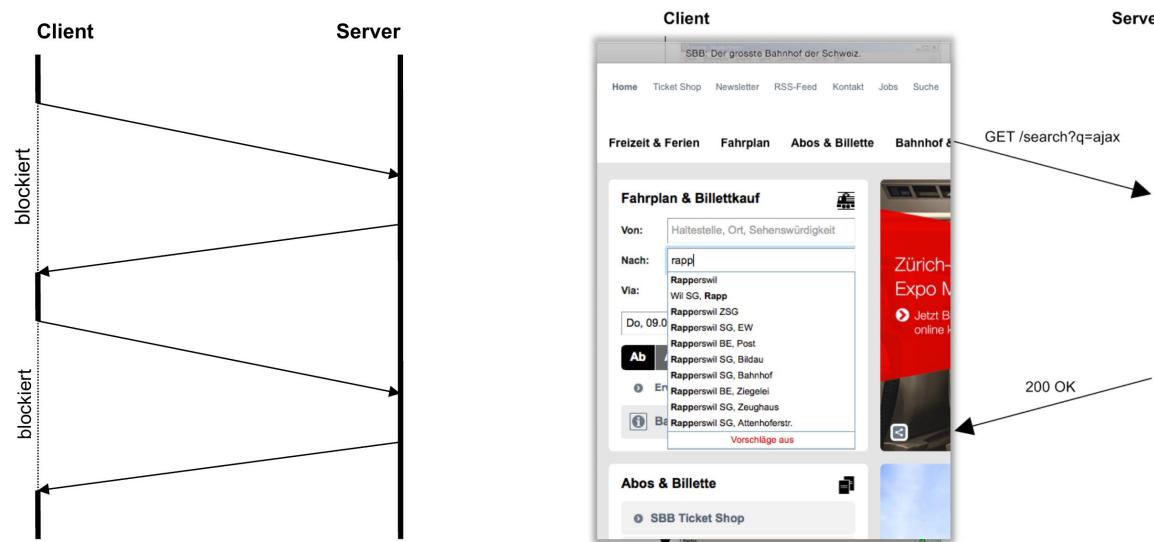
The screenshot shows the SBB website's search interface for travel plans. On the left, there is a form with fields for 'Von' (Rapperswil), 'Nach' (Rapperswil, Rapperswil ZSG), 'Via' (Rapperswil West, Rapperswil SG, EW), 'Di, 19.04.' (date), 'Ab' (origin station), and 'Er' (arrival station). There is also a button 'Vorschläge aus' (suggestions out) and a link 'Bahn' (train). On the right, there is a promotional banner for a roller coaster ride titled 'Zu zweit durchs Erlebnisland Schweiz.' It features a couple on a red roller coaster car against a backdrop of Swiss mountains and a lake. The banner includes the text 'Jetzt online kaufen.' and a price comparison: 'MITFAHRBILLET STATT CHF 124.- NUR CHF 37.-'.

- Fast application → desktop feeling
  - No flickering
- Minimized payload → mobile friendly

# What is AJAX?

« With AJAX, web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page »

[wikipedia.org](https://en.wikipedia.org/wiki/Ajax_(programming))



[www.sbb.ch](http://www.sbb.ch)

# AJAX Native

Demo

## Sky Webshop

### Products



**T-Shirt airplane "A380"**

[⊕ product details](#)



**Memory-Stick bus "L5"**

[⊕ product details](#)



**Bolster "dreams"**

[⊕ product details](#)



**Earplugs "sleep quiet"**

[⊕ product details](#)

# XMLHttpRequest (XHR) Demo

Demo

```
// create new AJAX request
var request = new XMLHttpRequest();

// handle successfull response
request.onreadystatechange = function() {
    if (request.readyState == 4 && request.status == 200) {

        // do something with 'request.responseText'
        window.alert(request.responseText);
    }
};

// fire asynchronous AJAX request (true is default)
request.open('GET', 'http://domain.tld/product-details', true);
request.send();
```

