Cache Generation Guidelines

The guideline aims to provide step by step execution of tools required for the generation of Trento urban cache. The purpose of this guideline is to enable the user to generate the cache zip file in an autonomous way. It is equipped with a cache-generator.zip file which includes a container environment pre-installed with all the major frameworks necessary for the execution of the toolkit. Download the toolkit from the link below and unzip it.

```
https://github.com/smartcommunitylab/pt-data-generator/releases/download/v1.0.0/cache-generator.zip
```

There are four major tools required in this process which are provided inside a docker container image. In this context, a central role is played by the GTFS zip file, which can be downloaded from the open data portal of Trentino or from the file server of Trentino Trasporti. In some cases, this file is used as input. The output of this tutorial are the zipped cache files.

```
routesdb_trento.zip
routesdb_trento_extended.zip
```

There are four major tools required in this process.

1. **CSV Generator** (Perl script)
2. **Annotated Timetables Generator** (Java executable jar)
3. **OpenTripPlanner** (Java executable jar)
4. **Smart Planner** (Java executable jar).

The container image is pre-installed with java and perl framework distribution required for tools execution. The tools are required to execute in the following order

1. Container launch
2. CSV generation
3. GTFS fetch
4. CSV annotation
    1. Bus timetables
    2. Funivia timetables
5. Smartplanner GTFS alignment
6. Train GTFS generation
7. Opentripplanner (OTP) Graph Generation
8. Smartplanner Cache

1. Container Launch

Run the container using the following command from the root of the extracted zip (cache-generator.zip).

```
\cache-generator>docker compose up
```

Open a new terminal and connect to the `cache-generator` container in interactive mode using the following command.

```
..\cache-generator>docker exec -it cache-generator /bin/bash
```

Once inside the container shell, change to director `cache-generator`

2. CSV generation

The first step in the process of cache generation starts with converting the timetable provided on Trentino trasporti website into csv format. In the context of smartplanner engine this major functionality is achieved with the development of `CSV generation` tool in Perl programming language. In this step we will demonstrate the use of perl script to convert the timetables from pdf to csv format downloading it Trentino trasporti official website

```
https://www.trentinotrasporti.it/pdforari/urbani/linee/
```

The scripts are present inside the 'BusExtractTn' folder. In this first step it is required to configure the script. Open the '**extract-tn.pl**' script using a text editor.

Configuration

The following parameters are required to be configured before script execution.

Url Prefix

The first important parameter to configure inside script is located at line 26 (extract-tn.pl), the variable '$BASE_TTE_NAME' corresponds to the url of the urban bus timetable. In winter the url could be like

```
https://www.trentinotrasporti.it/pdforari/urbani/linee/OrariDiDirettrice-T23L-T-01_A-Festivo.PDF
```

In summer time the it could be

```
https://www.trentinotrasporti.it/pdforari/urbani/linee/OrariDiDirettrice-T24L-T-01_A-Festivo.PDF
```

$BASE_TTE_NAME variable corresponds to orange text in the url

Note: Please check the url pattern and set this variable accordingly.

Date

The second important parameter to configure is the date string inside the PDF files. The script performs pattern matching of date string in PDF to validate the dates during the PDF scan process. It is required to align the date strings pattern accordingly in the script file. In order to do so, one has to modify the regex pattern string at line 170 of perl script file (BusExtractTn.pl)

```
/<b>(ORARIO Trento Inverno 2023-24 Valido dal ..\/..\/2023 al ..\/06\/2024 )<\/b>/
```

Execute

The script is ready to be executed however , it is recommended to clean the old data by running the (clean-tn.sh) script inside the 'cache-generator/BusExtractTn' folder.

```
root@229c03ad87f7:/cache-generator/BusExtractTn# sh clean-tn.sh
```

Run the perl script.

```
root@229c03ad87f7:/cache-generator/BusExtractTn# perl extract-tn.pl
```

Error handling.

The script one by one downloads the pdf per bus route, performing PDF to XML conversion. In the later phase it transforms the XML files to CSV format. One can see inside the 'BusExtractTn/pdf-Trento' folder the downloaded pdf files. The converted xml files are inside the 'BusExtractTn/xml-Trento' folder, and the generated CSV lies inside the 'BusExtractTn/csv-Trento' folder.

Before running the script, it would be a good idea to verify whether the $DEBUG variable on line 9 is set as follows:

```
my $DEBUG = 0;
```

In this manner, if errors arise during the course of the process, it becomes possible to immediately identify the file responsible for these errors. In fact, errors related to line ordering may occur on occasion. Since these errors need to be fixed manually, it is necessary to have as much information as possible. This information can be obtained by configuring the script so that it runs in DEBUG mode. Set debug flag level to 5 (maximum) inside perl script. Therefore, the $DEBUG variable on line 9 should be set as follows:

```
my $DEBUG = 5;
```

If one runs the script in NON-DEBUG mode and an error occurs, the following logs are generated:

```
XMLLING 10_A-Feriale

Page-1

..

Page-12

EXTRACTING LINE: 10_A-Feriale

ERRORE INTESTAZIONE (Validit�):
```

The error indicates a problem of line order during the handling of a bus route's pdf file (in this case the pdf file whose name contains the string **10_A-Feriale**). It is required to fix it by opening the corresponding xml file inside the 'BusExtractTn/xml-Trento' folder (in this case the xml file whose name contains the string **10_A-Feriale**). Running the script again in DEBUG mode will provide more detailed logs:

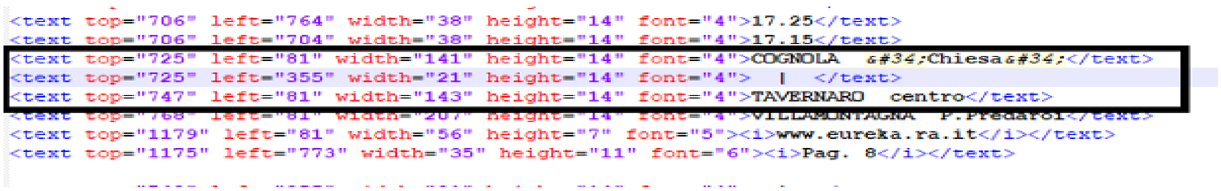
```
<text top="684" left="704" width="38" height="14" font="4">17.13</text>
<text top="704" left="81" width="181" height="14" font="4">COGNOLA  &#34;Centro civico&#34;</text>
<text top="706" left="347" width="38" height="14" font="4">16.10</text>
<text top="706" left="406" width="38" height="14" font="4">16.25</text>
<text top="706" left="525" width="38" height="14" font="4">16.35</text>
<text top="706" left="585" width="38" height="14" font="4">16.55</text>
<text top="706" left="764" width="38" height="14" font="4">17.25</text>
<text top="706" left="704" width="38" height="14" font="4">17.15</text>
<text top="725" left="81" width="141" height="14" font="4">COGNOLA  &#34;Chiesa&#34;</text>
<text top="747" left="81" width="143" height="14" font="4">TAVERNARO  centro</text>
<text top="768" left="81" width="207" height="14" font="4">VILLAMONTAGNA  P.Predaroi</text>
<text top="1179" left="81" width="56" height="7" font="5"><i>www.eureka.ra.it</i></text>
<text top="1175" left="773" width="35" height="11" font="6"><i>Pag. 8</i></text>
<text top="725" left="355" width="21" height="14" font="4"> | </text>
ERRORE INTESTAZIONE (Validit��):  |
```

Search for the line inside the xml file. You can see that after the last line of the PDF, there are some more lines printed in xml with wrong ordering with respect to the `top` field. In order to fix them, move the tuple in the correct position by ordering them by `top` field.

