

1 Anzeigen von Freiburg(Brsg) Hbf

XML-DATEIEN

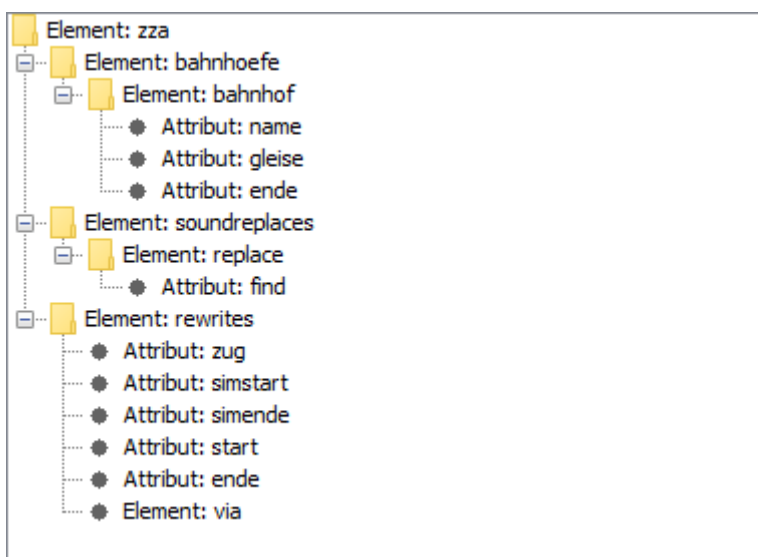
Anleitung zur Erstellung eigener Dateien für
das ZZA-Plugin

Andy (AP98) und Florian (floflo)

Aufbau eines XML-Dokuments

Grundlegend besteht ein XML-Dokument aus ineinander verschachtelten Elementen, denen Attribute mit Werten zugeordnet werden können. Bildlich kann man es sich so vorstellen, dass das Dokument selber ein großer Umzugswagen ist. In diesem Wagen befinden sich die Umzugskartons, die Elemente. Auf einigen der Kartons steht, in welches Zimmer sie kommen, die Attribute (in diesem Fall z.B. „Raum“) mit den Werten. Dann gibt es aber auch Kartons, auf denen gar nichts steht. In denen findet man aber wiederum noch mehr kleinere Kartons, auf denen vielleicht die Anzahl der Teller steht, die man darin findet.

Ein anderes Beispiel wäre ein Baumdiagramm, das sich immer weiter verzweigt. Die Struktur der Dateien, die für das Plugin benötigt werden wird in Abbildung 2 dargestellt. Was man bei den einzelnen Elementen einträgt oder welche Werte bei



den Attributen benötigt werden, erfährt ihr im Folgenden.

2 Baumdiagramm einer ZZA-Datei

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<zza>
  <bahnhoefe>
    <bahnhof name="Bahnhof" gleise="Bhf 1" ende="(Gleis
Bhf.*|Abstellung|BW|Bahnhof)"></bahnhof>
  </bahnhoefe>

  <soundreplaces>
    <replace find="Hbf">Hauptbahnhof</replace>
  </soundreplaces>

  <rewrites>

    <!--RB-Linie 1 A-Dorf - C-Stadt-->

    <rewrite zug="RB 17600" simstart="Besendorf"
simende="Franzenhausen" start="A-Dorf" ende="C-Stadt">
      <via>B-Dorf</via>
      <via>Franzenhausen</via>
    </rewrite>