- Use **async** request → sync request will block main thread

```
request.open("GET", url, false)
```

- If necessary, use worker thread



Demo

```
// wait until DOM is ready
document.addEventListener('DOMContentLoaded', function() {
    // get all 'detail'-links
    var detailButtons = document.querySelectorAll('.details > a');

    // loop through all links
    Array.prototype.forEach.call(
        detailButtons, function(detailButton) {
            // add click listener to every link
            detailButton.addEventListener('click', function(event) {
                // detail view container (containing the link)
                var linkElement = event.target;

                // get url from link element
                var url = linkElement.getAttribute('href');
            });
        });
});
```

Demo [examples/webshop/3-solution/](#)



Demo

```
var request = new XMLHttpRequest();

request.onreadystatechange = function() {
    var status = request.status;
    if (request.readyState == 4 && (status == 200 || status == 201 || status == 304)) {

        // replace link by response
        linkElement.parentNode.innerHTML =
            request.responseText;
    }
};

request.open("GET", url);
request.send();

// prevent opening original link
event.preventDefault();
}) ;
} ) ;
} ) ;
```

Demo [examples/webshop/3-solution/](#)

Headers | Cookies | Params | Response | Timings | Preview

**Request URL:** http://localhost:63342/HSR.WED1.AJAX.Slides/1-AJAX/examples/webshop/product-details/t-shirt-a380.html

**Request method:** GET

**Remote address:** 127.0.0.1:63342

**Status code:** 200 OK

**Version:** HTTP/1.1

Filter headers

Response headers (0.209 KB)

**Cache-Control:** "private, must-revalidate"

**Content-Length:** "246"

**Content-Type:** "text/html"

**Date:** "Mon, 25 Apr 2016 19:26:31 GMT"

**Last-Modified:** "Tue, 19 Apr 2016 22:45:55 GMT"

**Server:** "WebStorm 11.0.1"

Request headers (0.510 KB)

**Host:** "localhost:63342"

**User-Agent:** "Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:45.0) Gecko/20100101 Firefox/45.0"

**Accept:** "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8"

**Accept-Language:** "en-US,en;q=0.5"

**Accept-Encoding:** "gzip, deflate"

**DNT:** "1"

**Referer:** "http://localhost:63342/HSR.WED1.AJAX.Slides/1-AJAX/examples/webshop/3-solution/index.html"

**Connection:** "keep-alive"

**Pragma:** "no-cache"

**Cache-Control:** "no-cache"

Edit and Resend |  Raw headers

Demo

- GET-request
- Content-Length: 246 bytes
- Content-type: HTML

x Headers Preview Response Timing

1   
2 <dl>  
3 <dt>Price</dt>  
4 <dd>CHF 39.90</dd>  
5 <dt>Colors</dt>  
6 <dd>Blue, red, green</dd>  
7 <dt>Sizes</dt>  
8 <dd>S, M, L, XL</dd>  
9 <dt>Material</dt>  
10 <dd>Cotton</dd>  
11 </dl>

Demo

### Response source (HTML)

Headers	Cookies	Params	Response	Timings	Preview
T-Shirt A380					

Price  
CHF 39.90  
Colors  
Blue, red, green  
Sizes  
S, M, L, XL  
Material  
Cotton

### Response preview (HTML) in Firefox

Demo

```
▼<li>
  ▶<h3>...</h3>
  ▼<div class="details">
    <span>⊕</span>
    <a class="expand" href=".../product-details/t-shirt-a380.html" target="blank">product details
  </div>
</li>
```

