

AJAX



18.11.2017 | Jan Huber

Themen

- Grundlagen: **Client-Server-Modell**
- Was ist **AJAX**?
- Das **HTTP(S)-Protokoll**
- Geschichte: Vom **Web 1.0** zum **Web 2.0**
- AJAX anwenden: **Codebeispiele**
- **Web Services** und **API's**?
- **Übungen**

„**AJAX** bezeichnet ein **Konzept** der **Datenübertragung**
zwischen einem Browser und einem Server“

Das Client-Server-Modell



HTML



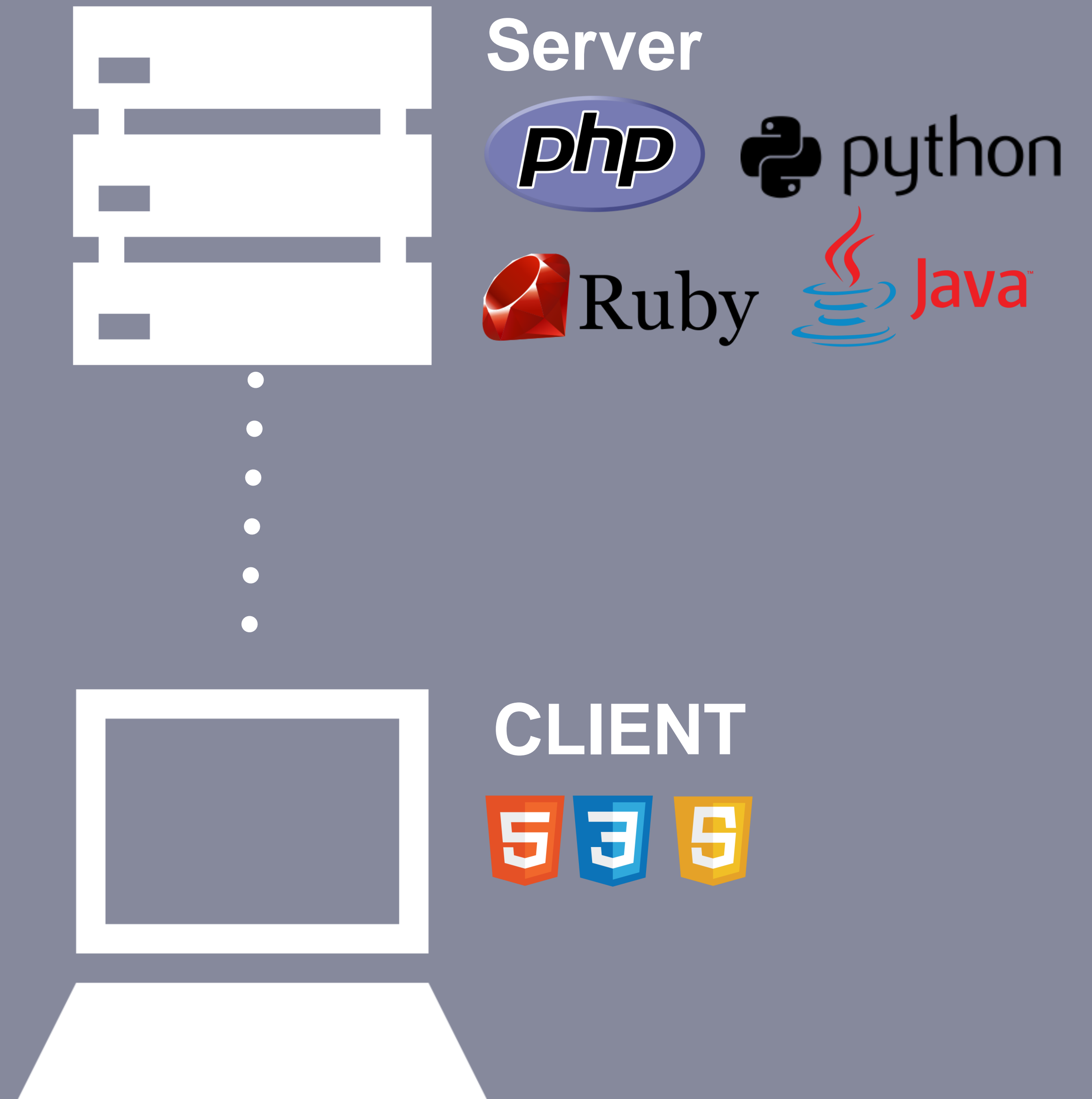
CSS



JavaScript



Das Client-Server-Modell





Von

Nach

Verbindung suchen →

Erleben Sie den
Zauber der
Weihnachtsmärkte.

Mehr erfahren →

BIS ZU
-50%



SBB CFF FFS

Zürich HB → Genf

Mi, 15.11.2017. Abfahrt 07:14. [Ändern](#)

Frühere Verbindungen

Standard-Ansicht

IC

Richtung Genève-Aéroport

07:32

10:18

Gl. 32

2 h 46 min

1.

2.

