

# JAVA I PROJEKT SERVICE NEWS

Benjamin Ehnes (Dev Lead)

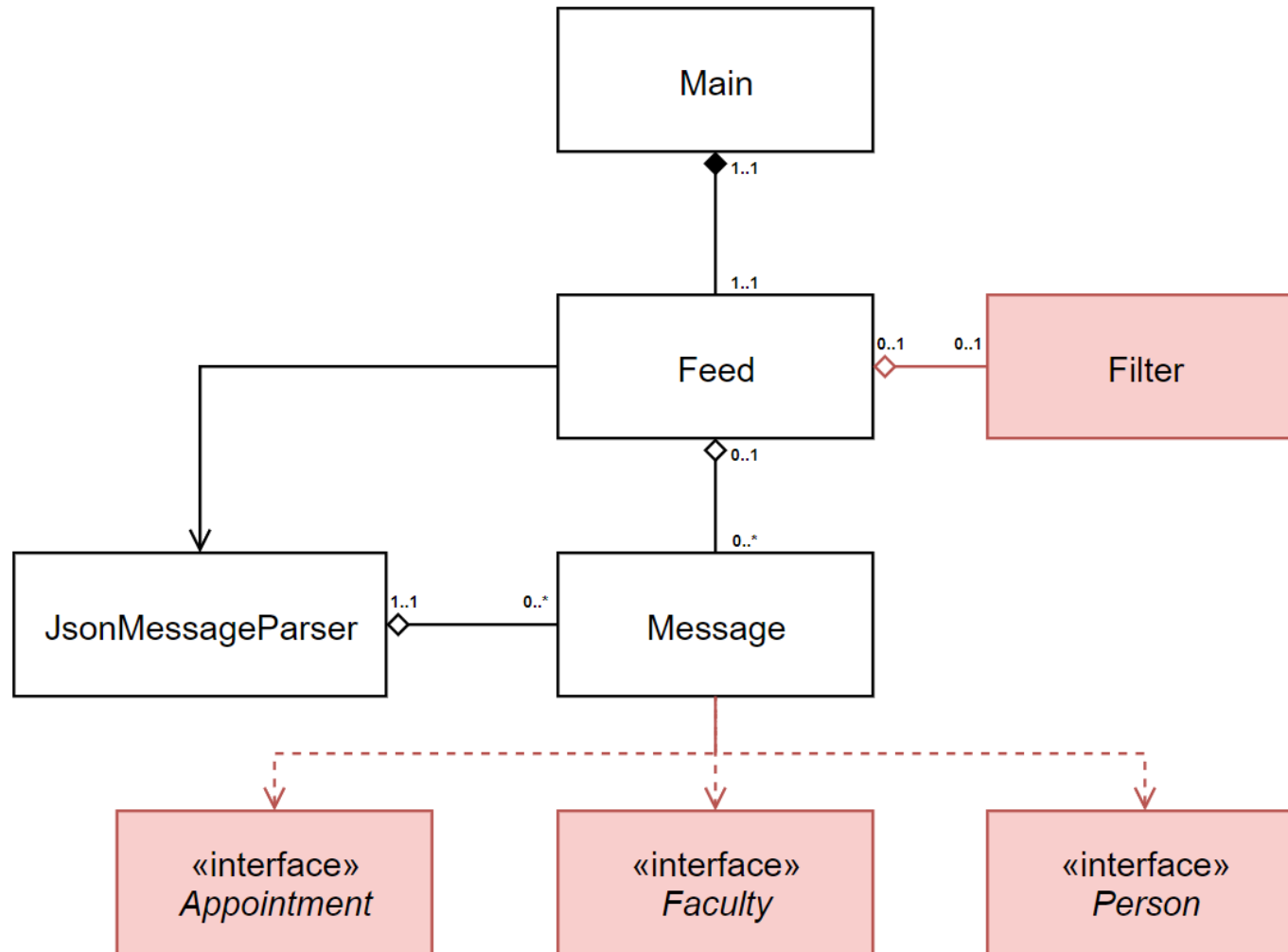
Antonia Geschke (PO)

Lucian Gerasch - Celina Ludwigs - Lisa Sluka

# Agenda

1. Klassendiagramm
2. Dateiformat
3. class: Message
4. class: JsonMessageParser
5. class: Feed
6. next steps

# Klassendiagramm



# Dateiformat

```
▶ {
  "status": "done",
  "messages": [
    {
      "author": "34857238958732758295",
      "title": "Tolle Neuigkeiten",
      "description": "Wir haben tolle Neuigkeiten",
      "url": "https://www.fh-erfurt.de/news/detailansicht/fh-erfurt-macht-mobil",
      "urlToImage": "https://www.fh-erfurt.de/news/detailansicht/fh-erfurt-macht-mobil",
      "publishedAt": "2022-12-03 10:30",
      "content": "Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor
invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo
dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore
et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea
rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.",
      "topic": "Veranstaltungen",
      "appointmentName": "Tag der Mobilität",
      "appointmentDateTime": "2022-07-30 10:30"
    }
  ]
}
```

- ▶ inspiriert von newsapi.org
- ▶ gute Alternative zum XML/RSS
- ▶ sehr gut von Menschen lesbar und von Maschinen serialisierbar/deserialisierbar
- ▶ notwendige Felder: Autor, Titel und Content

