

jQuery – AJAX-Funktionen

Ingo Köster

Diplom-Informatiker (FH)

Das klassische Web-Modell

Campaign Inclusion Report

Contacts that were...

	Campaign	Status	
<input type="button" value="included"/>	<input type="text" value="Outreach Lunch"/>	<input type="button" value="Attended"/>	AND
<input type="button" value="excluded"/>	<input type="text" value="Holiday Sale Letter"/>	<input type="button" value="none"/>	AND
<input type="button" value="excluded"/>	<input type="text" value="VIP Nurture Track"/>	<input type="button" value="none"/>	AND

Browser

Campaign Inclusion Report

Contacts that were...

	Campaign	Status	
<input type="button" value="included"/>	<input type="text" value="Outreach Lunch"/>	<input type="button" value="Attended"/>	AND
<input type="button" value="excluded"/>	<input type="text" value="Holiday Sale Letter"/>	<input type="button" value="none"/>	AND
<input type="button" value="excluded"/>	<input type="text" value="VIP Nurture Track"/>	<input type="button" value="none"/>	AND

Formular anfordern - GET

Daten ändern - POST

Daten ändern - POST

HTTP-Server



Das klassische Web-Modell - II

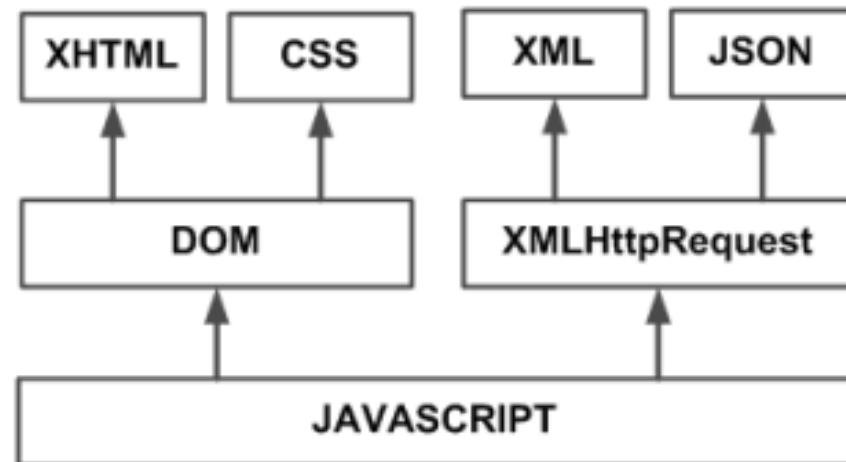
- › Jede neue Anfrage (REQUEST), ganz gleich ob GET oder POST erfordert einen Neuaufbau der Seite (DOM)
- › Es müssen ggf. viele Daten übertragen werden
 - › Z.B. bei komplexem Formular
- › „Web-Anwendungen“ verhalten sich gewöhnungsbedürftig
- › Ziel:
 - › Verhalten einer Desktop-Anwendung angleichen
 - › Nur benötigte Daten übertragen/ändern
 - › Unnötige Server-Round-Trips vermeiden

Was ist AJAX?