For e.g.

```
<text top="725" left="355" width="21" height="14" font="4"> | </text>
```

Must be moved above in between the following two lines



```
<text top="706" left="764" width="38" height="14" font="4">17.25</text>
<text top="706" left="704" width="38" height="14" font="4">17.15</text>
<text top="725" left="81" width="141" height="14" font="4">COGNOLA  &#34;Chiesa&#34;</text>
<text top="725" left="355" width="21" height="14" font="4"> | </text>
<text top="747" left="81" width="143" height="14" font="4">TAVERNARO  centro</text>
<text top="768" left="81" width="207" height="14" font="4">VILLAMONTAGNA  P.Predaroi</text>
<text top="1179" left="81" width="56" height="7" font="5"><i>www.eureka.ra.it</i></text>
<text top="1175" left="773" width="35" height="11" font="6"><i>Pag. 8</i></text>
```

Move and arrange all the element above in correct order which are present after the last line

```
<text top="1175" left="773" width="35" height="11" font="6"><i>Pag. 8</i></text>
```

The fixed XML for route file will look like the following

```
<page number="8" position="absolute" top="0" left="0" height="1262" width="892">
<image top="77" left="77" width="124" height="43"
src="xml-Trento/OrariDiDirettrice-T23L-T-10_A-Feriale-8_1.jpg"/>
<text top="117" left="366" width="442" height="12" font="0"><b>ORARIO Trento Inverno 2023-24   Valido
dal 11/09/2023 al 11/06/2024 </b></text>
<text top="89" left="628" width="180" height="19" font="1"><b>ORARIO FERIALE</b></text>
<text top="151" left="140" width="544" height="22" font="2"><b> P.Dante - Martignano - Cognola /
Montevaccino</b></text>
```

```
<text top="151" left="93" width="27" height="22" font="2"><b>10</b></text>
<text top="181" left="81" width="60" height="11" font="3"><b>Frequenza</b></text>
<text top="180" left="360" width="11" height="16" font="7">ê</text>
<text top="180" left="420" width="10" height="16" font="7">ß</text>
<text top="180" left="480" width="10" height="16" font="7">ß</text>
<text top="180" left="539" width="10" height="16" font="7">Þ</text>
<text top="180" left="658" width="10" height="16" font="7">Þ</text>
<text top="180" left="718" width="10" height="16" font="7">Þ</text>
<text top="180" left="777" width="10" height="16" font="7">ß</text>
<text top="198" left="81" width="31" height="11" font="3"><b>Linea</b></text>
<text top="199" left="367" width="17" height="11" font="3"><b>10/</b></text>
<text top="213" left="367" width="3" height="11" font="3"><b> </b></text>
<text top="199" left="486" width="10" height="11" font="3"><b>M</b></text>
<text top="213" left="486" width="3" height="11" font="3"><b> </b></text>
<text top="199" left="665" width="10" height="11" font="3"><b>M</b></text>
<text top="213" left="665" width="3" height="11" font="3"><b> </b></text>
<text top="215" left="81" width="190" height="14" font="4">Piazza Dante  &#34;Stazione FS&#34;</text>
<text top="217" left="355" width="21" height="14" font="4"> | </text>
<text top="217" left="406" width="38" height="14" font="4">16.01</text>
<text top="217" left="525" width="38" height="14" font="4">16.11</text>
<text top="217" left="585" width="38" height="14" font="4">16.31</text>
<text top="217" left="764" width="38" height="14" font="4">17.01</text>
<text top="217" left="704" width="38" height="14" font="4">16.51</text>
<text top="236" left="81" width="181" height="14" font="4">Rosmini  S.Maria Maggiore</text>
<text top="237" left="355" width="21" height="14" font="4"> | </text>
<text top="238" left="406" width="38" height="14" font="4">16.03</text>
<text top="238" left="525" width="38" height="14" font="4">16.13</text>
<text top="238" left="585" width="38" height="14" font="4">16.33</text>
<text top="238" left="764" width="38" height="14" font="4">17.03</text>
<text top="238" left="704" width="38" height="14" font="4">16.53</text>
<text top="258" left="81" width="162" height="14" font="4">Travai  al Nuoto / MUSE</text>
```

```
:
:
<text top="578" left="466" width="38" height="14" font="4">16.22</text>
<text top="578" left="534" width="21" height="14" font="4"> | </text>
<text top="578" left="593" width="21" height="14" font="4"> | </text>
<text top="578" left="772" width="21" height="14" font="4"> | </text>
<text top="578" left="713" width="21" height="14" font="4"> | </text>
<text top="578" left="644" width="38" height="14" font="4">16.52</text>
<text top="598" left="81" width="120" height="14" font="4">MASO BOLLERI  </text>
<text top="599" left="347" width="38" height="14" font="4">16.17</text>
<text top="599" left="415" width="21" height="14" font="4"> | </text>
<text top="599" left="466" width="38" height="14" font="4">16.24</text>
```

```
<text top="599" left="534" width="21" height="14" font="4"> | </text>
<text top="599" left="593" width="21" height="14" font="4"> | </text>
<text top="599" left="772" width="21" height="14" font="4"> | </text>
<text top="599" left="713" width="21" height="14" font="4"> | </text>
<text top="599" left="644" width="38" height="14" font="4">16.54</text>
<text top="619" left="81" width="131" height="14" font="4">MONTEVACCINO </text>
<text top="621" left="347" width="38" height="14" font="4">16.26</text>
<text top="621" left="415" width="21" height="14" font="4"> | </text>
<text top="621" left="466" width="38" height="14" font="4">16.33</text>
<text top="621" left="534" width="21" height="14" font="4"> | </text>
<text top="621" left="593" width="21" height="14" font="4"> | </text>
<text top="621" left="772" width="21" height="14" font="4"> | </text>
<text top="621" left="713" width="21" height="14" font="4"> | </text>
<text top="621" left="644" width="38" height="14" font="4">17.03</text>
<text top="640" left="81" width="164" height="14" font="4">MARTIGNANO  Costiole</text>
<text top="642" left="406" width="38" height="14" font="4">16.20</text>
<text top="642" left="525" width="38" height="14" font="4">16.30</text>
<text top="642" left="585" width="38" height="14" font="4">16.50</text>
<text top="642" left="764" width="38" height="14" font="4">17.20</text>
<text top="642" left="704" width="38" height="14" font="4">17.10</text>
<text top="662" left="81" width="186" height="14" font="4">MARTIGNANO  Formigheta</text>
<text top="663" left="406" width="38" height="14" font="4">16.21</text>
<text top="663" left="525" width="38" height="14" font="4">16.31</text>
<text top="663" left="585" width="38" height="14" font="4">16.51</text>
<text top="663" left="764" width="38" height="14" font="4">17.21</text>
<text top="663" left="704" width="38" height="14" font="4">17.11</text>
<text top="683" left="81" width="107" height="14" font="4">ZELL  Marnighe</text>
<text top="684" left="406" width="38" height="14" font="4">16.23</text>
<text top="684" left="525" width="38" height="14" font="4">16.33</text>
<text top="684" left="585" width="38" height="14" font="4">16.53</text>
<text top="684" left="764" width="38" height="14" font="4">17.23</text>
<text top="684" left="704" width="38" height="14" font="4">17.13</text>
<text top="704" left="81" width="181" height="14" font="4">COGNOLA  &#34;Centro civico&#34;</text>
<text top="706" left="347" width="38" height="14" font="4">16.10</text>
<text top="706" left="406" width="38" height="14" font="4">16.25</text>
<text top="706" left="525" width="38" height="14" font="4">16.35</text>
<text top="706" left="585" width="38" height="14" font="4">16.55</text>
<text top="706" left="764" width="38" height="14" font="4">17.25</text>
<text top="706" left="704" width="38" height="14" font="4">17.15</text>
<text top="725" left="81" width="141" height="14" font="4">COGNOLA  &#34;Chiesa&#34;</text>
<text top="725" left="355" width="21" height="14" font="4"> | </text>
<text top="747" left="81" width="143" height="14" font="4">TAVERNARO  centro</text>
<text top="748" left="355" width="21" height="14" font="4"> | </text>
<text top="768" left="81" width="207" height="14" font="4">VILLAMONTAGNA  P.Predaroi</text>
<text top="769" left="355" width="21" height="14" font="4"> | </text>
```

```
<text top="1179" left="81" width="56" height="7" font="5"><i>www.eureka.ra.it</i></text>
<text top="1175" left="773" width="35" height="11" font="6"><i>Pag. 8</i></text>
</page>
```