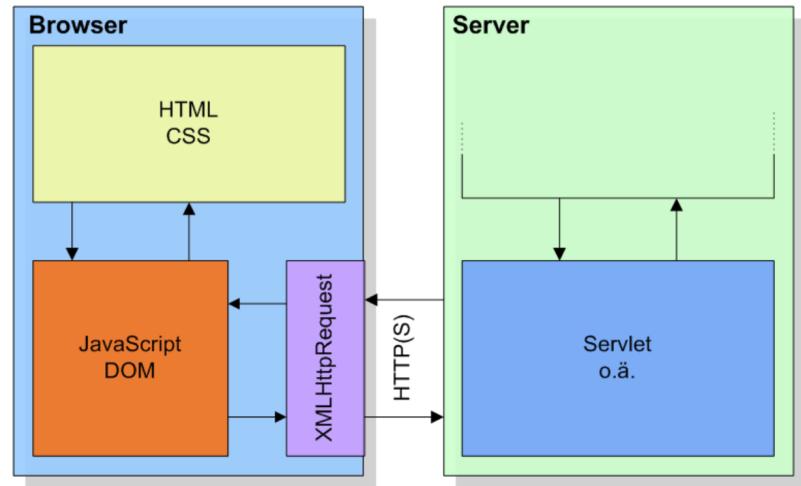
## DOM before insertion

```
▼<li>
  ▶<h3>...</h3>
  ▼<div class="details">
    
    ▼<dl>
      <dt>Price</dt>
      <dd>CHF 39.90</dd>
      <dt>Colors</dt>
      <dd>Blue, red, green</dd>
      <dt>Sizes</dt>
      <dd>S, M, L, XL</dd>
      <dt>Material</dt>
      <dd>Cotton</dd>
    </dl>
  </div>
</li>
```

## DOM after insertion

# AJAX Principles

- **AJAX request:**  
Asynchronous HTTP-Request
- **XHR (XmlHttpRequest):**  
JS API for AJAX-requests
- **Transfer formats:**
  - XML
  - Text
  - HTML
  - Binary
  - JSON
  - ...
- Server is not able to differ between normal HTTP and AJAX requests



# XMLHttpRequest

## 0 uninitialized

```
var request = new XMLHttpRequest();  
  
// get status  
console.log(request.readyState); // 0
```

Request object not yet initialized

## 1 open

```
request.open("GET", 'some-url.tld/', true);
```

Request object initialized but not yet fired

## 2 sent

```
request.send();
```

Request fired

## 3 receiving

Parts of the response are already available

## 4 completed

Request terminated

```
// request dump
XMLHttpRequest {
    onreadystatechange: xr.onreadystatechange(),
    readyState: 4,
    timeout: 0,
    ...
    responseURL: "http://localhost:63342/...",
    status: 200,
    statusText: "OK",
    responseType: "",
    response: "<img src=\"./images/t-shirt-a380.jpg\"..."
}
```

```
if (request.readyState == 4 && request.status == 200) { ...
```

- **onreadystatechange**
  - Called on every state change
- **request.responseText**
  - Response as plain text
- **request.responseXML**
  - Response as XMLHttpRequest object
  - For **XML-responses** only
  - Content type **text/xml** nessecary
- **Status**
  - **request.status**: Status code, z.B. 200, 404
  - **request.statusText**: z.B. OK, Not Found

# POST-Request

```
var data = "id=96&category=6";  
  
request.open("POST", url);  
  
// Content type necessary  
request.setRequestHeader(  
    "Content-Type",  
    "application/x-www-form-urlencoded"  
);  
  
request.send(data);
```

- Every type of data may be sent in POST-body
  - Correct content type **necessary**
  - Server is able to handle response by content type

# AJAX & jQuery

Demo

## Connections to Rapperswil

From  To  Get connections

[examples/autocomplete/client/](#)

Suggestion API:

`http://localhost:8080/suggestions?search=Rapp`

```
<option value="Rapperswil">Rapperswil</option>
<option value="Rapperswil, ZSG">Rapperswil, ZSG</option>
<option value="Rapperswil, Bahnhof">Rapperswil, Bahnhof</option>
<option value="Rapperswil, Sonnenhof">Rapperswil, Sonnenhof</option>
```

[localhost:8080/](#)

# \$.get() demo

Demo

```
var url = 'http://domain.tld/suggestions?search=' + searchString;
// fetch from URL -> handle data
$.get(url, function(suggestionData) {
    // do something with suggestions
    window.alert(suggestionData);
}) ;
```

