Karlsruher Institut für Technologie

Agiler Publikationsprozess

Praktikum Werkzeuge für Agile Modellierung

Anna Ostrovskaya | 24. Juli 2019

FAKULTÄT FÜR INFORMATIK / INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION

Outline/Gliederung

Motivation

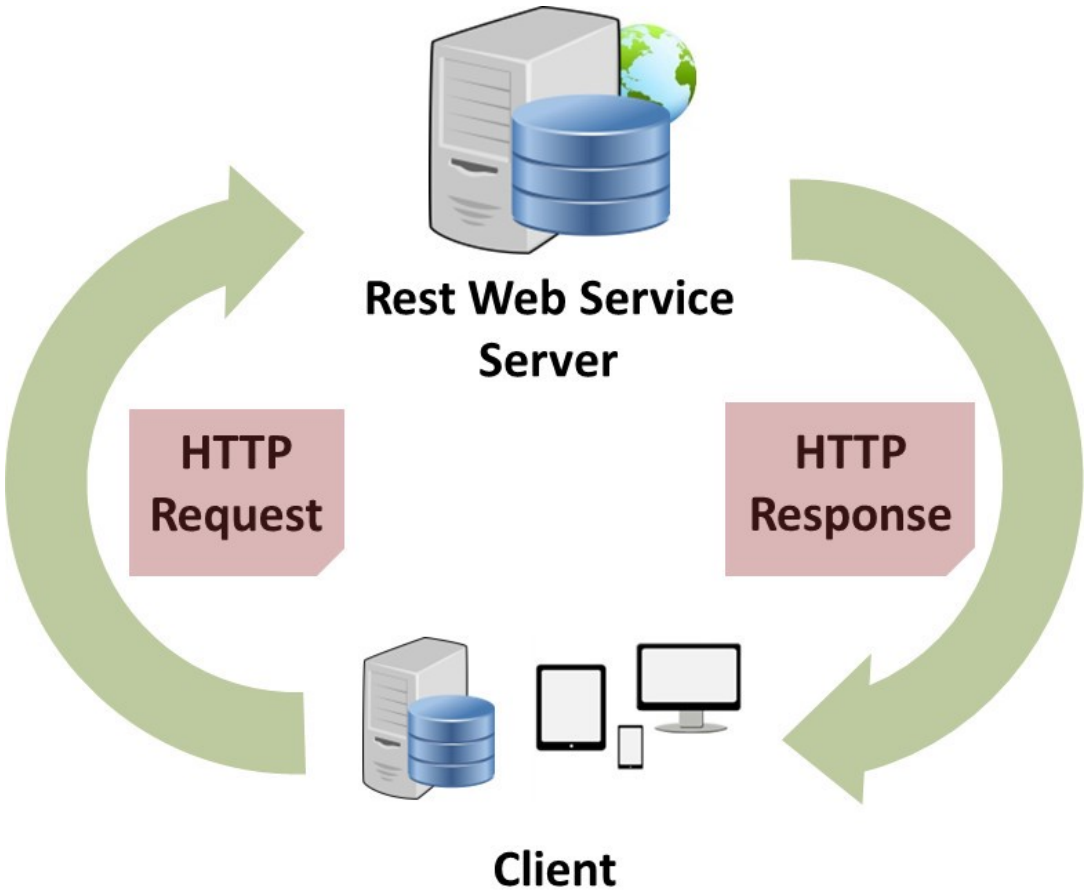
- Forschungsprozesse mit Reviews sind oft schwergewichtig, mit langen Iterationen, intransparent.