IC

Richtung Brig

08:02

10:48

Gl. 31

2 h 46 min

1.

2.

ICN

Richtung Genève-Aéroport

08:03

10:45

Gl. 12

2 h 42 min

1.

2.

IC

Richtung Genève-Aéroport

08:32

11:18

Von

Haltestelle, Ort, Sehenswürdigkeit

Nach

Haltestelle, Ort, Sehenswürdigkeit

Datum

<

Mi, 15.11.2017

>

Zeit

07:19

Ab An

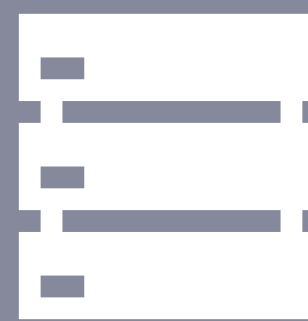
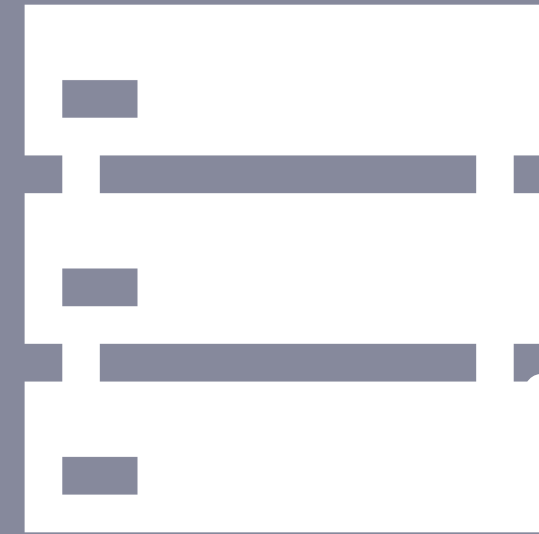
```
graph TD; UI[User Interface] --> DB[(Database)]; DB --> S[Server]; S --> UI;
```

The diagram illustrates a data flow cycle involving three main components: a User Interface (UI), a Database, and a Server. The UI is represented by a laptop icon at the bottom, displaying a simplified version of the SBB CFF FFS search interface. The Database is represented by a stack of four cylinders at the top right. The Server is represented by a stack of three horizontal bars with small vertical lines on the left side. Three large, curved white arrows indicate the flow of data: one arrow points from the UI to the Database, another from the Database to the Server, and a third from the Server back to the UI, forming a continuous loop.

Datenbank



Server



Client



„**AJAX** bezeichnet ein **Konzept** der **Datenübertragung**
zwischen einem Browser und einem Server“

JSON

XML

HTML

...

(Asynchronous)