Rerun the script after reordering to fix the error. This kind of error can occur for multiple files and must be fixed as described above. One can check the generated CSV files total counts against the provided PDF on the website to make sure all timetables are converted in .csv format. Another possible error that can happen is due to unavailability of certain pdf. This can be ignored. For e.g. bus B does not offer service in a certain period of year for e.g. winter time.

```
XMLLING %20B_C-Feriale
Syntax Warning: May not be a PDF file (continuing anyway)
```

The list of all the routes(daily and festival) that will be used for the generation of `.csv` is specified at line 14 of the script `extract-tn.pl`.

```
my @routes=
('01_A','01_R','02_C','03_A','03_R','04_A','04_R','05_A','05_R','05_b','06_A','06_R','07_A','07_R','08_A','08_R','
09_A','09_R','10_A','10_R','11_A','11_R',
'12_A','12_R','13_A','13_R','14_A','14_R','15_A','15_R','16_A','16_R','17_A','17_R','NP_C','%20A_D','%20B_C','
%20C_A','%20C_R','%20G_A','%20G_R','CM_A','CM_R');

my @routes_festivo =
('01_A','01_R','02_C','03_A','03_R','04_A','04_R','05_A','05_R','06_A','06_R','08_A','08_R','10_A','10_R','12_A','
12_R','17_A','17_R','%20A_C');
```

Note: Please check this variable `@routes` in case some files are skipped during the process. It can happen due to change in name at the part of trentino trasporti.

3. GTFS fetch

Download the gtfs from the open data portal of trentino using the following link or from the ftp file server of Trentino trasporti.

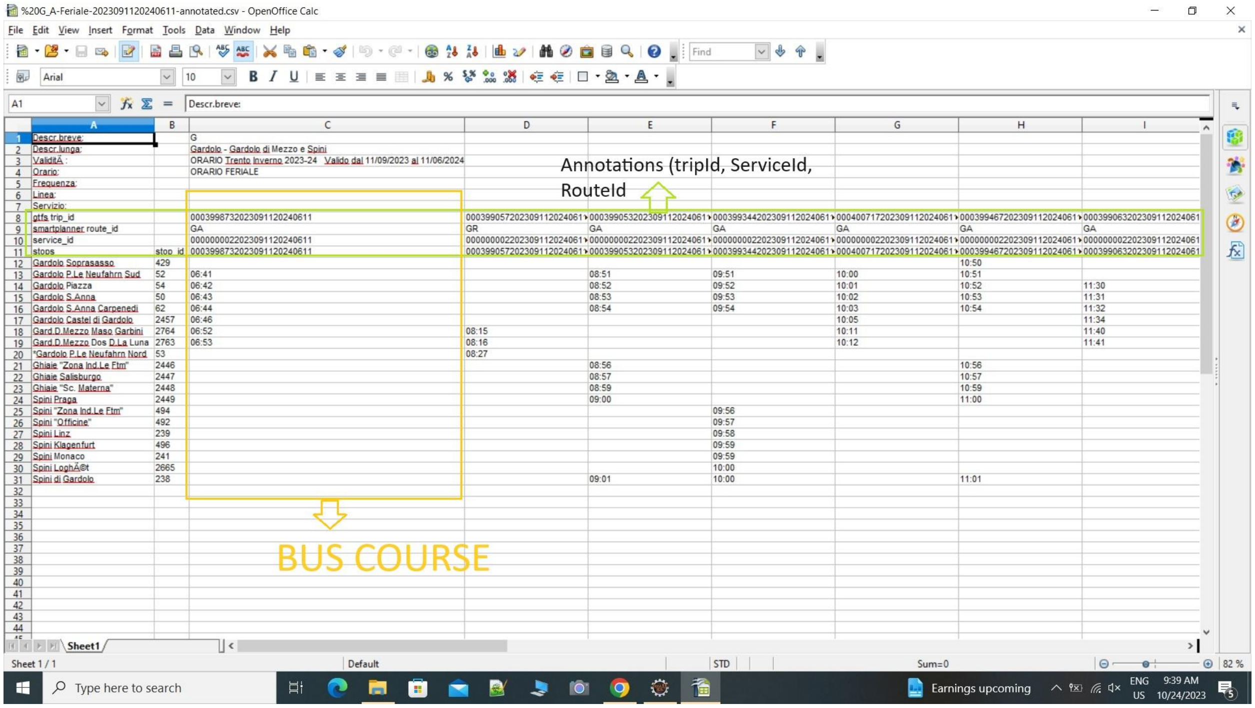
```
https://www.trentinotrasporti.it/opendata/google_transit_urbano_tte.zip
```

4. CSV annotation

The second important step in the cache-generation process is the annotation of csv files with GTFS data .In this step the .csv files generated in previous steps are annotated with tripId and serviceId from GTFS using the Annotated TimeTable Generator tool.

CSV structure

The generated csv for each bus is structured as follows.