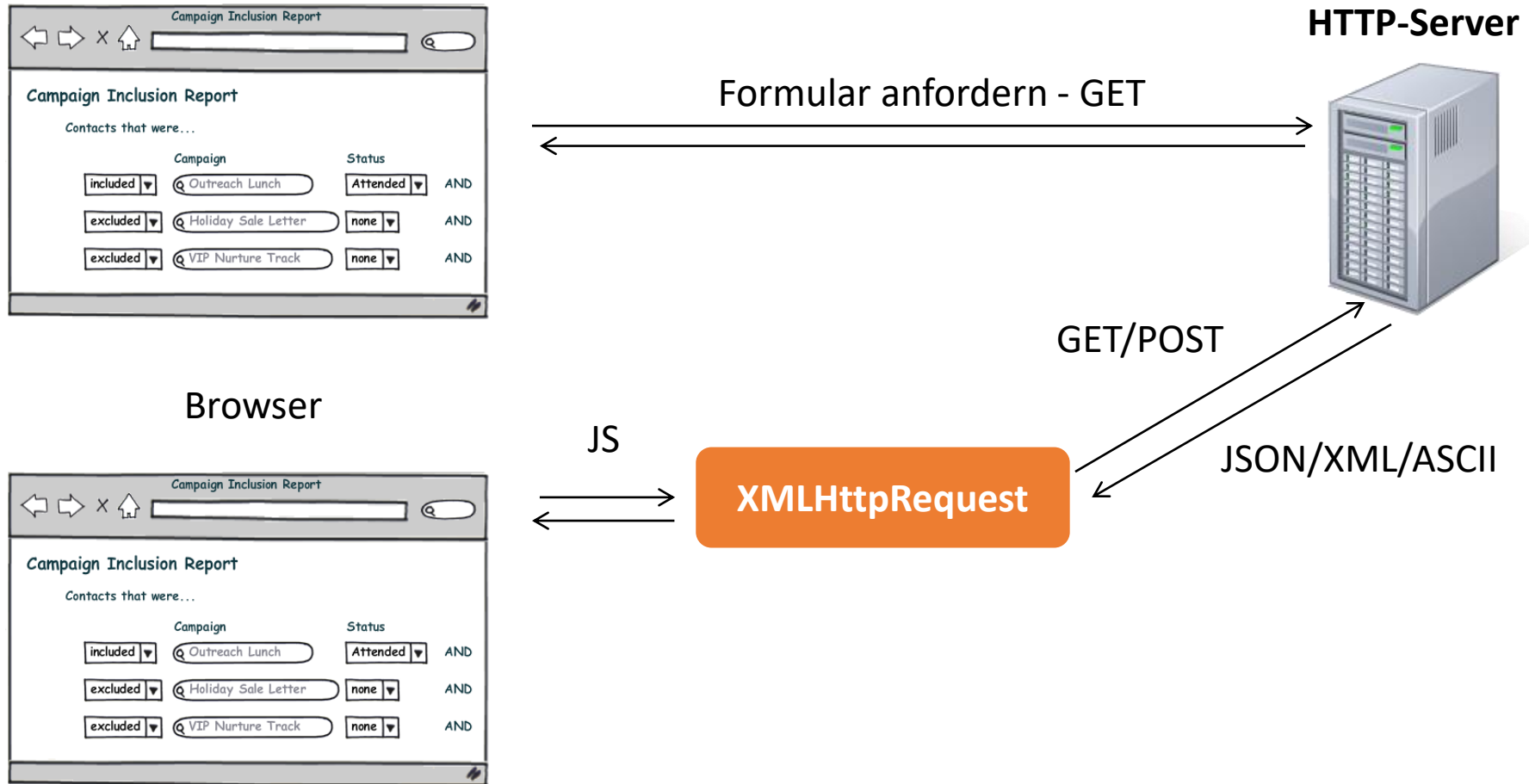
- › AJAX = Asynchronous JavaScript and XML
- › Heute eher Asynchronous JavaScript and JSON
 - › JSON = JavaScript Object Notation
- › Die XML oder JSON Daten kommen oft von einem REST Dienst
 - › REST = Representational State Transfer

Das XMLHttpRequest-Objekt

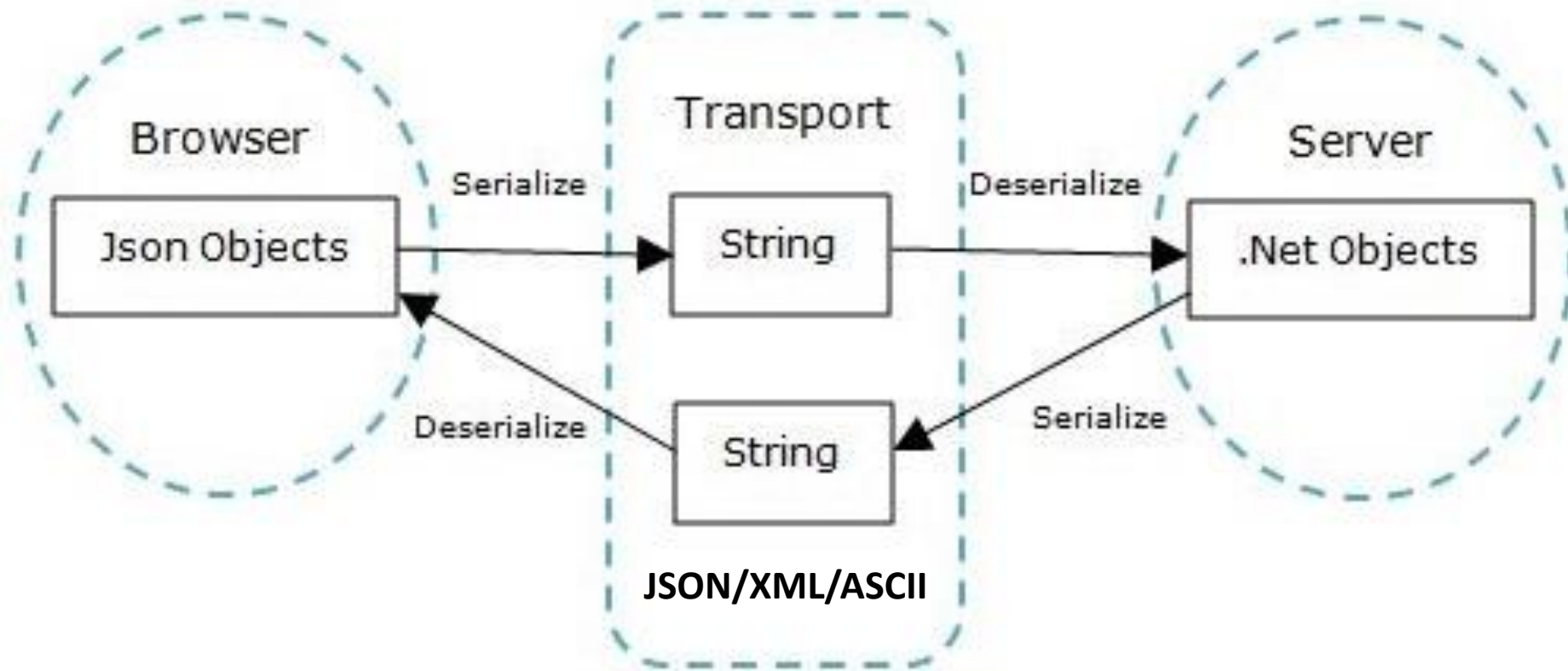
- › Basis ist das XMLHttpRequest-Objekt
 - › Eingeführt 1999 von Microsoft
- › Erlaubt das asynchrone Senden von Anfragen und Verarbeiten von Ergebnissen mittels JavaScript