JavaScript

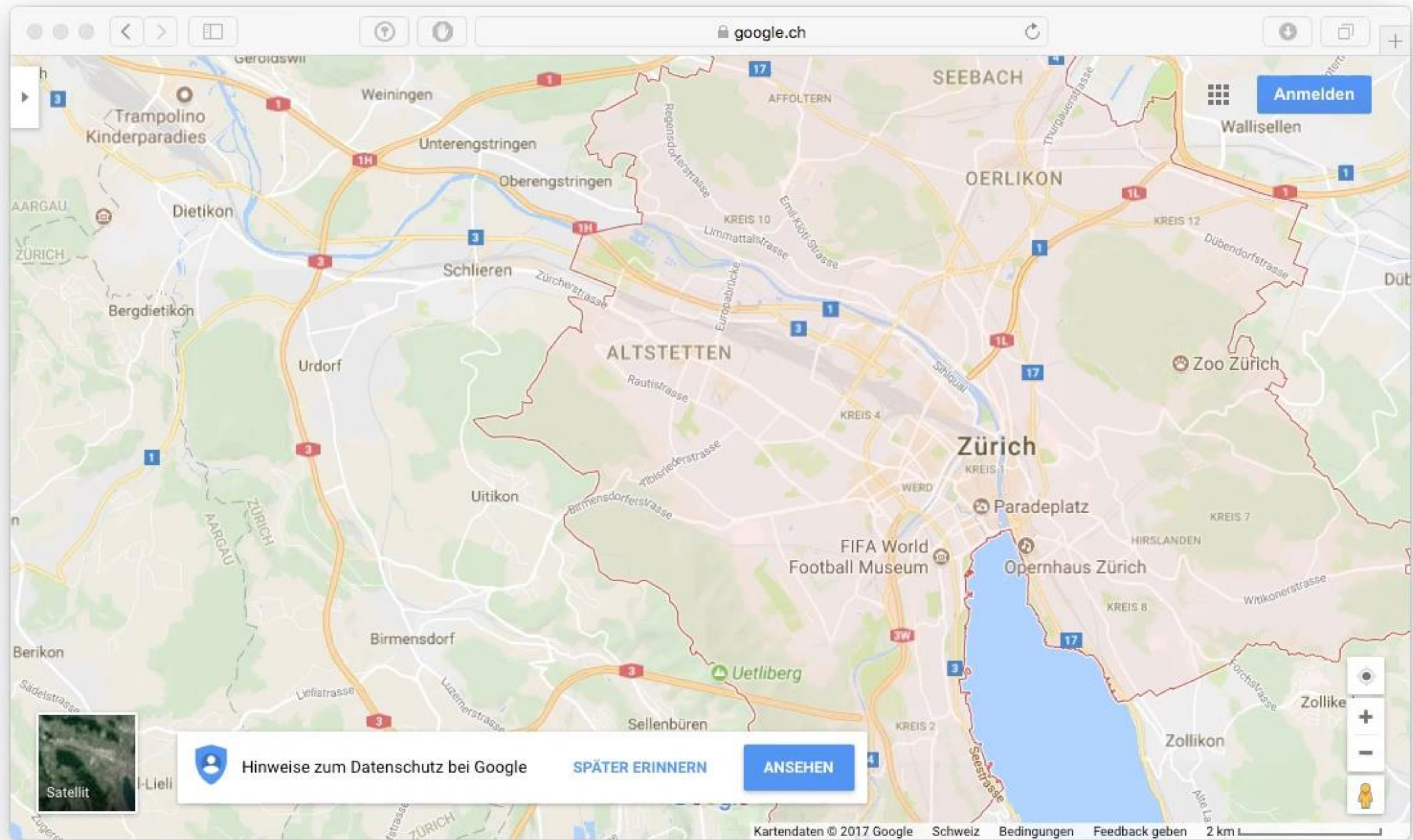
and

XML / JSON / HTML / ...

Mit AJAX kann man...

- ...**Daten** im Hintergrund von einem Webserver **laden**
- ...**Daten** im Hintergrund an einen Webserver **senden**
- ...**Teile einer Webseite austauschen**, ohne die gesamte Seite zu laden