%20G_A-Feriale-2023091120240611-annotated.csv - OpenOffice Calc

	A	B	C	D	E	F	G	H	I
1	Descr.breve:		G						
2	Descr.lunga:		Gardolo - Gardolo di Mezzo e Spini						
3	ValiditÃ :		ORARIO Trento Inverno 2023-24 Valido dal 11/09/2023 al 11/06/2024						
4	Orario:		ORARIO FERIALE						
5	Frequenza:								
6	Linea:								
7	Servizio:								
8	gtfs trip_id		0003998732023091120240611	000399057202309112024061▸	000399053202309112024061▸	000399344202309112024061▸	000400717202309112024061▸	000399467202309112024061▸	000399063202309112024061
9	smartplanner route_id		GA	GR	GA	GA	GA	GA	GA
10	service_id		0000000022023091120240611	000000002202309112024061▸	000000002202309112024061▸	000000002202309112024061▸	000000002202309112024061▸	000000002202309112024061▸	000000002202309112024061
11	stops	stop_id	0003998732023091120240611	000399057202309112024061▸	000399053202309112024061▸	000399344202309112024061▸	000400717202309112024061▸	000399467202309112024061▸	000399063202309112024061
12	Gardolo Soprasasso	429						10:50	
13	Gardolo P.Le Neufahrn Sud	52	06:41		08:51	09:51	10:00	10:51	
14	Gardolo Piazza	54	06:42		08:52	09:52	10:01	10:52	11:30
15	Gardolo S.Anna	50	06:43		08:53	09:53	10:02	10:53	11:31
16	Gardolo S.Anna Carpenedi	62	06:44		08:54	09:54	10:03	10:54	11:32
17	Gardolo Castel di Gardolo	2457	06:46				10:05		11:34
18	Gard.D.Mezzo Maso Garbini	2764	06:52	08:15			10:11		11:40
19	Gard.D.Mezzo Dos D.La Luna	2763	06:53	08:16			10:12		11:41
20	*Gardolo P.Le Neufahrn Nord	53		08:27					
21	Ghiaie "Zona Ind.Le Ftm"	2446			08:56			10:56	
22	Ghiaie Salisburgo	2447			08:57			10:57	
23	Ghiaie "Sc. Materna"	2448			08:59			10:59	
24	Spini Praga	2449			09:00			11:00	
25	Spini "Zona Ind.Le Ftm"	494				09:56			
26	Spini "Officine"	492				09:57			
27	Spini Linz	239				09:58			
28	Spini Klagenfurt	496				09:59			
29	Spini Monaco	241				09:59			
30	Spini LoghÃ©t	2665				10:00			
31	Spini di Gardolo	238			09:01	10:00		11:01	

Each column in the .csv file corresponds to the bus course. The annotated parameters like tripId, routeId, service Id are added after the match with GTFS data. The tools aimed at mapping the bus course inside the .csv file to the corresponding unique 'TripId'. Inside GTFS each mobility course is identified by a unique tripId. For more detailed information about how GTFS works, please refer to

https://gtfs.org/

For the scope of this tutorial it is important to understand that each column of the .csv file is a bus course timetable. The Annotated Timetable Generator tools scan each column one by one. It reads the GTFS data to map the course to the corresponding GTFS trip Id using a complex matching algorithm. As a result of this step each column in the .csv file gets annotated with gtfs trip id. With such correlation, it is possible to identify and associate real time information to the mobility course such as time delay, notification etc..

Bus timetables

The executable `annotated.timetable.generator.jar` is located inside the folder `annotated.timetable.generator`. The tool structured is as follows

```
|- annotated.timetable.jar
|- file-route.json
|- tn-routemodel.json
|- urban-tn-annotater.json
`- resources
      |- annotatedtimetable
      |- gtfs
      `- inputtimetable
```

The `.csv` files generated in the previous steps must be copied in the folder `resources/inputtimetable/12` . The GTFS files (calendar.txt. Calendar_dates.txt, trips.txt, stoptimes.txt) downloaded in step 4 (GTFS fetch) must be copied inside the `resources/gtfs/12` folder.

Configuration

The configuration files are `file-route.json`, `tn-routemodel.json` and `urban-tn-annotater.json`. The file-route.json file contains mapping of multiple GTFS route ids to a single route which is used during the course of matching algorithm. In the context of Trento urban, often a bus course is being covered by a bus with a different route id. The mapping of such cases is specified in the `file-route.json` section `fileRouteMappings`. It contains a list of .csv file names mapped to a possible list of routeId which the search algorithm takes into consideration for correlating bus courses to corresponding tripId inside GTFS. For the sake of this tutorial it is not required to change this file. The smartplanner timetable for the sake of complexity avoids courses with different routes in a timetable and is completely coherent. The `tn-routemodel.json` file defines the mapping between GTFS routeId and smartplanner routeIds. For the sake of simplicity and by application design, the smartplanner provides a more meaningful title to routes in both directions(ongoing/return). This file is required to be modified only in case the GTFS routeId for a bus is changed in the GTFS data. The `urban-tn-annotater.json` file is the actual configuration file which needs to be modified in different seasons of timetable. The file looks like the following

```
{
        "ignoreServiceIdPattern": [
                "20230910"
        ],
        "outputPattern": "2023091120240611",
        "funiviaRouteId": "531"
}
```

The parameter ‘ignoreServiceIdPattern’ indicates the serviceId to be ignored during the course of match algorithm execution. Often it happens that the GTFS data comes with trips of summer and winter season. In this situation the `ignoreServiceIdPattern` configuration parameter excludes during the course of match, all service Ids that end with such a pattern. The `outputPattern` configuration parameter defines the postfix of generated timetable files in the **resources/annotatedtimetable/12** folder.

Execute

Run the tool using the following command

```
root@229c03ad87f7:/cache-generator/annotated.timetable.generator# java -jar annotated-timetable-generator.jar
```

Error Handling

There could be some error due to invalid GTFS data for e.g. often stoptimes.txt file comes with strange empty lines in between that must be removed.

```
;;;;;
;;;;;
```

In general the scripts end with following logs about coverage. The warnings are related to the fact that different bus routes are included in some pdf. The smartplanner timetables are coherent with respect to the bus route Id and does not include different routes within a single timetable.

```
---------- WARNINGS ----------
check pdf 14_A-Feriale-2023091120240611-annotated.csv for columns 8,
check pdf 09_R-Feriale-2023091120240611-annotated.csv for columns 32,
check pdf 08_R-Feriale-2023091120240611-annotated.csv for columns 11,
check pdf 14_R-Feriale-2023091120240611-annotated.csv for columns 9,
----------------------------

%%%%%%%%%% CSV STATS %%%%%%%%%%
total csv trips: 2638
csv trips covered by GTFS: 2404.0
csv trips not covered by GTFS: 4.0
csv trips ignored (merged routes): 230
csv coverage: 99.83388704318938
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

There were cases where the GTFS data was not coherent to the generated PDF file. In that case often manual adjustment to the local GTFS file inside`resources/gtfs/12` can be applied to match with the times inside CSV (in case).

Note: In the current version of GTFS route name `P`(routeId 617) has been added in addition to route name `16`(routeId 484). In the context of smartplanner only route name 16(routeId 484) exists which means that in order to generate the annotated timetable correctly for bus 16, it is required to replace inside trips.txt file all the route Id 617 with 484. Like another bus route name `M`(routeId 615) is added besides route Id 408 for route name `10` which is known to smartplanner. So all the occurrences of routeId 615 inside trips.txt must be replaced with 408.

After fixing the trips.txt file as can be seen inside the distribution (cache-generation.zip) file, the result after the suggested fixes is the following.