- Webbasierte agile Lösung für transparenten Publikationsprozess von wissenschaftliche Papiere

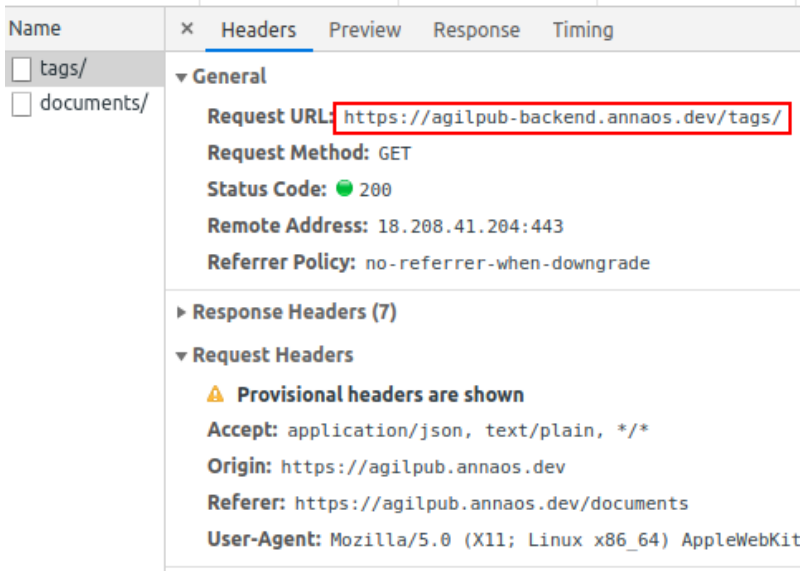
Features

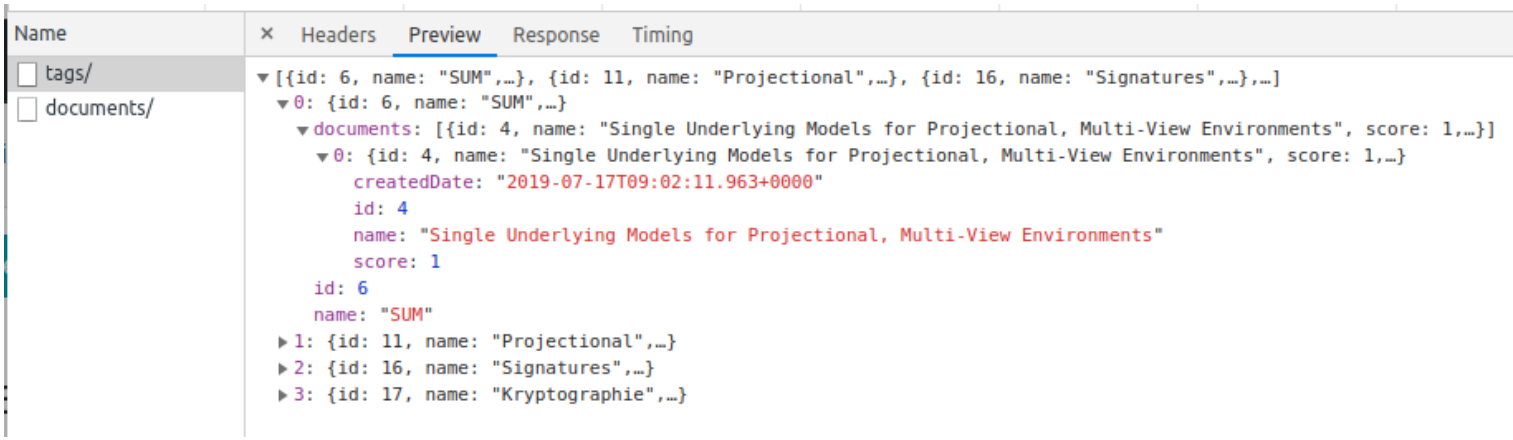
- Vereinfachter Registrierung/Login-Prozess
- Hochladen von Artikeln
- Versionierung von Artikeln
- Personalisierte Kommentare
- Bewertungssystem für Artikeln
- Unterstützung von Tagging