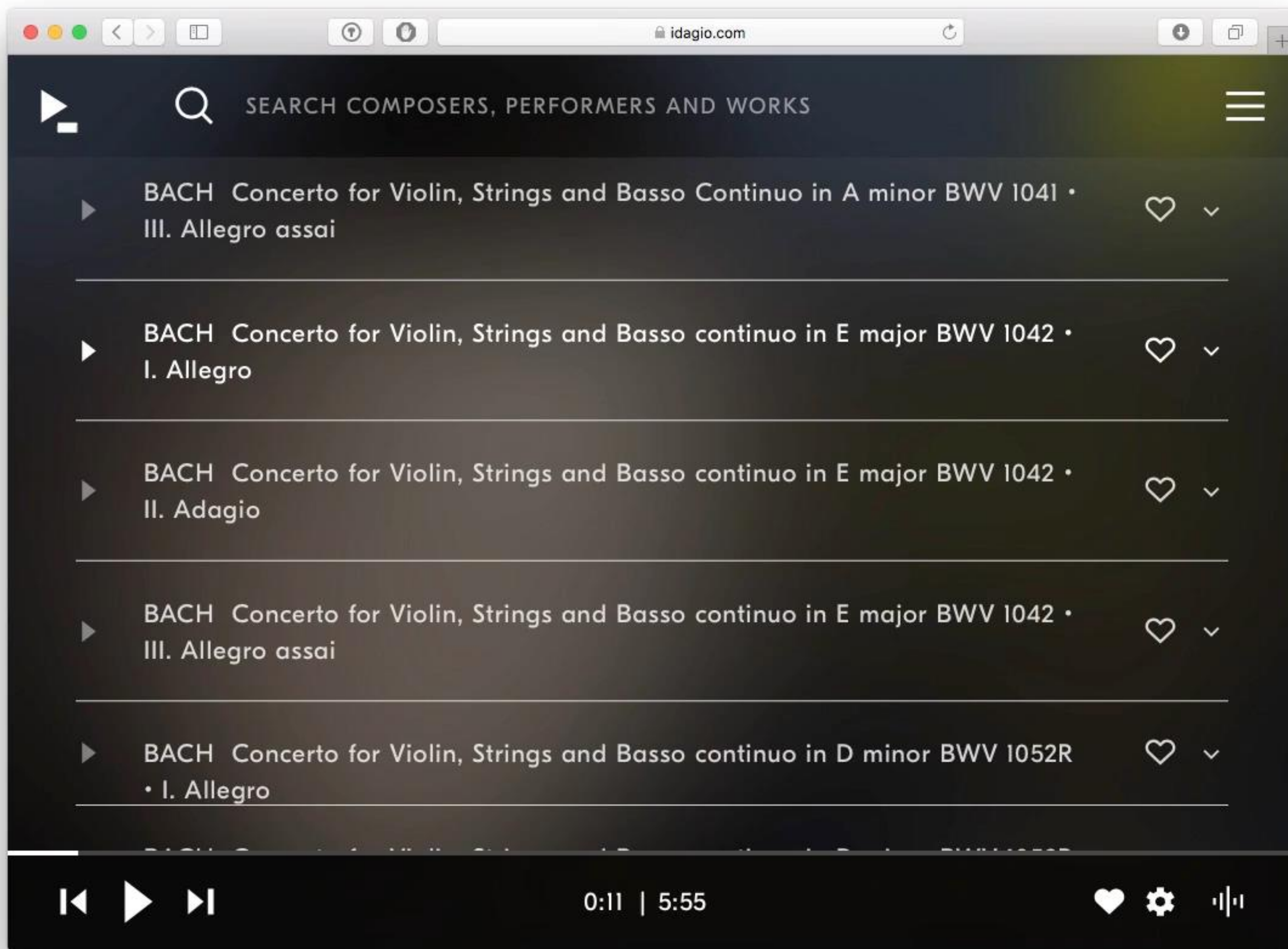
AJAX: **Beispielanwendungen**

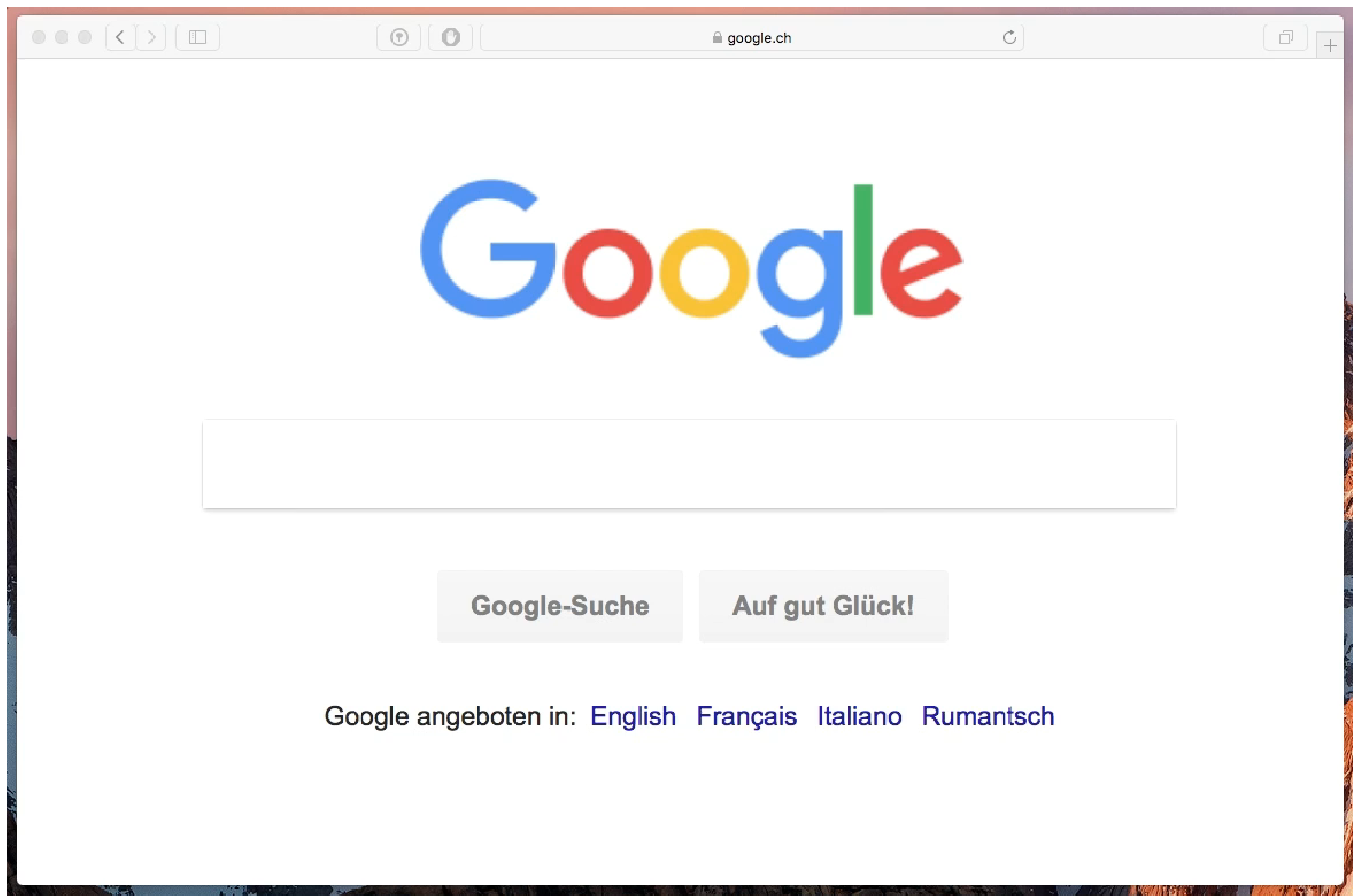


Hinweise zum Datenschutz bei Google

SPÄTER ERINNERN

ANSEHEN





(Traffic mit Chrome beobachten)

Das HTTP(S)-Protokoll

- **H**ypertext **T**ransfer **P**rotocol
- Protokoll zur Datenübertragung im Netz
- Wird oftmals von AJAX verwendet
- ...aber nicht immer!