```
21:12
21:13
21:14
21:15
21:18
21:20
21:22
21:24
NP_C-Feriale.csv -> 36

---------- WARNINGS ----------
----------------------------
```

```
%%%%%%%%%% CSV STATS %%%%%%%%%%
total csv trips: 2638
csv trips covered by GTFS: 2408.0
csv trips not covered by GTFS: 0.0
csv trips ignored (merged routes): 230
csv coverage: 100.0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Funivia timetables

The funiva timetables are generated using the `Funivia Generator` application which reads the GTFS and generates the .csv file contents to be copied inside the annotated csv files.

Configuration

The configuration file is `urban-tn-annotater.json` where it is required to configure the GTFS route_id for Funivia(by default it is 531) as can be seen in the GTFS routes.txt file inside folder `resources/gtfs/12`.

```
531,12,Fu,Funivia Trento Sardagna,5,,
```

Execute

In order to generate Funiva timetable for ongoing course TRENTO-SARDAGNA, run the following command with a parameter.

```
root@229c03ad87f7:/cache-generator/annotated.timetable.generator# java -cp
annotated-timetable-generator.jar sayservice.FuniviaGenerator a
```

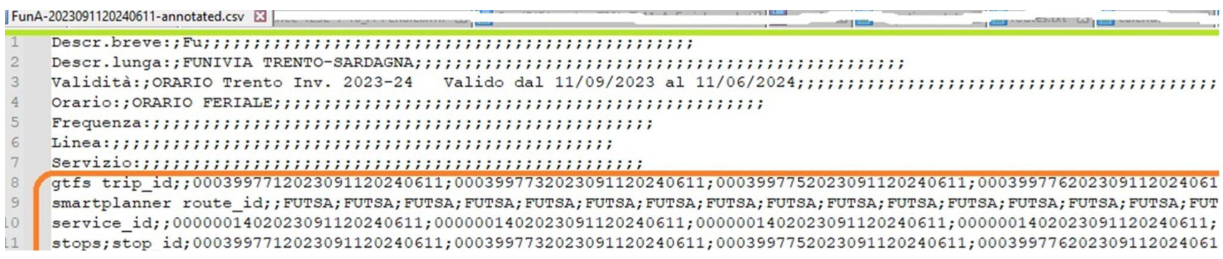
The result of execution will result in the following timetable .csv snippet.

```
gtfs
trip_id;;0003929692023061220230910;0003929702023061220230910;0003929712023061220230910;0003929
722023061220230910;0003929732023061220230910;0003929742023061220230910;000392975202306122202
30910;0003929762023061220230910;0003929772023061220230910;0003929782023061220230910;00039297
92023061220230910;0003929802023061220230910;0003929812023061220230910;00039298220230612202З
0910;0003929832023061220230910;0003929842023061220230910;0003929852023061220230910;000392986
2023061220230910;0003929872023061220230910;0003929882023061220230910;000392989202306122023
910;0003929902023061220230910;0003929912023061220230910;0003929922023061220230910;0003929932
023061220230910;0003929942023061220230910;0003929952023061220230910;00039299620230612202309
10;0003929972023061220230910;0003929982023061220230910;0003929992023061220230910;00039300020
23061220230910;0003930012023061220230910;0003930022023061220230910;00039300320230612202309l
0;0003930042023061220230910;0003930052023061220230910;0003930062023061220230910;000393007202
3061220230910;0003930082023061220230910;0003930092023061220230910;0003930102023061220230910;
0003930112023061220230910;0003930122023061220230910;0003930132023061220230910;000393014202З
061220230910;0003930152023061220230910;0003930162023061220230910;0003930172023061220230910;0
003997712023091120240611;0003997732023091120240611;0003997752023091120240611;00039977620230
91120240611;0003997772023091120240611;0003997782023091120240611;0003997792023091120240611;00
03997802023091120240611;0003997812023091120240611;0003997822023091120240611;000399783202309
1120240611;0003997842023091120240611;0003997852023091120240611;0003997862023091120240611;000
3997872023091120240611;0003997882023091120240611;0003997892023091120240611;000399790202309l
120240611;0003997912023091120240611;0003997922023091120240611;0003997932023091120240611;0003
```

```
99794202309112024061l;000399795202309112024061l;000399796202309112024061l;00039979720230911
20240611;000399798202309112024061l;000399799202309112024061l;000399800202309112024061l;00039
9801202309112024061l;000399802202309112024061l;000399803202309112024061l;00039980420230911 2
0240611;000399805202309112024061l;000399806202309112024061l;000399807202309112024061l;000399
808202309112024061l;000399809202309112024061l;000399810202309112024061l;0003998112023091120
240611;000399812202309112024061l;000399813202309112024061l;000399814202309112024061l;0003998
15202309112024061l;000399816202309112024061l;000399817202309112024061l;0003998182023091120 2
40611;000399819202309112024061l;000399820202309112024061l;000399821202309112024061l;
smartplanner
route_id;;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;F
UTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUT
SA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;
FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FU
TSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA
;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FU
TSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;
service_id;;000000140202306122023091O;000000140202306122023091O;000000140202306122023091O;0000
00140202306122023091O;000000140202306122023091O;000000140202306122023091O;00000014020230612
2023091O;000000140202306122023091O;000000140202306122023091O;000000140202306122023091O;00000
0140202306122023091O;000000001202306122023091O;000000001202306122023091O;00000000120230612 2
0230910;000000001202306122023091O;000000001202306122023091O;000000001202306122023091O;000000
001202306122023091O;000000001202306122023091O;000000001202306122023091O;0000000012023061220
230910;000000001202306122023091O;000000140202306122023091O;000000140202306122023091O;0000001
40202306122023091O;000000001202306122023091O;000000001202306122023091O;000000001202306122 02
30910;000000001202306122023091O;000000001202306122023091O;000000001202306122023091O;00000000
1202306122023091O;000000001202306122023091O;000000001202306122023091O;000000001202306122023
0910;000000001202306122023091O;000000001202306122023091O;000000001202306122023091O;000000001
202306122023091O;000000001202306122023091O;000000001202306122023091O;000000140202306122023 0
910;000000140202306122023091O;000000140202306122023091O;000000140202306122023091O;0000001402
02306122023091O;000000140202306122023091O;000000140202306122023091O;000000140202306122023 09
10;000000140202309112024061l;000000140202309112024061l;000000140202309112024061l;00000014020
2309112024061l;000000140202309112024061l;000000140202309112024061l;00000014020230911202406 1
1;000000140202309112024061l;000000140202309112024061l;000000140202309112024061l;000000140202
309112024061l;000000001202309112024061l;000000001202309112024061l;000000001202309112024061l;
000000001202309112024061l;000000001202309112024061l;000000001202309112024061l;00000000120 23
091120240611;000000001202309112024061l;000000001202309112024061l;000000001202309112024061l;0
00000001202309112024061l;000000140202309112024061l;000000140202309112024061l;0000001402023 0
91120240611;000000001202309112024061l;000000001202309112024061l;000000001202309112024061l;00
0000001202309112024061l;000000001202309112024061l;000000001202309112024061l;000000001202309
1120240611;000000001202309112024061l;000000001202309112024061l;000000001202309112024061l;000
000001202309112024061l;000000001202309112024061l;000000001202309112024061l;000000001202309 1
120240611;000000001202309112024061l;000000001202309112024061l;000000140202309112024061l;0000
00140202309112024061l;000000140202309112024061l;000000140202309112024061l;00000014020230911
20240611;000000140202309112024061l;000000140202309112024061l;000000140202309112024061l;
stops;stop_id;000392969202306122023091O;000392970202306122023091O;000392971202306122023091O;00
0392972202306122023091O;000392973202306122023091O;000392974202306122023091O;000392975202306
1220230910;000392976202306122023091O;000392977202306122023091O;000392978202306122023091O;000
392979202306122023091O;000392980202306122023091O;000392981202306122023091O;0003929822023061
220230910;000392983202306122023091O;000392984202306122023091O;000392985202306122023091O;0003
92986202306122023091O;000392987202306122023091O;000392988202306122023091O;00039298920230612
20230910;000392990202306122023091O;000392991202306122023091O;000392992202306122023091O;00039
2993202306122023091O;000392994202306122023091O;000392995202306122023091O;000392996202306122
```

```
0230910;0003929972023061220230910;0003929982023061220230910;0003929992023061220230910;000393
0002023061220230910;0003930012023061220230910;0003930022023061220230910;0003930032023061220
230910;0003930042023061220230910;0003930052023061220230910;0003930062023061220230910;0003930
072023061220230910;0003930082023061220230910;0003930092023061220230910;00039301020230612202
30910;0003930112023061220230910;0003930122023061220230910;0003930132023061220230910;00039301
42023061220230910;0003930152023061220230910;0003930162023061220230910;000393017202306122023
0910;0003997712023091120240611;0003997732023091120240611;0003997752023091120240611;000399776
2023091120240611;0003997772023091120240611;0003997782023091120240611;0003997792023091120240
611;0003997802023091120240611;0003997812023091120240611;0003997822023091120240611;0003997832
023091120240611;0003997842023091120240611;0003997852023091120240611;00039978620230911202406
11;0003997872023091120240611;0003997882023091120240611;0003997892023091120240611;00039979020
23091120240611;0003997912023091120240611;0003997922023091120240611;000399793202309112024061
1;0003997942023091120240611;0003997952023091120240611;0003997962023091120240611;000399797202
3091120240611;0003997982023091120240611;0003997992023091120240611;0003998002023091120240611;
0003998012023091120240611;0003998022023091120240611;0003998032023091120240611;0003998042023
091120240611;0003998052023091120240611;0003998062023091120240611;0003998072023091120240611;0
003998082023091120240611;0003998092023091120240611;0003998102023091120240611;00039981120230
91120240611;0003998122023091120240611;0003998132023091120240611;0003998142023091120240611;00
03998152023091120240611;0003998162023091120240611;0003998172023091120240611;000399818202309
1120240611;0003998192023091120240611;0003998202023091120240611;0003998212023091120240611;
Funivia-Staz. di
Valle-Trento;2939;07:00;07:15;07:30;07:45;08:00;08:15;08:30;08:45;09:00;09:30;10:00;10:30;11:00;11:30;11:4
5;12:00;12:15;12:30;12:45;13:00;13:15;13:30;13:45;14:00;14:15;14:30;15:00;15:30;16:00;16:30;17:00;17:15;17
:30;17:45;18:00;18:15;18:30;18:45;19:00;19:15;19:30;19:45;20:00;20:15;20:30;21:00;21:30;22:00;22:30;07:00;
07:15;07:30;07:45;08:00;08:15;08:30;08:45;09:00;09:30;10:00;10:30;11:00;11:30;11:45;12:00;12:15;12:30;12:4
5;13:00;13:15;13:30;13:45;14:00;14:15;14:30;15:00;15:30;16:00;16:30;17:00;17:15;17:30;17:45;18:00;18:15;18
:30;18:45;19:00;19:15;19:30;19:45;20:00;20:15;20:30;21:00;21:30;22:00;22:30;
Funivia-Staz. di
Monte-Sardagna;2940;07:05;07:20;07:35;07:50;08:05;08:20;08:35;08:50;09:05;09:35;10:05;10:35;11:05;11:35;
11:50;12:05;12:20;12:35;12:50;13:05;13:20;13:35;13:50;14:05;14:20;14:35;15:05;15:35;16:05;16:35;17:05;17:2
0;17:35;17:50;18:05;18:20;18:35;18:50;19:05;19:20;19:35;19:50;20:05;20:20;20:35;21:05;21:35;22:05;22:35;07
:05;07:20;07:35;07:50;08:05;08:20;08:35;08:50;09:05;09:35;10:05;10:35;11:05;11:35;11:50;12:05;12:20;12:35;
12:50;13:05;13:20;13:35;13:50;14:05;14:20;14:35;15:05;15:35;16:05;16:35;17:05;17:20;17:35;17:50;18:05;18:2
0;18:35;18:50;19:05;19:20;19:35;19:50;20:05;20:20;20:35;21:05;21:35;22:05;22:35;
```