HTML snippet:

```
<datalist id="suggestions"></datalist>
<form>
    <label>
        From
        <input type="text" id="from" name="from"
               list="suggestions" />
    </label>
    ...
</form>
```



Demo

```
// wait until DOM is ready
$(document).ready(function () {

    // get from-field and datalist element
    var suggestionAPI = 'http://localhost:8080/suggestions?search='
    var fromField = $('#from');
    var suggestions = $('#suggestions');

    // input change event listener
    fromField.on('input', function() {

        if(fromField.val().length >= 3) {
            var url = suggestionAPI+fromField.val();
            $.get(url, function(suggestionData) {
                // insert suggestions in datalist
                // jQuery .html() is slow in some Browsers
                suggestions.get(0).innerHTML = suggestionData;
            });
        } else {
            // clear suggestions if input < 3 chars
            suggestions.empty();
        }
    });
});
```

From  To  Get connections

Rapperswil  
Rapperswil, ZSG  
Rapperswil, Bahnhof  
Rapperswil, Sonnenhof



Inspect... Console Debug... Style E... Perform... Network Storage Response

✓ Method File

- 200 GET suggestions?search=Rap
- 200 GET suggestions?search=Rapp

	Headers	Cookies	Params	Response
1	<option value="Rapperswil">Rapperswil</option>			
2	<option value="Rapperswil, ZSG">Rapperswil, ZSG</option>			
3	<option value="Rapperswil, Bahnhof">Rapperswil, Bahnhof</option>			
4	<option value="Rapperswil, Sonnenhof">Rapperswil, Sonnenhof</option>			
5				

## Suggestion requests

```

<h1>Connections to Rapperswil</h1>
▼<datalist id="suggestions">
  <option value="Rapperswil">Rapperswil</option>
  <option value="Rapperswil, ZSG">Rapperswil, ZSG</option>
  <option value="Rapperswil, Bahnhof">Rapperswil, Bahnhof</option>
  <option value="Rapperswil, Sonnenhof">Rapperswil, Sonnenhof</option>
</datalist>
▼<form>
  ▶<label>...</label>
  ▶<label>...</label>
  <button type="submit">Get connections</button>
</form>

```

## DOM after data insertion

# jQuery get(), post(), ajax() & load()

- `$.ajax()` → based on native XHR

```
$.ajax({  
    type: "POST",  
    url: "some/url/",  
    dataType: "html",  
    data: { name: "...", id: 456 }  
}).done(function(data, textStatus, jqXHR) {  
    $('#detail-view').html(data);  
});
```

- `$.get()` → ajax()-shortcut

```
$.get("some/url/", function(data) { /*do something with data*/ });
```

- `$.post()` → ajax()-shortcut

```
$.post("some/url/", postData, function(data) { /* do something ... */ });
```

- `$.load()` → get()-shortcut

```
$('#detail-view').load("some/url/");
```

`$.ajax()` is as powerful as native XHR (example):

```
$.ajax({
  type: "POST",
  url: "some/url/",
  data: { name: "...", id: 456 },
  dataType: "mycustomtype",
  cache: false,
  processData: false,
  accepts: {
    mycustomtype: 'application/x-some-custom-type'
  },
  beforeSend: function(xhr) {
    xhr.overrideMimeType("text/plain; charset=x-user-def");
  },
  converters: {
    'text mycustomtype': function(result) { ... return ... }
  },
  ...
}) // returns a jqXHR object
.done(function(data, textStatus, jqXHR) { ... })
.fail(function(jqXHR, textStatus, errorThrown) { ... })
.always(function(data|jqXHR, textStatus, jqXHR|error) { ... });
```

# AJAX Restrictions

## Same Origin Policy

- XHR is restricted to same origin (**security reasons**)
  - same protocol, domain and port
    - E.g. `https://domain.tld:80/api/suggestions/`
  - restriction is enforced by **client (Browser)**

## No Server Push

- Server is not able to **call the client**
  - Push technologies (WED2)
    - Poll, long poll, comet
    - WebSockets

