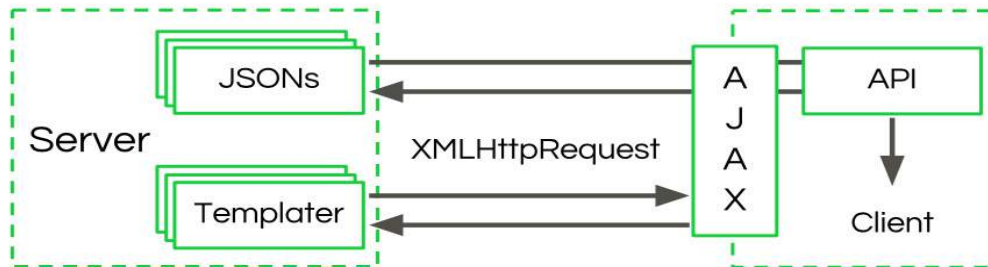


Team 8 – HTML vereinigen / Templating

- 1) Überblick

// zeige gesamtes Flussdiagramm der Kommunikation Server/Client //



- GitHub als Server verwenden
- Daten auf Server: JSON und Templates
- Client fordert Template über AJAX von Server an
- AJAX = Asynchronous Javascript And XML
- Asynchronität → Templates/Daten können ohne neuen Seitenaufruf aktualisiert werden
- Template benötigt spezifische Daten
- Client fordert JSON mit Hilfe der API an
- API kommuniziert mit Server über AJAX
- Verarbeitung durch Programmlogik
- Template wird mit Daten der Programmlogik befüllt

2) Templating

- Templateengine = handlebars.js
 - Erweiterung von moustache //hier ev Bild von moustache einfügen//
- Nachteile:
 - Komplexe Implementierung
 - Längere Ladezeiten bei vielen Templates
- Vorteile:
 - Logik durch eingebaute Helfer → z.B `{{#each}}`, `{{#if}}`
 - Einfache Lesbarkeit, Nutzung
 - Tags können mehrere Werte annehmen

3.1) Funktionsweise

- Programmlogik fordert durch AJAX ein Template *//zeige view_templaterender//*
- falls gefordertes Template eine pre_function besitzt → fordere Daten über API
 - ansonsten leeres Template
- *//zeige view_callback//* callback teilt view-Funktion mit, wann Template vollständig geladen ist

```
46 pre_quizoverview: function (callback) {  
47     //Random Quiz Vorschlag  
48     getQuizView(10);  
49  
50     $(document).on("onQuizView", function (event, data) {  
51         Quizobject.quizes = data; //copy object  
52  
53         if (typeof callback == "function") {  
54             callback(Quizobject);  
55         }  
56     });  
57 },
```

```
12 name: "quizoverview",  
13 sessionobject: {},  
14 render: function (view, callback) {  
15     if (this.availableViews.indexOf(view) > -1) {  
16         var pre_function = this["pre_" + view];  
17         if (typeof pre_function == "function") {  
18             pre_function(function (data) {  
19                 $.get("templates/" + view + ".tpl", function (source) {  
20                     var template = Handlebars.compile(source);  
21                     var templatespaceholder = $("#templatespaceholder");  
22                     templatespaceholder.empty();  
23                     templatespaceholder.append(template(data));  
24                     this.name = view;  
25  
26                     if (typeof callback == "function") {  
27                         callback();  
28                     }  
29                 });  
30             });  
31         } else {  
32             $.get("templates/" + view + ".tpl", function (source) {  
33                 var template = Handlebars.compile(source);  
34                 var templatespaceholder = $("#templatespaceholder");  
35                 templatespaceholder.empty();  
36                 templatespaceholder.append(template());  
37                 this.name = view;  
38  
39                 if (typeof callback == "function") {  
40                     callback();  
41                 }  
42             });  
43         }  
44     }  
45 },
```

3.2) Funktionsweise

//zeige template//

- Template-Datei beinhaltet HTML und handlebar-Tags
- Tags werden mit Daten aus JSON gefüllt //zeige json//
- zwischen statischen Header/Footer eingefügt

```
11 {
12   "quizID":1,
13   "titel":"Das Bundesliga Quiz",
14   "autor":"AntoninoFederico",
15   "datum":"10.11.2016",
16   "spielzahl":67,
17   "bild":"https://raw.githubusercontent.com/th-koeln/wba1-2016/master/bilder/10112016.jpg",
18   "text":"Wie gut kennst du dich mit der Bundesliga aus? Alle Spieler haben eine Chance, die richtige Antwort zu finden. Die richtige Antwort ist: 10.11.2016"
19 },
```

```
32 <!-- Quizspiel -->
33 {{#each quizzes}}
34 <li class="js-collapse-section">
35   <div class="row qo-quiz-summary js-quiz-summary">
36
37     <!-- Headline -->
38     <div class="col-sm-8">
39       <a href="#" class="qo-collapse-link js-collapse-link"><i>
40         class="qo-arrow ico-light-arrow-right"></i>
41       <h2 class="qo-subheadline">{{titel}}</h2></a>
42     </div>
43     <div class="col-sm-4 _hidden-xs">
44       <p class="qo-infos"><strong>{{spielzahl}}</strong> mal gespielt</p>
45     </div>
46   </div>
47
48   <div class="row qo-quiz-details js-collapse-details _same-height-sm">
49     <!-- Quizbild -->
50     <div class="col-sm-6 col-md-4">
51       
52     </div>
53     <!-- End Quizbild -->
54
55     <!-- Quizinfos -->
56     <div class="col-sm-6 col-md-8">
57       <div class="row">
58         <div class="col-sm-12 col-md-10">
59           <p class="qo-text">{{text}}</p>
60         </div>
61       </div>
62
63       <div class="row qo-quizinfo-box _same-height-xs">
64         <div class="col-xs-6 col-sm-4 col-md-6">
65           <p class="qo-time">
66             <time>{{datum}}</time>
67           </p>
68           <p class="qo-author">{{autor}}</p>
69         </div>
70         <div class="col-xs-6 _visible-xs-block qo-counter">
71           <p><strong>{{spielzahl}}</strong> mal gespielt</p>
72         </div>
73         <div class="col-sm-8 col-md-6 _hidden-xs">
74           <a data-view=2 href="#" data-quizid="{{this.quizID}}"
75             class="btn qo-button js-change-view">Jetzt spielen!</a>
76         </div>
77       </div>
78
79       <div class="row qo-quizinfo-box">
80         <div class="col-xs-12 _visible-xs-block">
81           <a data-view=2 href="#" data-quizid="{{this.quizID}}"
82             class="btn qo-button js-change-view">Jetzt spielen!</a>
83         </div>
84       </div>
85     </div>
86     <!-- End Quizinfos -->
87   </div>
88 </li>
89 {{/each}}
90 <!-- End Quizspiel -->
```

3.3) Funktionsweise

//zeige main_render//

- In main.js werden Templates über switch-Funktion gerendert
 - Per default → quizoverview
- restliche Scripte aktiviert (z.B. collapse, flickety)

```
89         default:
90             view.render("quizoverview", function () {
91                 collapse.init();
92                 var flickityConfig = {
93                     // options
94                     cellAlign: 'left',
95                     cellSelector: '.js-carousel-cell',
96                     contain: true,
97                     imagesLoaded: true,
98                     prevNextButtons: false,
99                     setGallerySize: true
100                 };
101                 var $carousel = $('.js-carousel').flickity(flickityConfig);
102                 slideshowNavi.init($carousel);
103                 clickListener();
104             });
105             break;
```