Copy the result (.csv snippet) and paste it inside `resources/annotatedtimetable/FunA-2023091120240611-annotated.csv` file after line 8.



```
Descr.breve:;Fu;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Descr.lunga:;FUNIVIA TRENTO-SARDAGNA;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Validità:;ORARIO Trento Inv. 2023-24   Valido dal 11/09/2023 al 11/06/2024;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Orario:;ORARIO FERIALE;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Frequenza:;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Linea:;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Servizio:;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
gtfs trip_id;;0003997712023091120240611;0003997732023091120240611;0003997752023091120240611;000399776202309112024061
smartplanner route_id;;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUTSA;FUT
service_id;;0000001402023091120240611;0000001402023091120240611;0000001402023091120240611;0000001402023091120240611;
stops;stop id;0003997712023091120240611;0003997732023091120240611;0003997752023091120240611;000399776202309112024061
```

Save the file. One has to run the following command to generate timetables for the return course (SARDAGNA-TRENTO).

```
root@229c03ad87f7:/cache-generator/annotated.timetable.generator# java -cp
annotated-timetable-generator.jar sayservice.FuniviaGenerator r
```

Likewise the result of this command to be pasted inside
`resources/annotatedtimetable/FunR-2023091120240611-annotated.csv` file.

Error Handling

At times the GTFS data is unaligned indicating the wrong direction of courses. According to the GTFS specification for the trips.txt file (see the link below), the `direction_id` field indicates the direction of travel for a trip.

https://developers.google.com/transit/gtfs/reference#tripstxt

This field is being used in routing and it provides a way to separate trips by direction when publishing time tables. Valid options are:

0 - Travel in one direction (e.g. outbound travel).
1 - Travel in the opposite direction (e.g. inbound travel).