  </rewrites>

</zza>

```

3 Beispieldatei

Der Aufbau

Ganz am Anfang steht immer `<?xml version="1.0" encoding="ISO-8859-1"?>` als Deklaration dafür, dass es ein XML-Dokument mit der ISO-8859-1 Kodierung ist.

Darauf folgend kommt das `<zza>`-Tag. Es zeigt uns, dass es sich um eine Datei für die Zugzielanzeigen handelt.

Bahnhof

Zwischen den `<bahnhof>`-Tags steht die Definition, für welche Gleise und Bahnhöfe die Zugzielanzeigen angezeigt werden sollen.

- **name:** Der Name des Bahnhofes
- **gleise:** Definition der Gleise, für die die ZZA angezeigt werden soll. Am Beispiel von Freiburg würde man dort zwischen die Anführungszeichen `RF [1-8]` eintragen. Was die eckigen Klammern bedeuten, darauf kommen wir im Abschnitt „Reguläre Ausdrücke“ zu sprechen
- **ende:** Hier trägt man alle Ausfahrten bzw. Einfahrten des Stellwerkes ein, in die ein Zug reinfahren kann, wenn er in dem genannten Bahnhof endet oder beginnt. Das sind zum Beispiel die Abstellungen und Betriebswerke. In Freiburg wären das `(BW 5[7-9] | Posthof | Gleis RF.* | Freiburg(Brsg) Hbf)`. Die Erklärungen zu den weiteren Zeichen finden sich im Abschnitt „Reguläre Ausdrücke“

Soundreplaces

In den Soundreplaces stehen, wie die Übersetzung schon sagt, Ausdrücke, die durch andere Ausdrücke ersetzt werden sollen. Dies wird vor allem bei Abkürzungen und Orten, die anders ausgesprochen werden als geschrieben benötigt, da die Vorlesedame alles so spricht, wie wir es geschrieben haben. So schreibt man beim

Attribut `find` des Elementes `replace` den Ausdruck rein, den man ersetzen möchte und zwischen die beiden Tags dann den Ausdruck durch den er ersetzt werden soll.

Rewrites

In den Rewrites stehen alle Züge, die den Bahnhof anfahren. Damit das Programm weiß, wo der Zug herkommt und wo er hinfährt und um welchen Zug es sich überhaupt handelt, gibt es auch dort wieder einige Attribute im Element `rewrite`:

- `zug`: Hier kommt die Zuggattung und die Zugnummer des Zuges rein, so wie sie auch im Sim zu lesen ist, z.B. `ICE 76`
- `simstart`: Wie der Name schon sagt, kommt hier der Name der Einfahrt rein, wo der Zug ins Stellwerk kommt. Das ist ebenfalls der Name, wie er im Stellwerk zu lesen ist. Im Falle des ICE 76 ist das `Ebringen`
- `simende`: Das ist analog zu `simstart` nun die Ausfahrt des Zuges. Am Beispiel des ICE 76 ist es `Köndringen`
- `start`: Hier trägt man den Startbahnhof ein. Beim ICE 76 ist das `Zürich HB`
- `ende`: Und hier kommt der Zielbahnhof rein – `Kiel Hbf`

Hat man alle Attribute festgelegt, kann man noch Unterwegshalte über das `<via>`-Tag definieren BEVOR man das Element `rewrite` wieder schließt.

Man kann so viele `soundreplaces` und `rewrites` für Züge anlegen, wie man möchte. Ebenfalls kann man die Zugzielanzeigen für mehrere Bahnhöfe definieren. Dazu schreibt man einfach eine weitere Zeile zwischen `<bahnhoefe>` und `</bahnhoefe>`.

Reguläre Ausdrücke

Reguläre Ausdrücke werden genutzt, um Zeichenketten mit syntaktischen Regeln zu beschreiben. So beschreibt z.B. `[1-9]` eine Zahl von 1 bis 9 und `[ACE]` einen der Buchstaben A, C oder E (Groß- und Kleinschreibung ist wichtig!). Wichtige Ausdrücke sind folgende:

- `[ahCO]`: eines der Zeichen a, h, C oder O
- `[A-Za-z0-9]`: ein beliebiger Buchstabe von A bis Z oder a bis z oder eine beliebige Ziffer von 1 bis 9
- `[^t]`: ein beliebiges Zeichen außer dem t
- `ICE?`: das E kann einmal vorkommen, muss aber nicht. Daraus ergibt sich IC oder ICE
- `ICE+`: das E muss mindestens einmal vorkommen. Daraus ergeben sich die Zeichenketten ICE, ICEE, ICEEE, ...
- `ICE*`: dies ist eine Kombination aus ? und +. Das E darf beliebig oft vorkommen, also auch keinmal oder hundertmal.
- `[0-9]{2}`: zweimal eine Ziffer von 0 bis 9.
- `Gleis .:` der Punkt steht für ein beliebiges Zeichen. In Verbindung mit einem *, ist praktisch egal, was nach Gleis steht.
- `(Posthof | BW)`: Posthof oder BW
- `^ICE 76:` das ^ steht für den Zeilenanfang. Es kann verwendet werden, muss aber nicht

Weitere Beispiele sind bei Wikipedia zu finden:

https://de.wikipedia.org/wiki/Regul%C3%A4rer_Ausdruck

Möchte man seinen regulären Ausdruck überprüfen, so kann man das auf verschiedenen Seiten, wie z.B. <https://regex101.com/> tun.

Wollt ihr eure fertige XML-Datei auf Fehler überprüfen, so könnte ihr das online mit einem XML-Validator, wie z.B. <http://www.xmlvalidation.com/> machen.

Für Diejenigen, denen die ganze Tipperei mit den Tags und so weiter zu viel ist, für die ist das Programm STS-XML.jar geschrieben worden. Dieses findet sich im STS-Forum <http://www.stellwerksim.de/forum/viewtopic.php?f=96&t=28126>

Bei weiteren Fragen stehen euch die kompetenten Spieler im Forum zur Verfügung, unter anderem hier: <http://www.stellwerksim.de/forum/viewtopic.php?f=96&t=20214>

Die beste Übung ist übrigens, sich bereits erstellte Dateien anzuschauen und versuchen diese nachzuvollziehen, indem man sich nebenher das Stellwerk anschaut und vielleicht auch das Plugin mit der Datei laufen lässt.

Viel Spaß beim Schreiben wünschen Andy (AP98) und Florian (floflo)