Das AJAX-Modell



Das AJAX-Modell – Struktur



jQuery – wichtige AJAX-Funktionen

.load()

Also in: [Ajax > Shorthand Methods](#)

Load data from the server and place the returned HTML into the matched element.

jQuery.get()

Also in: [Ajax > Shorthand Methods](#)

Load data from the server using a HTTP GET request.

jQuery.post()

Also in: [Ajax > Shorthand Methods](#)

Load data from the server using a HTTP POST request.

jQuery.getJSON()

Also in: [Ajax > Shorthand Methods](#)

Load JSON-encoded data from the server using a GET HTTP request.

jQuery.ajax()

Also in: [Ajax > Low-Level Interface](#)

Perform an asynchronous HTTP (Ajax) request.

jQuery - .load()

- › Diese Methode ist eine einfach jedoch wenig flexible Variante für einen AJAX-Aufruf
- › Die Methode holt HTML-Text von einem URL und ersetzt den Inhalt eines angegebenen Elements durch den geladenen HTML-Code:

```
$( "#result" ).load( "ajax/test.html" );
```

Der Inhalt des Elements mit der id „result“ wird ersetzt durch den Inhalt des Dokuments im URL der load()-Methode

Für ActionMethoden von MVC
z.B. "/Home/Index" verwenden

jQuery - \$.get()

- › Diese Methode holt JSON-Daten (oder Text) per HTTP-Get vom Server
- › Die Methode erlaubt auch das Senden von Daten im Get-Request:

```
$.get( "test.cgi", { name: "John", time: "2pm" } )  
  .done(function( data ) {  
    alert( "Data Loaded: " + data );  
  });
```

Sendet Daten an das Server-Skript „test.cgi“ und zeigt das Ergebnis in einem alert-Dialog an.

Könnte mit "/Home/Index" an eine Action Methode senden
public ActionResult Index(string name,
string time)

•get(url [,data] [,success] [,dataType])

- › url
 - › Ein String mit dem URL
- › data
 - › Objekt oder ein String mit Daten, die mit dem Request gesendet werden
- › success
 - › Eine Funktion die aufgerufen wird, wenn der Request erfolgreich war
- › dataType
 - › Der Typ der Daten die vom Server gesendet werden
 - › xml, json, script oder html
 - › Default: Wird von jQuery versucht zu ermitteln

jQuery - \$.post()

- › Diese Methode sendet und holt JSON-Daten per HTTP-Post vom Server:

```
$.post( "test.php", { func: "getNameAndTime" }, function( data ) {  
    console.log( data.name ); // John  
    console.log( data.time ); // 2pm  
}, "json");
```

Sendet Daten an das Server-Skript „test.php“ und zeigt das Ergebnis in der JavaScript-Konsole des Browsers an.

jQuery - \$.getJSON()

- › Diese Methode holt JSON-Daten per HTTP-Post vom Server:

```
$.getJSON( "ajax/test.json", function( data ) {  
    var items = [];  
    $.each( data, function( key, val ) {  
        items.push( "<li id='" + key + "'>" + val + "</li>" );  
    });  
  
    $( "<ul/>", {  
        "class": "my-new-list",  
        html: items.join( "" )  
    }).appendTo( "body" );  
});
```

jQuery - \$.ajax()

- › Flexibelste Methode zum Senden und Empfangen von Daten von und zum Server
- › Wird intern von den anderen Funktionen verwendet

```
$.ajax({  
  url: "http://fiddle.jshell.net/favicon.png",  
  beforeSend: function( xhr ) {  
    xhr.overrideMimeType( "text/plain; charset=x-user-defined" );  
  }  
})  
.done(function( data ) {  
  if ( console && console.log ) {  
    console.log( "Sample of data:", data.slice( 0, 100 ) );  
  }  
});
```

jQuery - \$.getScript()

- › Lädt eine JavaScript-Datei und führt diese sofort aus
 - › Kann auch jQuery enthalten
- › Um Skript-Teile nachträglich oder nur bei Bedarf zu laden
- › Geladene Skripte werden im globalen Kontext ausgeführt
- › Haben daher zugriff auf alle Daten und Objekte
 - › DOM, jQuery-Objekt, etc.
- › Beispiel:
 - › `$.getScript('einScript.js');`