

Zwischenprojekt RailTraffic

1. Auftrag

Die SBB beauftragt Sie ein Programm für ihre Kunden zu schreiben, welche die Bahnverkehrsmeldungen benutzerfreundlich anzeigt. Dabei soll das Programm automatisch nach neuen Meldungen suchen und diese dem Benutzer darstellen. Zudem sollen neu eintretende Meldungen durch ein Benachrichtigungsfeld kurzzeitig dargestellt werden.

Das Programm soll dabei einen tabellarischen Aufbau vorweisen. In jeder Zeile soll eine Meldung in Kurzform (Zeit, Datum, Titel) dargestellt werden:

14:38	2018-06-18	Restriction: Wettingen – Baden
12:24	2018-06-17	End of announcement: Wettingen
11:12	2018-06-15	End of announcement: Wettingen
07:31	2018-06-15	Restriction: France
14:10	2018-06-13	Restriction: La Plaine
21:30	2018-06-12	Construction work: Between Loc...

Der Benutzer soll auch die Möglichkeit haben, die genaueren Informationen (Beschreibung) nach Auswahl der Meldung anzufordern. Durch eine Einstellung sollen die Anzahl zu anzeigenden Meldungen veränderbar sein.

Für das Projekt bietet die SBB Ihnen eine Web-Schnittstelle (Web API) an um die Daten der Verkehrsmeldungen abzufragen: <https://data.sbb.ch/explore/dataset/rail-traffic-information/api/>

Ihnen wird aufgetragen, das Programm einfach und benutzerfreundlich zu gestalten.

[illegible]

3. JSON-Format

Die JavaScript Object Notation, kurz **JSON**, ist ein kompaktes Datenformat in einer einfach Textform zum Zweck des Datenaustauschs zwischen Anwendungen. In Java können Sie das JSON-Format mit der externen Programmbibliothek *gson* in Java-Elemente umwandeln:

<https://github.com/google/gson>

Die JAR-Datei können Sie unter folgendem Link beziehen:

<https://repo1.maven.org/maven2/com/google/code/gson/gson/2.8.2/gson-2.8.2.jar>

Ein kurze Einführung in gson finden Sie hier:

<https://futurestud.io/tutorials/gson-getting-started-with-java-json-serialization-deserialization>

Um den *InputStreamReader* aus dem Webzugriff-Beispiel mit gson zu verwenden, kann folgendes Code-Beispiel betrachtet werden:

```
...
Gson gson = new Gson();
Elements elements = gson.fromJson(reader, Elements.class);

System.out.println(elements.description.swagger_ui);
```

Dabei wurde folgende Strukturklasse verwendet:

```
public class Elements{
    public Description description;
    public Links links;

    public static class Description{
        public String swagger;
        @SerializedName("swagger-ui")
        public String swagger_ui;
    }

    public static class Links{
        public String product_url;
        public String vendors_url;
        public String categories_url;
        public String customers_url;
        public String orders_url;
    }
}
```

4. Benachrichtigungsfeld

JavaFX bietet nicht direkt ein Benachrichtigungsfeld an. Sie können jedoch die Bibliothek TrayNotification verwenden, welche auf der Basis von JavaFX aufgebaut ist:

<https://github.com/PlusHaze/TrayNotification>

Die entsprechende JAR-Datei können Sie über folgenden Link beziehen:

<https://github.com/PlusHaze/TrayNotification/raw/master/TrayTester.jar>