# Solutions for Cross-Origin Requests

- Proxy on same server → redirect

1. XHR calls proxy
2. proxy calls remote

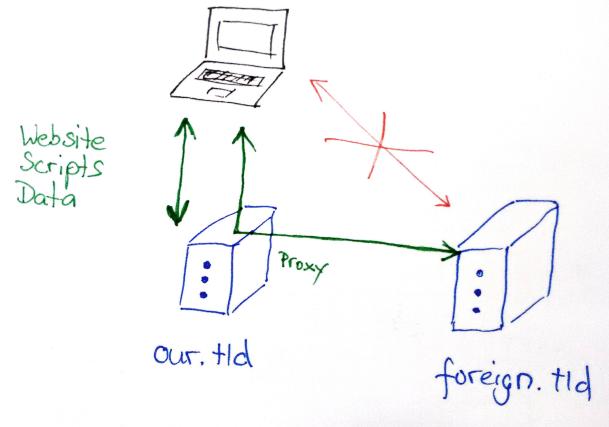
- On demand JavaScript

- Load data by including **scripts** containing Data

- No POST, only GET requests
  - **Outdated** → Use XHR **CORS** instead

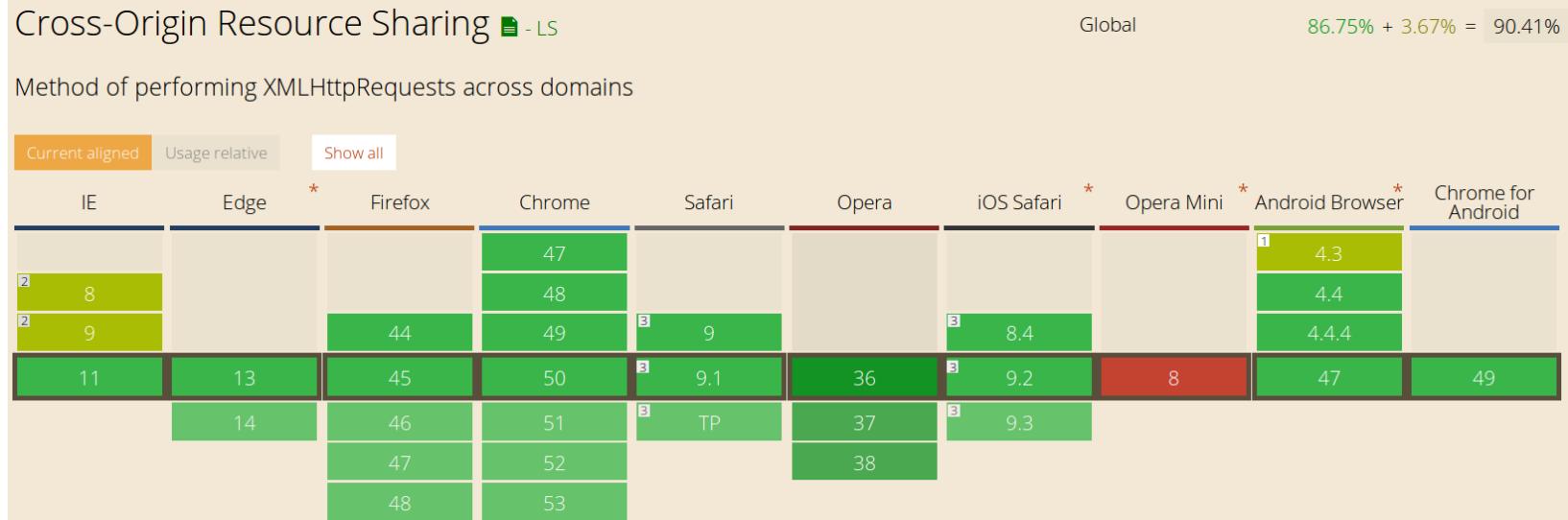
- XHR Level 2 → **CORS** (Cross Origin Resource Sharing)

