

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.

OOP2



2. Informationen

1. Web-API

Web-APIs (<u>REST</u>) sind weitverbreitete Schnittstellen, die es erlauben externe Strukturen zu lesen oder anzusteuern. In diesem Projekt können Sie von der SBB die Schnittstelle der Bahnverkehrsmeldungen unter folgendem Link beziehen:

https://data.sbb.ch/explore/dataset/rail-traffic-information/api/

Dabei wird über eine Web-Adresse (z.B. Get-Methode) die gewünschten Daten abgefragt und als HTML-Inhalt dargestellt. So können Sie die Daten unter folgendem Link erhalten:

https://data.sbb.ch/api/records/1.0/search/?dataset=rail-traffic-information&sort=published&pretty_print=true&rows=100

Die Daten werden Ihnen im sogenannten JSON-Format zurückgegeben. In diesem Beispiel-Link erhalten Sie die 100 letzten Meldungen der SBB.

Die Meisten API-Seiten geben ein kompaktes JSON-Format zurück, was die Betrachtung erschwert. Es empfiehlt sich, diese Daten mit dem Browser Firefox aufzurufen, da dieser die Daten formatiert und daher lesbarer ausgibt.

2. Webzugriff

Um den Inhalt einer Website zu lesen, können Sie die Klasse *java.net.URL* aus der Standard-Bibliothek nehmen. Mit folgendem Beispielcode erhalten Sie durch diese Klasse einen InputStreamReader von der API-Seite https://api.predic8.de/shop/:

```
String API_URL = "https://api.predic8.de/shop/";
URL url;
url = new URL(API_URL);
URLConnection request = url.openConnection();
request.connect();
InputStreamReader reader = new InputStreamReader((InputStream) request.getContent());
```

OOP2



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 *InputStreamReder* 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;
    }
}
```

OOP2



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