

# SMARTYPANTS

## Das Multiplayer-Webgame

Sommersemester 2017

Mario Koch, Micha Christ, Niklas Janssen, Patrick Brenner, Samantha Stach

### Was ist Smartypants?

Smartypants ist ein webbasiertes Multiplayer Game. Gespielt wird, indem man selbst einen Raum erstellt oder einem bereits vorhandenen beiträgt. Das Spiel bietet die Möglichkeit mit seinen Freunden in einem privaten Raum zu spielen oder einen öffentlichen Raum zu erstellen, um mit Fremden zu spielen. Das Spiel hat mehrere Runden. Die Rundenanzahl kann variabel gewählt werden. In jeder Runde wird allen Spielern eine Frage gestellt, auf welche im Normalfall niemand die richtige Antwort kennen sollte. Nun ist es die Aufgabe des Spielers sich eine möglichst glaubwürdige Antwort zu überlegen. Kann man seine Gegner von seiner Antwort überzeugen, so erhält man Punkte. Weitere Punkte gibt es, wenn man sich selbst für die richtige Antwort entscheidet. Gewinner ist derjenige, welcher am Ende die meisten Punkte gesammelt hat.

### mongoDB

Für unsere Datenbank verwenden wir mongoDB. In dieser Datenbank gibt es folgende Collections:

- Rooms (hier werden alle Spielräume gespeichert. In einem Raum wird zum Beispiel gespeichert, wer sich darin befindet, wie viele Runden es gibt und ein Boolean-Wert ob das Spiel bereits gestartet ist)
- Users (hier werden alle Spieler gespeichert mit Ihren Usernamen, den aktuellen Punkten usw.)
- Questions (hier werden alle Fragen mit ihren richtigen Antworten gespeichert)

### Welche Technologien wurden verwendet?

- MEAN-Stack
  - mongoDB
  - ExpressJS
  - Angular2
  - Node.js
- Socket.io
- Bootstrap

### Node.js

Mit Node.js realisierten wir einen Webserver, der über die passenden Ports auf Kommunikations- und Verbindungsanfragen reagieren kann.

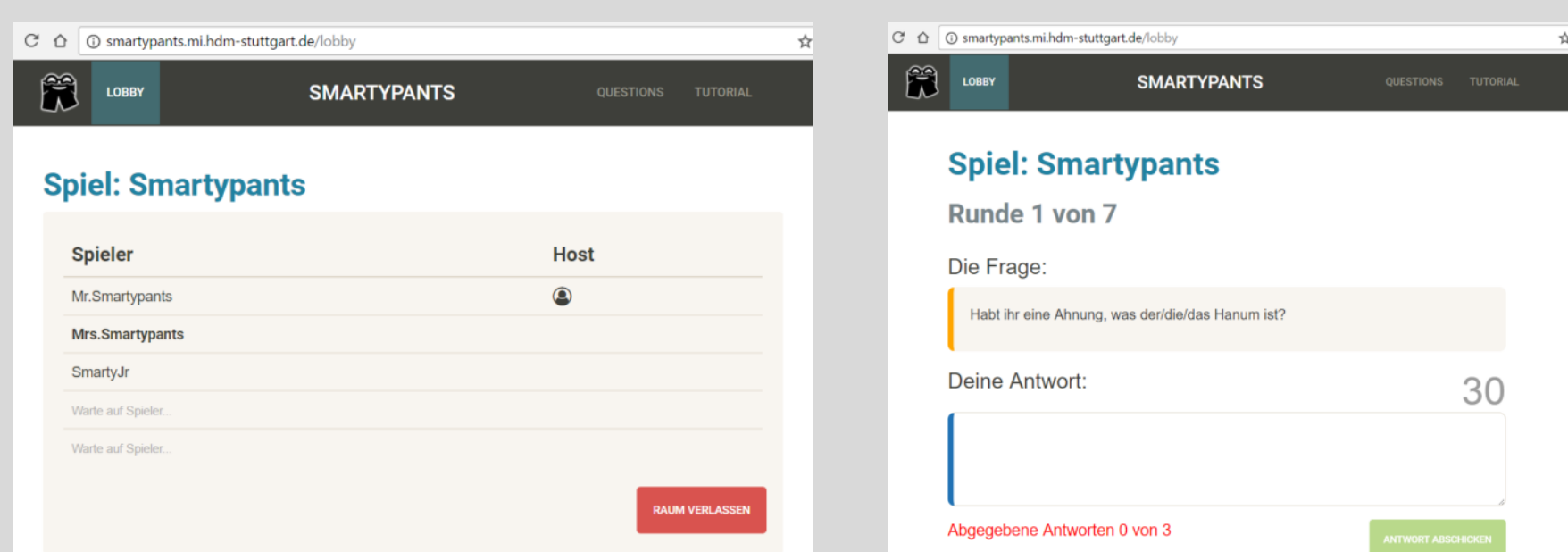
### Socket.io

In unserem Projekt wird Socket.io verwendet um Informationen zwischen Clients auszutauschen. Der Server fungiert hier als Vermittlungsstelle um die Daten, wenn benötigt, an alle User in einem Raum zu schicken. Mit Hilfe von Observables können so Daten zur Laufzeit verändert werden, wie zum Beispiel die Liste der Räume und die Spieler.

```
createRoom(room) {  
  this.socket.emit('createRoom', room);  
  
  return Observable.create((observer: any) => {  
    this.socket.on("roomCreation", (data: any) => observer.next(data) );  
  });  
}
```

### Angular2

Angular2 wird zur Erstellung dynamischer Webseiten verwendet. Es verwendet Components und Services um die unnötige Wiederholung von Code zu vermeiden. Ein Component ist ein Baustein. Durch die Verknüpfung der Components mit einem Router können diese über eine URL erreicht werden. Für eine dynamische Darstellung können Components ineinander verschachtelt werden, um so ständiges weiterleiten und neu laden zu vermeiden. Diese Funktionalität haben wir uns zu Nutzen gemacht. Während des Spiels müssen ständig neue Elemente angezeigt und die Darstellung verändert werden. Statt hier jedes Mal eine neue Seite zu laden, können essentielle Abschnitte dynamisch aus- und eingeblendet werden.



In der Abbildung ist zu sehen, dass es zwar nach zwei unterschiedlichen Seiten aussieht, tatsächlich haben aber beide die gleiche URL.

### Bootstrap

Wir haben Bootstrap verwendet, um das Design unserer Seite zu erstellen und dafür zu sorgen, dass sie responsive ist. Durch das Grid-System von Bootstrap können die Inhalte horizontal leicht angeordnet werden und es kann definiert werden, wie sich diese Inhalte von der Größe her anpassen sollen, falls z.B. zu einer mobilen Ansicht, wie auf einem Tablet oder Smartphone, gewechselt wird.

```
<div class="row">  
  <div class="col-md-4 col-sm-4">  
    <h3>250 einzigartige Fragen!</h3>  
    <p>  
      Bei so gut wie allen Fragen wird keiner die richtige Antwort wissen.  
    </p>  
  </div>  
  <div class="col-md-4 col-sm-4">  
    <h3>Variable Spiellänge!</h3>  
    <p>  
      Je nachdem wie viel Zeit du hast, lässt sich die Rundenlänge dynamisch einstellen.  
    </p>  
  </div>  
  <div class="col-md-4 col-sm-4">  
    <h3>Lade deine Freunde ein!</h3>  
    <p>  
      Smartypants macht mehr Spaß mit deinen Freunden, also lade sie zu einem privaten Spiel ein.  
    </p>  
  </div>  
</div>
```