HTTP(S) Request

Request URL

[https://www.google.ch/complete/search?
query=AJAX&client=Chrome](https://www.google.ch/complete/search?query=AJAX&client=Chrome)

Request Method

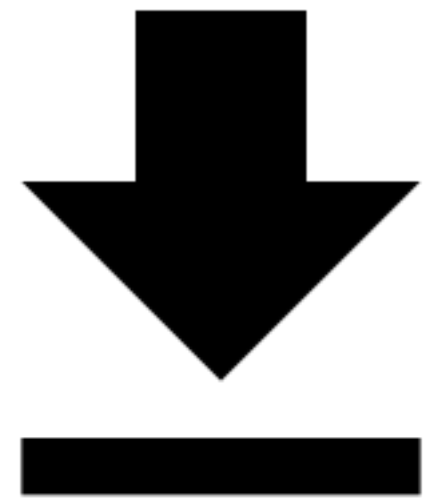
GET

Request Body

-

Die HTTP-Request Typen

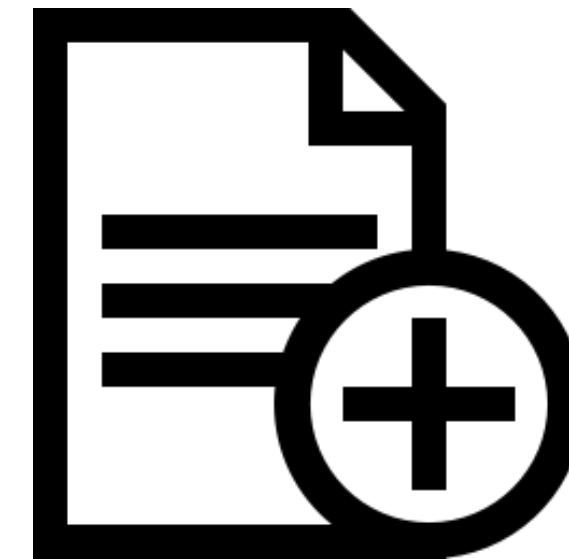
GET



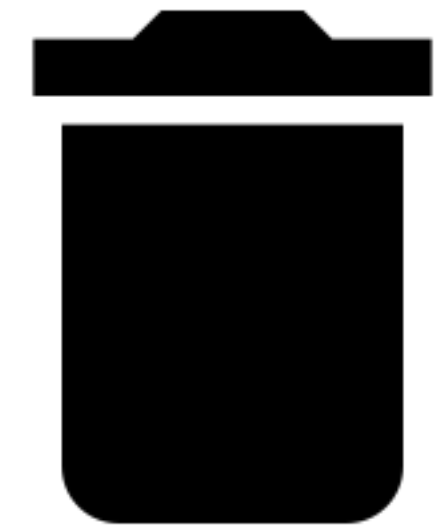
POST



PUT



DELETE



Das HTTP(S)-Protokoll

- **H**ypertext **T**ransfer **P**rotocol
- Protokoll zur Datenübertragung im Netz
- Wird oftmals von AJAX verwendet
- ...aber nicht immer!

HTTP(S) Request

Request URL

[https://www.google.ch/complete/search?
query=AJAX&client=Chrome](https://www.google.ch/complete/search?query=AJAX&client=Chrome)

Request Method

GET

Request Body

-

Das HTTP(S)-Protokoll

- **H**ypertext **T**ransfer **P**rotocol
- Protokoll zur Datenübertragung im Netz
- Wird oftmals von AJAX verwendet
- ...aber nicht immer!

HTTP(S) Response

Response Body

```
{  
  "suggestions":  
    [  
      "ajax jquery",  
      "ajax amsterdam",  
      "ajax tutorial"  
    ]  
}
```

Response Status

200 (OK)