In some cases, within the GTFS data of Trento, the TRENTO-SARDAGNA trips are written with the wrong `direction_id`. This requires manual adjustment of the local GTFS `trips.txt` file inside the `resources/gtfs/12` directory to correctly generate the timetables. In particular, it would be advisable to follow these steps:

1. find, inside the `routes.txt` file, the route_id of the route named `Funivia Trento Sardagna` (by default it is 531);
2. search, inside the `trips.txt` file, the trips whose route_id is the one found in the previous step;
3. for each record, ensure that if the `trip_headsign` field (text that appears on signage identifying the trip's destination to riders) is `Funivia-Staz. di Monte-Sardagna`, the `direction_id` should be set to 0. Conversely, if the 'trip_headsign' is `Funivia-Staz. di Valle-Trento`, the `direction_id` should be set to 1.

5. Smartplanner GTFS split

In this section we introduce a new application smartplanner which is the principal multimodal routing developed in the context of mobility value added service of Trentino region. We will cover different aspects of the engine in the coming sections. The engine is a standalone web application with a rich set of tools for data manipulation. In the context of this step we will use the engine to split and align the opened data portal GTFS. The GTFS provided on an open data portal included urban routes of Trento, Rovereto, and Lavis region. The smart planner engine by design supported different regions. The Trento Urban region is identified by unique agency Id `12` while Rovereto urban is identified by agency Id `16`. For the sake of this tutorial, it is required to post-process the GTFS available on open data portal of Trentino using a script of smartplanner engine which will extract Trento Urban GTFS. The generated zip file is

12.zip

As envisaged within the scope of this tutorial, we will be working over the Trento GTFS file generated by smartplanner identified as `12.zip`. In the next section we detail the step by step process of splitting the original GTFS file.

Structure

The executable `smart-planner.jar` is located inside the folder `smartplanner`. The tool structured is as follows

```
|- smart-planner.jar
|- fare
|- opendata
|- gen
`- tn-routemodel.json
```

Copy the GTFS zip file downloaded from open data portal inside `opendata` folder.

Configuration

The configuration file is `tn-routemodel.json` which defines the mapping between GTFS routeId and smartplanner routeIds. Often new routeIds are introduced inside the GTFS file which needs to be mapped properly in this configuration file. For e.g. Bus route `16` previously modeled as single route Id `484` was recently changed. In addition to routeId `484`, open data portal GTFS introduced another routeId `617` with route name `P`. The two routeIds map to single bus route 16, therefore inside the `tn-route model.json` file the route is configured as follows.

```
"routeMappings": {
        "484_12_0": "16A",
        "484_12_1": "16R",
        "617_12_0": "16A",
        "617_12_1": "16R",
```

This file is required to be modified only in case the GTFS routeId for a bus is changed in the GTFS data. The `fare` folder contains the GTFS fare attributes and rules information as provided on open data portal.

Execute

In order to generate Trento urban GTFS `12.zip`, run the following command with parameters.

```
root@229c03ad87f7:/cache-generator/smartplanner# java -cp smart-planner.jar
it.sayservice.platform.smartplanner.utils.TrentoGTFS -c true -a 12 -f opendata/google_transit_urbano.zip -o
gen -c true -t true -r true
```

The parameters usage can be seen by running the command

```
root@229c03ad87f7:/cache-generator/smartplanner# java -cp smart-planner.jar
it.sayservice.platform.smartplanner.utils.TrentoGTFS

usage: TrentoGTFS -a <AGENCIES> -c <CALENDAR> -f <GTFS> [-h] [-o  <DIR>] -r <FARE> -t <TRANSFER>
[-z <OLDGTFS>]

Split and align the GTFS files for different agencies
 -a,--agencies <AGENCIES>     List of agency IDs
 -c,--calendar <CALENDAR>   GTFS has calendar file
 -f,--gtfs <GTFS>                   Zip file containing GTFS data
```

```
-h,--help                   Print this help message
-o,--output <DIR>           Output directory, default current directory
-r,--fare <FARE>            GTFS has fares information
-t,--transfer <TRANSFER>    GTFS has calendar file
-z,--old <OLDGTFS>          Old zip file with GTFS for direction references
```

For the sake of this tutorial it is sufficient to run the command as shown in first place. If everything works fine, the Trento urban GTFS zip file `12.zip` will be generated inside the `gen` folder.

Error handling

Often errors like unmapped routeIds are shown which require adding the routeId in the `tn-routemodel.json` file. There exist other types of error related to `shape` files which can be avoided in case the route belongs to another region like Rovereto or Lavis.

In order to check the consistency of the generated zip file it is recommended to validate it using the GTFS FeedValidator tool.

https://github.com/google/transitfeed/wiki/FeedValidator

The instructions for performing validation can be found on the link above. In the next section, the tutorial will cover the approach of Train GTFS generation followed in the context of smartplanner engine. The generated train GTFS can also be validated using the `FeedValidator` tool as mentioned above.

6. Train GTFS

In this step we will demonstrate the process of generating the GTFS for Train Lines Trento-Bassano, Verona-Bolzano, and Trento-Malè using the smartplanner engine toolkit. The smarplanner `CSVGTFSGenerator` developed in this context reads the input CSV files annotated with trips, stops, and time information. The tool also requires a `trip.txt` to be present that maps the trip information to service dates. In the context of smartplanner engines the train lines are identified with agency_id 5 for Verona-Bolzano, agency_id 6 for Trento-Bassano, and agency_id 10 for ferrovia FTM (Trento-Malè). The GTFS available on opendata portal does not include or partly include these lines and for this reason in the context of smartplanner engine, `CSVGTFSGenerator` tool was developed to generate GTFS using the PDFs files provided for Trento-Malè and Trento-Bassano lines on Trentino trasporti official website:

https://www.trentinotrasporti.it/viaggia-con-noi/ferrovia

For the Verona-Bolzano line the pdfs can be downloaded from the Trentino province website:

https://www.provincia.tn.it/Documenti-e-dati/Documenti-di-supporto/L-orario-sulla-linea-Bolzano-Verona-Porta-Nuova-valido-dal-15-settembre-2023

Structure

The tool is structured as follows:

```
|- smart-planner.jar
|- fare
|- opendata
|- gen
|-5
|-6
|-10
`- tn-routemodel.json
```

The folder `5` corresponds to agency_id 5 for the Verona-Bolzano train line and contains all the data for GTFS generation. The folder `6` corresponds to agency_id 6 for Trento-Bassano train line while folder `10` corresponds to agency_id 10 for Trento-Malè train line.