- Server sent **Access-Control-Allow-Origin** header allows client to call the remote by AJAX



# XHR CORS

```
// CORS Header for Node.js
app.use('/remote-api/', function(req, res, next) {
  res.header('Access-Control-Allow-Origin', '*');
  res.header('Access-Control-Allow-Methods', 'GET, POST');
  res.header('Access-Control-Allow-Headers', 'Content-Type');
  next();
});
```



# CORS Challenges

- Firewall problems (corporate networks)
  - Block requests containing CORS-headers
  - Strip CORS-headers from requests
  - Difficult to detect and reproduce
- Maybe we can **not change** the target servers **configuration**
  - Corporate internal political problem, not technical ;-)
  - Target server is maintained by **another enterprise**

# Submit Form by AJAX

Demo

## Contact us

Name

Type of message

Feedback ▾

Message

submit

examples/form/  
localhost:8090



Demo

```
<h1>Contact us</h1>
<div id="status"></div>
<form action="http://localhost:8090/submit"
      method="POST" id="contactForm">
    ...
    <button type="submit">submit</button>
</form>
```

```
$(document).ready(function () {
  var form = $('#contactForm');

  // listen on form submit
  form.on('submit', function(event) {

    // send form data serializes (url encoded) by AJAX
    $.post(form.attr('action'), form.serialize(), function(data) {
      $('#status').get(0).innerHTML = data;
      form.get(0).reset();
    });

    // disable form default action
    event.preventDefault();
  });
});
```

Demo

## Thank you Hans Meyer

The following feedback was sent to us:

"I like your page very much!"

Name

Type of message

Message

submit

A screenshot of a browser's developer tools Network tab. The tab bar includes 'Inspec...', 'Console', 'Debug...', 'Style E...', 'Perfor...', 'Network' (which is selected), and 'Storage'. Below the tab bar, there is a table with the following data:

✓	Method	File	Domain	Type	Transfer
● 200	POST	submit	localhost:8090	xml	0.13 KB

To the right of the table, there are tabs for 'Headers', 'Cookies', 'Params' (which is selected), 'Response', and 'Timings'. Under the 'Params' tab, it shows 'Form data' with the following entries:

- name: "Hans+Meyer"
- message-type: "feedback"
- message: "I+like+your+page+very+much!"

## Form submit request

33 / 39



Web Engineering & Design 1

# AJAX

## AJAX concepts

Tobias Blaser



HOCHSCHULE FÜR TECHNIK  
RAPPERSWIL

FHO Fachhochschule Ostschweiz

# Self Check

1. Explain, what AJAX is.
2. How does an AJAX request affect the current visible page?
3. What's the difference between an AJAX request and a normal HTTP request?
4. Write a code snippet, to load a detail view from "/details/1/" and embed in

```
<div id="detail1"></div>
```

5. What restrictions has an AJAX request?

6. How to solve the same-origin issue? Discuss pros and contras.
7. Rewrite the snippet from above to use jQuery instead.
8. Rewrite the AJAX-form example to submit the form data by GET.

# Solutions

1. AJAX is a normal HTTP call fired in background without reloading the whole page.
2. Does not reload the current page.
3. AJAX requests are async in background.
4. See slide [#/5/3](#), replace linkElement by `document.getElementById('detail');` and the url by '/details/1/'.
5. No server push, no cross-origin (foreign domain) requests.

## 6. Proxy & CORS:

- Proxy: Reliable, works every time
- CORS: Easy to setup, maybe firewall problems

## 7. :

```
$.get('/details/1/' , function(detailView) {  
    document.getElementById('detail').innerHTML = detailView;  
} );
```

## 8. Append form data to url instead of POST body:

```
$.get(form.attr('action') + '?' + form.serialize() , function(data) {  
    ...  
} );
```



Web Engineering & Design 1

# AJAX

## AJAX Concepts

Tobias Blaser



HOCHSCHULE FÜR TECHNIK  
RAPPERSWIL

FHO Fachhochschule Ostschweiz