Die HTTP-Response Status

1xx - Information

Die Bearbeitung ist noch nicht abgeschlossen

2xx - Erfolgreich

Antwort des Servers liegt vor

3xx - Umleitung

Client-Aktion erforderlich

4xx - Client-Fehler

Ungültige Anfrage

5xx - Server-Fehler

Fehler auf dem Server



Geschichte

Jahr 2000: Web 1.0

Zeit

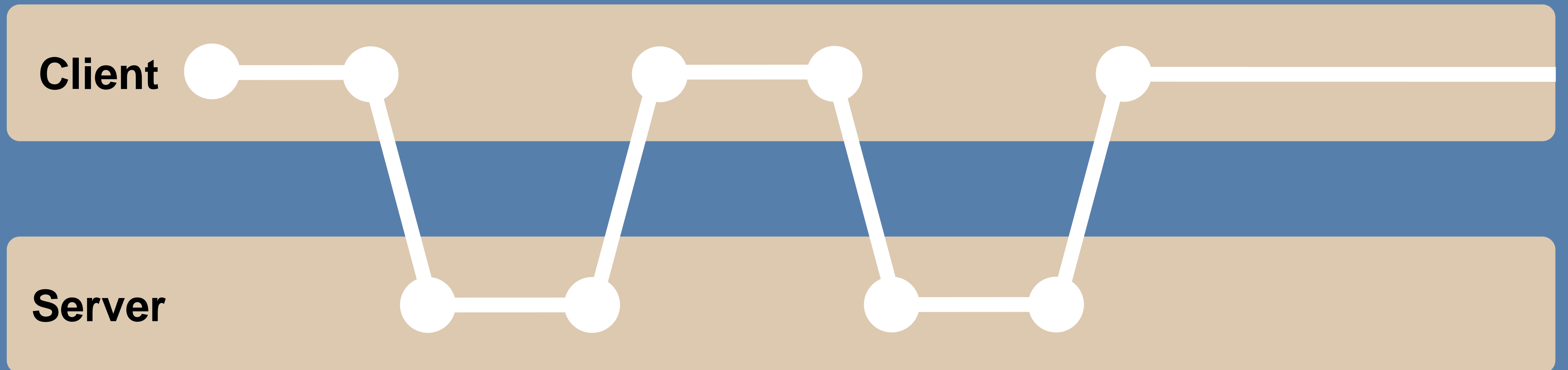


Client

Server

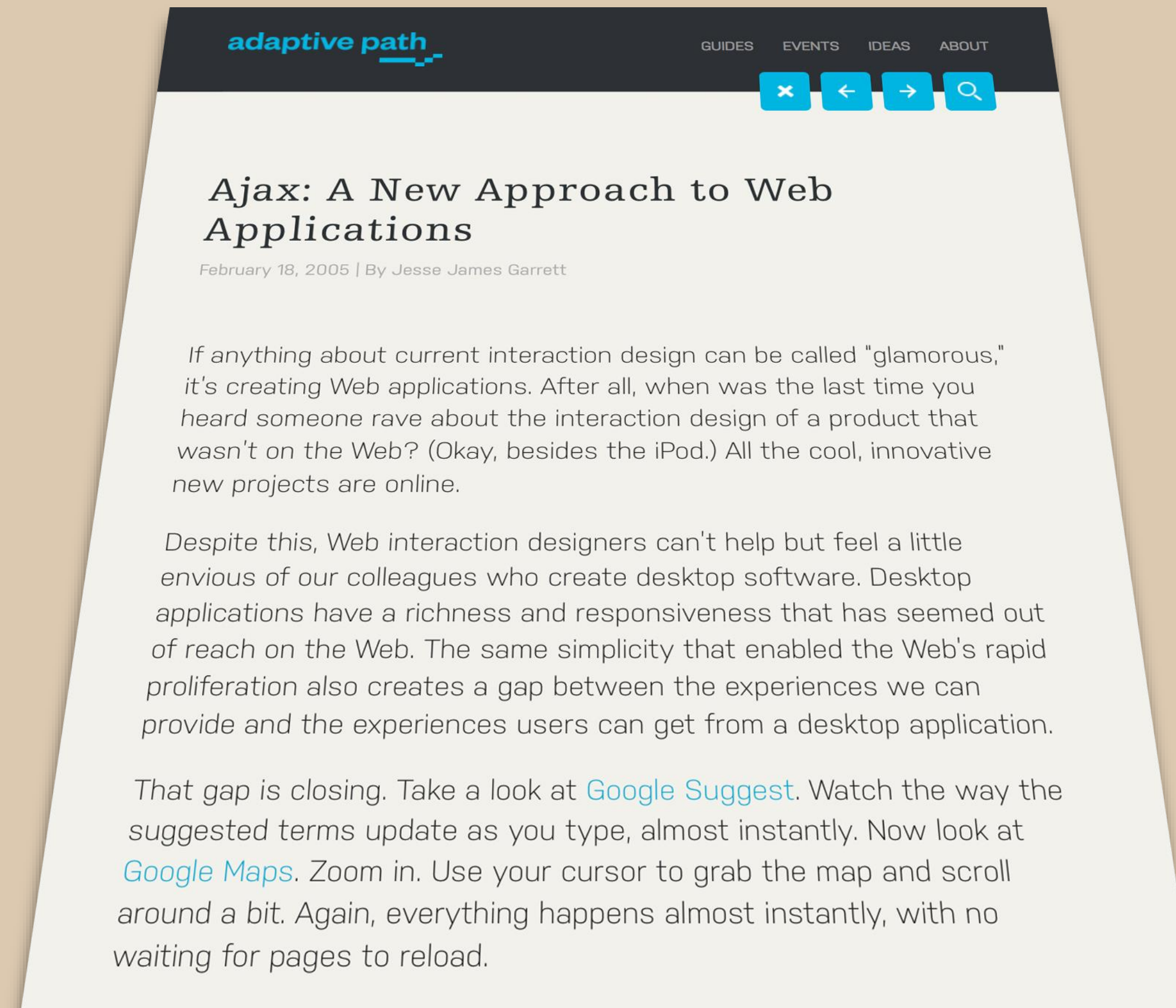
Geschichte

Jahr 2000: Web 1.0



Geschichte

Jahr 2000: Web 1.0



Geschichte

Jahr 2005: „A New Approach to Webapps“

Zeit



Client

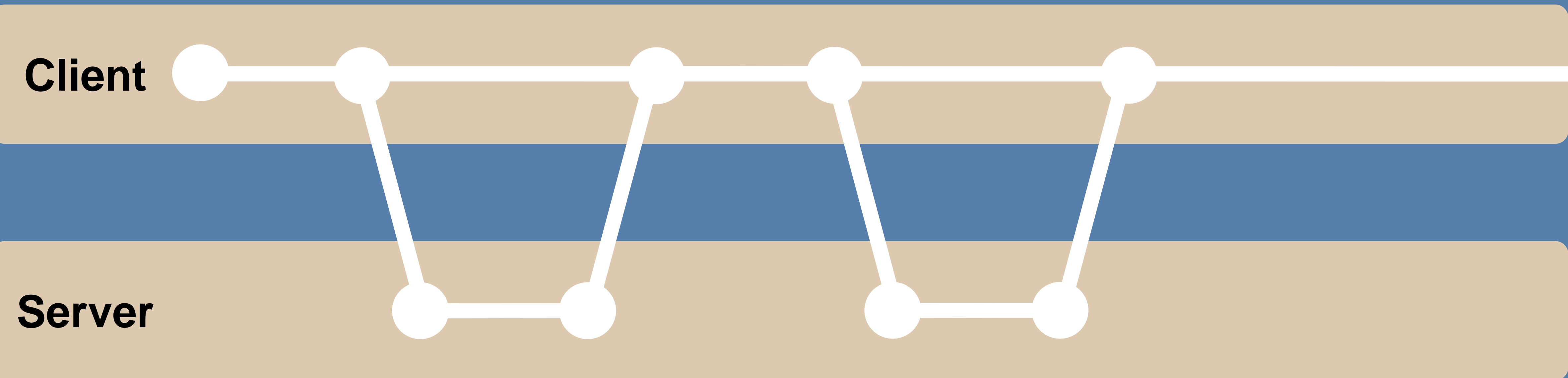


Server



Geschichte

Jahr 2005 - Heute: Web



Geschichte

Jahr 2005 - Heute: Web

Microsoft®

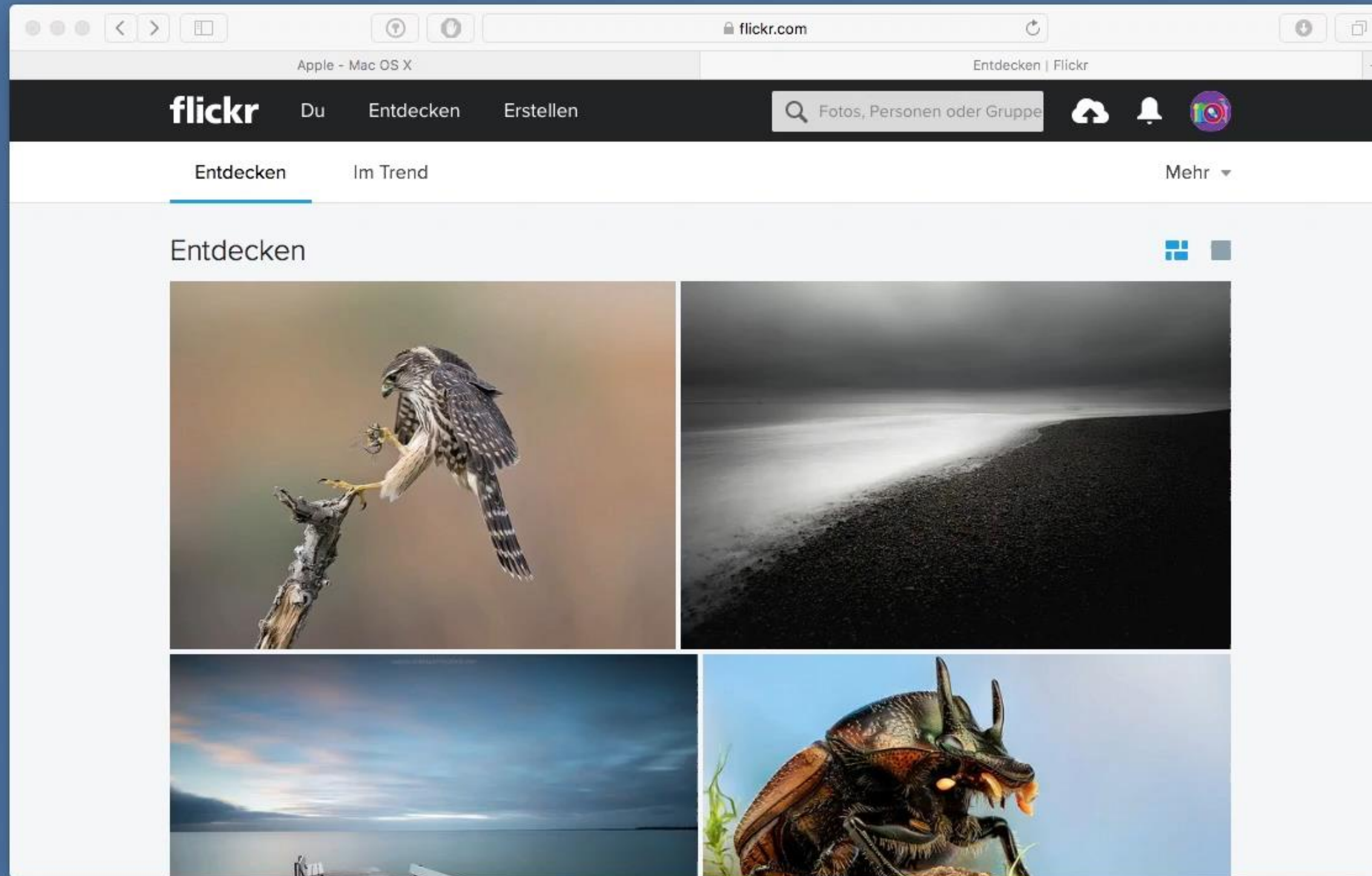
Google

W3C®

YAHOO!®

Geschichte

Jahr 2005 - Heute: Web 2.0



Geschichte

Jahr 2005 - Heute: Web 2.0

Vorteile von AJAX

- Weniger Wartezeiten
- Weniger Traffic
- Kein Browser-Plugin erforderlich
- Von allen neuen Browsern unterstützt



Nachteile von AJAX

- Viel Testaufwand
- „Zurück“-Schaltfläche
- Lesezeichen-Problematik
- Barrierefreiheit
- Polling-Problem



Wie wird AJAX implementiert?

AJAX Beispiel

```
function loadCurrentWeather() {  
  var xhttp = new XMLHttpRequest();  
  xhttp.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200) {  