Grundlagen einer webbasierten Anwendung

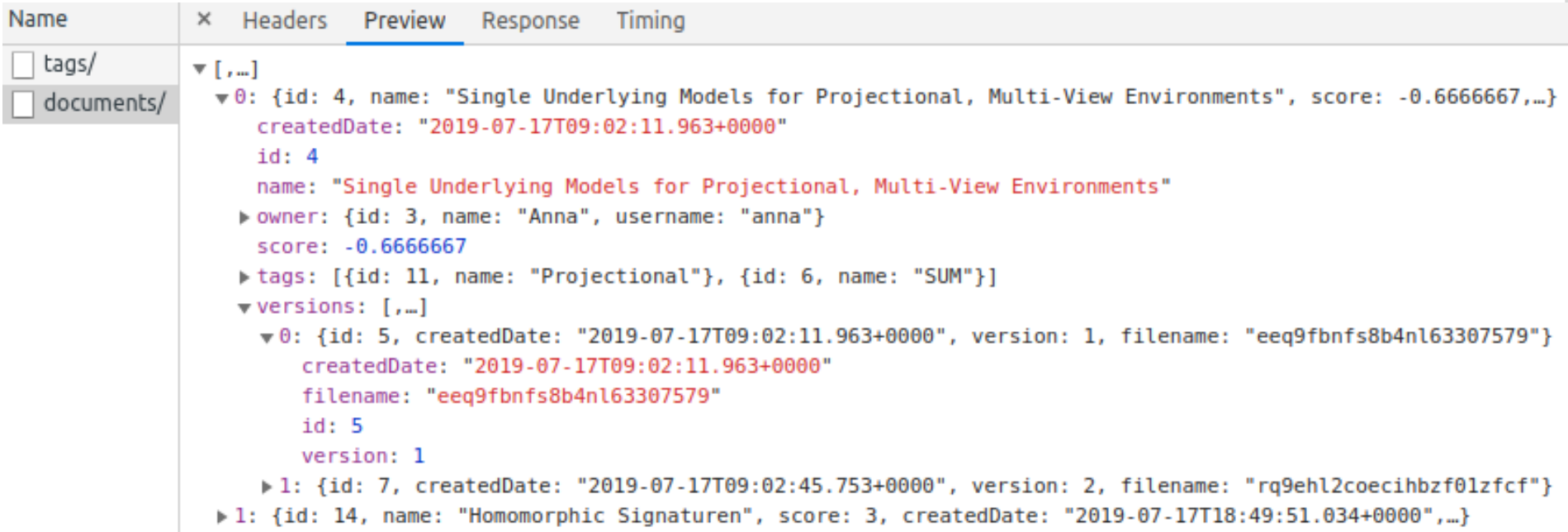


Wie funktioniert es?





Wie funktioniert es?



Technologien

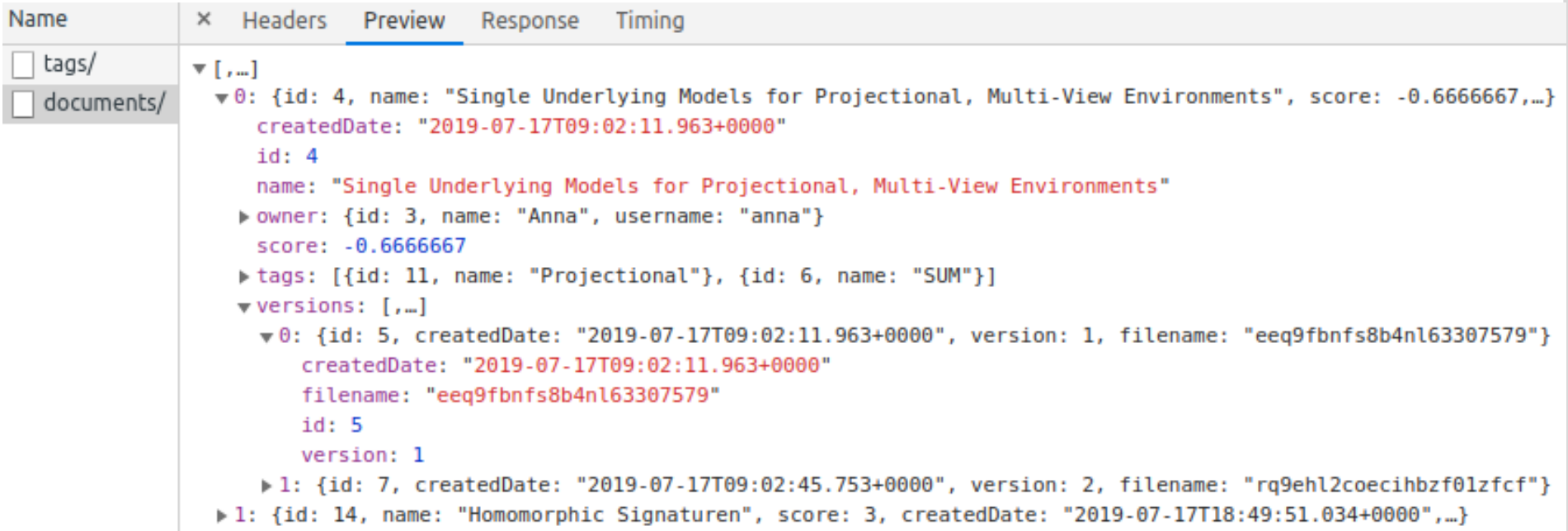
- Backend
  - Java Spring Boot Rest Service
  - In-memory H2 Datenbank (6 Tabellen)
  - Lombok
- Frontend
  - Angular (10 Komponenten)
  - ng2-file-upload
  - ng2-pdf-viewer PDF Darstellung
  - Rangy (Bibliothek für Textselektion)