# Message

```
14  */
15  public class Message {
16      private String author; // Interface?! // String oder Id?!
17      private String title;
18      private String description;
19      private String url;
20      private String urlToImage;
21      private LocalDateTime publishedAt;
22      private String content;
23
24      private String topic;
25      private ArrayList<Faculty> faculties; // Interface?!
26
27      private String appointmentName; // AppointmentService: createAppointment(String name, LocalDateTime appointmentDateTime);
28      private Optional<LocalDateTime> appointmentDateTime;
29
30      public Message(String author,
31                     String title,
32                     String description,
33                     String url,
34                     String urlToImage,
35                     LocalDateTime publishedAt,
36                     String content,
37                     String topic,
38                     ArrayList<Faculty> faculties,
39                     String appointmentName,
40                     LocalDateTime appointmentDateTime) {
41          this.author = author;
42          this.title = title;
43          this.description = description;
44          this.url = url;
45          this.urlToImage = urlToImage;
```

## Class Message

- DataHolder
- beinhaltet notwendige Daten
- genaues Abbild einer Nachricht aus dem JSON File

## Setter und getter für Membervariablen

## hasAppointmentAssociated()

- Prüft ob Termin mit Message verbunden ist

## Geplanter Builder:

- Entwurfsmuster
- bietet Konstruktionssicherheit
- höhere Code Lesbarkeit

# JsonMessageParser

```
public static Optional<Message> parseSingleMessage(JSONObject jsonObject) {
```

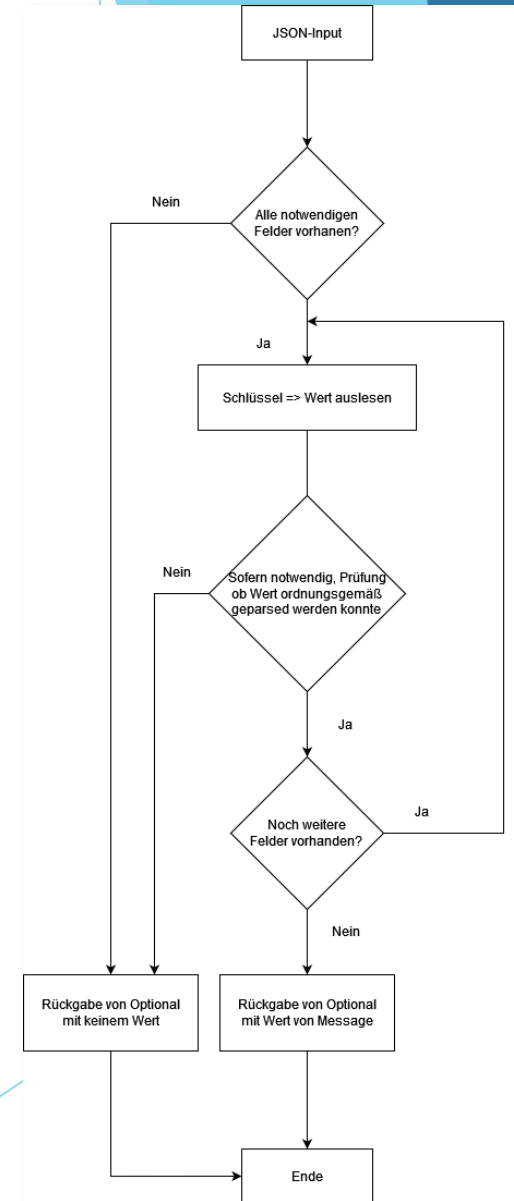
- ▶ Überprüft, ob in angegebenen JSON Objekt alle notwendigen Schlüsselwörter vorhanden sind

```
public static ArrayList<Message> parseMessages(JSONArray messageArray) {
```

- ▶ Ruft für jedes Objekt des JSON Arrays die parseSingleMessage() Methode auf
- ▶ Einzelne Objekte werden in Array Liste eingefügt

```
public static ArrayList<Message> parseJsonFile(JSONObject jsonFile) {
```

- ▶ Ruft für alle Messages des Files die parseMessage() Methode auf



# Feed

- **Feed Konstruktor**
  - Zuweisung einer URL
- **fetch()**
  - Einlesen und Verarbeiten der URL
  - Erzeugung eines neuen JSONObjects
- **buildFeed()**
  - Ausgabe der zuvor geparsten Liste
- Abfrage der Nachricht von Discord

```
16 public class Feed {
17     List<Message> entries = new ArrayList<>();
18     URL url = null;
19
20     public Feed(URL url) { this.url = url; }
21
22
23
24     public void fetch() throws IOException {
25         StringBuilder stringBuilder = new StringBuilder();
26         String line;
27
28         InputStream inputStream = url.openStream();
29
30         try {
31             BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream));
32             while ((line = reader.readLine()) != null) {
33                 stringBuilder.append(line).append(System.LineSeparator());
34             }
35         } finally {
36             inputStream.close();
37         }
38
39         String content = stringBuilder.toString();
40
41         // json file could fail
42         JSONObject jsonObject = new JSONObject(content);
43
44         entries = JsonMessageParser.parseJsonFile(jsonObject);
45     }
46
47     public void buildFeed() {
48         System.out.println(entries);
49     }
50 }
```

```
35 Feed feed = new Feed(new URL( spec: "https://cdn.discordapp.com/attachments/906109518142918688/916683041127141386/news.json"));
36 feed.fetch();
37 feed.buildFeed();
```

# next Steps

- ▶ Einbinden der anderen Services mittels Interfaces
- ▶ implementieren der Filter Funktionalität
- ▶ Code Qualität anheben
- ▶ ausweiten von Unit Tests



The background features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. These shapes are primarily located on the right side of the slide, creating a modern, dynamic feel.

# Vielen Dank!

Noch Fragen?