      showWeatherData(this.responseText);  
    }  
  };  
  xhttp.open("GET", "https://5dayweather.org/api.php?city=Zuerich", true);  
  xhttp.send();  
}
```

```
function showWeatherData(pWeatherData) {  
  var parent = document.getElementById('content');  
  parent.innerHTML = "";  
  var weatherJSON = JSON.parse(pWeatherData);  
  var weatherDescription =  
    '<p>' +  
    'It is ' + weatherJSON.data.skytext +  
    'today in ' + weatherJSON.data.location +  
    ' at a temperature of ' + weatherJSON.data.temperature + '  
    Fahrenheit.' +  
    '</p>';  
  parent.insertAdjacentHTML('beforeend', weatherDescription);  
}
```

Client-Event tritt auf

XMLHttpRequest erstellen

Response vom Server

Request an Server

HTML DOM anpassen

AJAX Beispiel mit jQuery

```
function loadCurrentWeather() {  
  $.get("https://5dayweather.org/api.php?city=Zuerich", function (pData, pStatus) {  
    if (pStatus == "success") {  
      showWeatherData(pData);  
    } else {  
      //error handling  
    }  
  });  
}
```

```
function showWeatherData(pWeatherDataJSON) {  
  var parent = $("content");  
  parent.empty();  
  
  var weatherDescription = $('<p></p>')  
    .text(  
      'It is ' + pWeatherDataJSON.data.skytext +  