Herausforderungen

- Übergabe von verknüpften Daten: *JsonIgnore* und *JsonIgnoreProperties*

```java
@Entity
@Table
@Data
public class Document {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long id;
    private String name;
    private float score;
    private final Date createdDate;

    @ManyToOne(fetch=FetchType.EAGER)
    @JoinColumn(name="OWNER_ID")
    @JsonIgnoreProperties({"files", "comments", "scores"})
    private final User owner;

    @ManyToMany(fetch = FetchType.LAZY)
    @JoinTable(
            name="DOCUMENT_TAG",
            joinColumns=@JoinColumn(name="DOCUMENT_ID"),
            inverseJoinColumns=@JoinColumn(name="TAG_ID"))
    @JsonIgnoreProperties("documents")
    private Set<Tag> tags = new HashSet<>();

    @OneToMany(mappedBy = "document", fetch=FetchType.LAZY)
    @JsonIgnoreProperties({"document", "comments"})
    private List<DocumentVersion> versions;

    @OneToMany(mappedBy = "document")
    @JsonIgnore
    private List<Score> scores;
```

Herausforderungen



Herausforderungen

- Unifizierung von Datumdarstellung

```
<td>{{ version.createdDate | date:'dd.MM.yyyy \'um\' HH:mm:ss'}}</td>
<td>{{ version.createdDate | createdDateFormat }}</td>
```

```
@Pipe({
  name: 'createdDateFormat',
})
export class createdDateFormatPipe implements PipeTransform {
  transform(value: Date) {
    registerLocaleData(localeDe);
    let datePipe = new DatePipe( locale: "de-DE");
    return datePipe.transform(value,  format: 'dd.MM.yyyy \'um\' HH:mm:ss');
  }
}
```

Herausforderungen

- Übergabe PDF-Datei per REST-Service
  - Backend: FileSystemStorageService
    - erhält MultipartFile, Speicherung im Dateisystem
    - Übertragung per HTTP-Response als Content Type application/pdf
  - Frontend
    - Upload
    - Empfang per HTTP-Response als Content Type application/pdf

Herausforderungen

- Speichern von Kommentare
  mit **Rangy**: cross-browser JavaScript range and selection library

```
14/1/3:62,14/1/3:66{f910c7cc}
start     ,end       {checksum}

14/1/3    :62
node_path:index
```

Live-Demo

Fazit

DONE

- REST-Service Backend
- Single-Site-Webanwendung mit Basisfunktionalität

TODO

- Datenbank-Anbindung
- Login- / Registrierungsprozess
- Backend-Autorisierungsprozess
- Performance
- Dateigröße beschränken
- Überlappende Kommentare ermöglichen