

Multimedia im Netz
Online Multimedia
Winter semester 2015/16

Tutorial 07 – Minor Subject



Today's Agenda

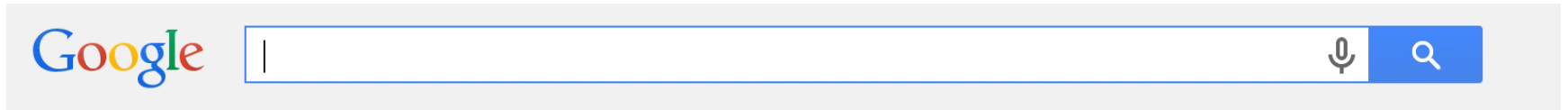
- Recap
- AJAX with jQuery
 - XML vs JSON
 - Example: Asynchronous Contact Form with PHP back-end
 - Form serialization
- Breakout Session: Movie Search
- Quiz

Evaluation & Feedback

Please help us improve the tutorials and assignments by filling out this survey:

<http://goo.gl/forms/DTdGh4TYaC>

AJAX: Google Instant Search



Recap

1. What does the \$ stand for in jQuery?
2. What is a „selector“?
3. What is „DOM-Manipulation“ good for?
4. What is a callback function?
5. How do you handle a „click“ event?
6. What is returned by `$('div');`;

AJAX with jQuery

AJAX

- Acronym: **A**synchronous **J**avaScript **A**nd **X**ML
- Allows passing around data between client- and server-side applications back and forth – **without refreshing the page**
- AJAX requests (also: XHR = XMLHttpRequest):
 - GET: retrieve data – no manipulation on the server
 - POST: retrieve and/or modify data

XML

- Extensible Markup Language
- Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<contact>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <phone>
    <type>Home</type>
    <number>5648978965</number>
  </phone>
  <phone>
    <type>Mobile</type>
    <numner>6458979878</numner>
  </phone>
</contact>
```


~~XML~~ JSON!

- Before JSON was widely adopted, XML was the state of the art
- JavaScript Object Notation
- Human-readable format for data exchange
- Based on key-value pairs and smaller than XML
- **Example:**

```
{
  "firstName": "John",
  "lastName": "Doe",
  "phone": [
    {
      "type": "Home",
      "number": "5648978965"
    },
    {
      "type": "Mobil",
      "nummer": "6458979878"
    }
  ]
}
```

AJAX with jQuery

- jQuery offers different functions to create AJAX-requests:
 - \$.ajax()
 - \$.get()
 - \$.post()
- More detailed information:
 - <https://api.jquery.com/jquery.ajax/>
 - <https://api.jquery.com/jquery.get/>
 - <https://api.jquery.com/jquery.post/>

Example: Asynchronous Contact Form

- Sending a contact form is a typical use case where AJAX is useful, because a complete page refresh is not necessary.
- Imagine the server responds with this JSON:

```
{  
  "status": "OK",  
  "info": {  
    "name": "Peter",  
    "message": "This is nice!"  
  }  
}
```

PHP Script handling the contact request

```
<?php
if (isset($_POST['Name']) &&
    isset($_POST['Message'])) {
    $response =
        array(
            "status" => "OK",
            "info" => array(
                "name" => $_POST['Name'],
                "message" => $_POST['Message']));
} else {
    $response = array("status" => "missingParameter");
}
header("Content-type: application/json");
echo json_encode($response);
?>
```

Example: Asynchronous Contact Form (1)

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="UTF-8">
<title>Asynchronous Contact Form</title>
<script src="https://code.jquery.com/jquery-2.1.3.min.js"></script>
</head>
<body>

<form id="contactForm" onsubmit="submitContactForm(event)">
  <input type="text" name="Name"/>
  <textarea name="Message" cols="50"></textarea>
  <input type="submit" value="Submit!"/>
</form>

<div id="response"></div>

<script>
// script from next slide
</script>

</body>
</html>
```

Example: Asynchronous Contact Form (2)

```
function submitContactForm(event) {  
  
    var contactForm = $('#contactForm');  
    var data = contactForm.serialize();  
  
    var settings = {  
        type: 'POST',  
        data: data,  
        success: function (response) {  
            var feedback =  
                '<p>Thanks for your message, '  
                + response.info.name+ '!</p>'  
                + '<p>Your message: ' + response.info.message + '</p>';  
            $("#response").html(feedback);  
        }  
    };  
  
    // this prevents the page from refreshing  
    event.preventDefault();  
  
    $.ajax('contactFormHandler.php', settings);  
}
```

\$.ajax(url, settings) - Rundown

- **url:**
String with server URL
- **settings:**
Optional settings as object with key/value pairs.
The most commonly used settings are:
 - **type:** type of the request (either **'post'** or **'get'**)
 - **data:** data that are sent (e.g. form data)
 - **dataType:** the result data type that we expect. jQuery automatically guesses if this key is missing.
(e.g. text, xml, html, ...)
 - **success:** callback function that is called once the request is done

.serialize()

- Useful to construct a string containing all form data
- Example:

```
<form id="contactForm" >  
  <input type="text" name="Name"/>  
  <textarea name="Message" cols="50"></textarea>  
  <input type="submit" value="Submit!"/>  
</form>
```



Peter

This is nice!

Submit!

- Result: **Name=Peter&Message=This+is+nice!**

Breakout: Movie Search

Movie Search

Search Movie...

Result:

No result available. Try typing in the search box...

Example Response for

<http://www.omdbapi.com/?t=blue%20velvet>

```
{
  "Title": "Blue Velvet",
  "Year": "1986",
  "Rated": "R",
  "Released": "30 Oct 1986",
  "Runtime": "120 min",
  "Genre": "Crime, Drama, Mystery",
  "Director": "David Lynch",
  ...
  "Metascore": "75",
  "imdbRating": "7.8",
  "imdbVotes": "122,811",
  "imdbID": "tt0090756",
  "Type": "movie",
  "Response": "True"
}
```

Breakout: Tasks

- Download the material from the webpage:
http://www.medien.ifi.lmu.de/lehre/ws1516/mmn/uebung/material/mmn_material07_NF.zip
- In the file `movie-search.html` complete all the TODO entries
- Goal:
 - typing instantly searches the OMDb API
<http://www.omdbapi.com/>
 - The title and year of the result are displayed in the `#result <div>`

Round-up Quiz

1. What does the acronym AJAX stand for?
2. What is an advantage of AJAX?
3. What is an advantage of JSON compared to XML?
4. Describe all elements that appear in this call:

```
$.ajax('https://api.spotify.com/v1/search',{
  method : 'get',
  data : {
    type : 'track',
    q : 'the cars'
  },
  success : function(data){
    /// do something
  }
})
```

Thanks!

What are your questions?

Let's begin with the Assignment!

- Download the assignment sheet
- Start with task 1
- You can collaborate with your neighbor
- Turn in the assignment by December 2nd, 12:00 noon via UniWorX