      'today in ' + pWeatherDataJSON.data.location +  
      ' at a temperature of ' + pWeatherDataJSON.data.temperature +  
      'Fahrenheit.'  
    );  
  
  parent.append(weatherDescription);  
}
```

Client-Event tritt auf

jQuery-AJAX-Call

Response vom Server

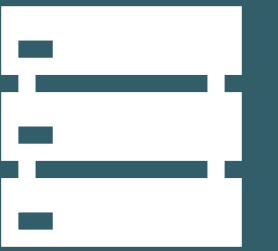
HTML DOM anpassen

Web Services (Web API's)

Was sind Web Services (Web API's)?

- Programmierschnittstellen
- öffentlich oder authentisiert
- Vorsicht: Same-Origin-Policy!

Beispiel API: Foto-Verwaltung



GET	/photos
-----	---------

GET	/photos/{id}
-----	--------------

POST	/photos
------	---------

PUT	/photos/{id}
-----	--------------

DELETE	/photos/{id}
--------	--------------

Beispiel Web Services (Web API's)

- JSON Placeholder
- Open Transport Data
- Open Air Quality

Das Wichtigste zusammengefasst

- Unterscheidung zwischen **Client-** und **Server-Code**
- Die Grundlagen des HTTP(S)-Protokolls
- AJAX ist ein schlechter Name
- AJAX ist ein Konzept
- AJAX hat Vor- und Nachteile
- API's sind Server-Schnittpunkte

Begriffe

- Synchron / Asynchron
- Client / Server
- XML
- HTTP-Request / HTTP-Response
- GET / POST / PUT / DELETE
- XMLHttpRequest
- Same-Origin-Policy

Fragen?

Beispiel-Übung

Übungen