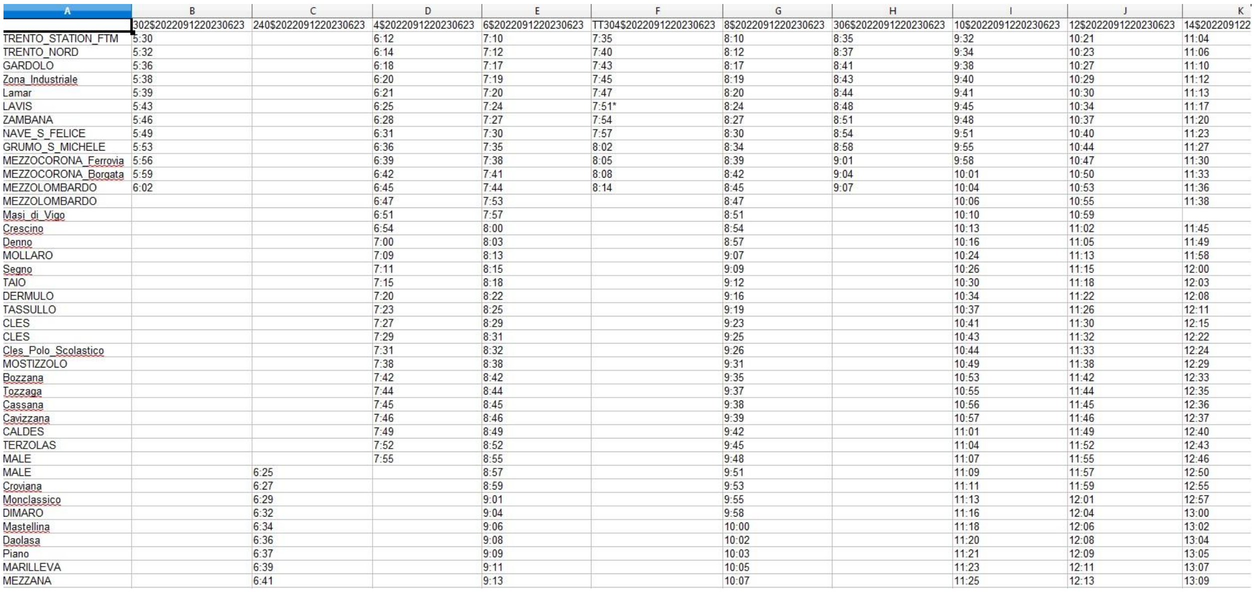
PDF to CSV conversion

The first step involves converting each file from .pdf to .csv. There are many possible ways to perform this task. The simplest of them is to copy the PDF inside a text editor and replace the empty spaces with commas as in the csv. It is important to note that the delimiter must be a comma (**,**) and not colon (**:**).
The names of the csv files are as follows:

- for the Verona-Bolzano train line (agency_id 5), the files must be called **ORARI_BRENNERO_ANDATA.csv** and **ORARI_BRENNERO_RITORNO.csv**
- for the Trento-Bassano train line (agency_id 6), the files must be called **ORARI_VALSUGANA_ANDATA.csv** e **ORARI_VALSUGANA_RITORNO.csv**
- for the Trento-Malè train line (agency_id 10), the files must be called **ORARI_FTM_ANDATA.csv** e **ORARI_FTM_RITORNO.csv**

Input CSV Structure

The input .csv files contained in the **5**, **6**, and **10** folders look like the following image.
As can be seen from the image, the first column is for stops. The name assumed by each of them must be the `stop_id` field of the corresponding record contained in the `stops.txt` file of the folder (`**5**`,` **6**` or `**10**`). To shorten the time needed to add the stops in a csv file, we recommend starting from the



A	B	C	D	E	F	G	H	I	J	K
	302$2022091220230623	240$2022091220230623	4$2022091220230623	6$2022091220230623	TT304$2022091220230623	8$2022091220230623	306$2022091220230623	10$2022091220230623	12$2022091220230623	14$202209122
TRENTO_STATION_FTM	5:30		6:12	7:10	7:35	8:10	8:35	9:32	10:21	11:04
TRENTO_NORD	5:32		6:14	7:12	7:40	8:12	8:37	9:34	10:23	11:06
GARDOLO	5:36		6:18	7:17	7:43	8:17	8:41	9:38	10:27	11:10
Zona_Industriale	5:38		6:20	7:19	7:45	8:19	8:43	9:40	10:29	11:12
Lamar	5:39		6:21	7:20	7:47	8:20	8:44	9:41	10:30	11:13
LAVIS	5:43		6:25	7:24	7:51*	8:24	8:48	9:45	10:34	11:17
ZAMBANA	5:46		6:28	7:27	7:54	8:27	8:51	9:48	10:37	11:20
NAVE_S_FELICE	5:49		6:31	7:30	7:57	8:30	8:54	9:51	10:40	11:23
GRUMO_S_MICHELE	5:53		6:36	7:35	8:02	8:34	8:58	9:55	10:44	11:27
MEZZOCORONA_Ferrovia	5:56		6:39	7:38	8:05	8:39	9:01	9:58	10:47	11:30
MEZZOCORONA_Borgata	5:59		6:42	7:41	8:08	8:42	9:04	10:01	10:50	11:33
MEZZOLOMBARDO	6:02		6:45	7:44	8:14	8:45	9:07	10:04	10:53	11:36
MEZZOLOMBARDO			6:47	7:53		8:47		10:06	10:55	11:38
Masi_di_Vigo			6:51	7:57		8:51		10:10	10:59	
Crescino			6:54	8:00		8:54		10:13	11:02	11:45
Denno			7:00	8:03		8:57		10:16	11:05	11:49
MOLLARO			7:09	8:13		9:07		10:24	11:13	11:58
Segno			7:11	8:15		9:09		10:26	11:15	12:00
TAIO			7:15	8:18		9:12		10:30	11:18	12:03
DERMULO			7:20	8:22		9:16		10:34	11:22	12:08
TASSULLO			7:23	8:25		9:19		10:37	11:26	12:11
CLES			7:27	8:29		9:23		10:41	11:30	12:15
CLES			7:29	8:31		9:25		10:43	11:32	12:22
Cles_Polo_Scolastico			7:31	8:32		9:26		10:44	11:33	12:24
MOSTIZZOLO			7:38	8:38		9:31		10:49	11:38	12:29
Bozzana			7:42	8:42		9:35		10:53	11:42	12:33
Tozzaga			7:44	8:44		9:37		10:55	11:44	12:35
Cassana			7:45	8:45		9:38		10:56	11:45	12:36
Cavizzana			7:46	8:46		9:39		10:57	11:46	12:37
CALDES			7:49	8:49		9:42		11:01	11:49	12:40
TERZOLAS			7:52	8:52		9:45		11:04	11:52	12:43
MALE			7:55	8:55		9:48		11:07	11:55	12:46
MALE		6:25		8:57		9:51		11:09	11:57	12:50
Croviana		6:27		8:59		9:53		11:11	11:59	12:55
Monclassico		6:29		9:01		9:55		11:13	12:01	12:57
DIMARO		6:32		9:04		9:58		11:16	12:04	13:00
Mastellina		6:34		9:06		10:00		11:18	12:06	13:02
Daolasa		6:36		9:08		10:02		11:20	12:08	13:04
Piano		6:37		9:09		10:03		11:21	12:09	13:05
MARILLEVA		6:39		9:11		10:05		11:23	12:11	13:07
MEZZANA		6:41		9:13		10:07		11:25	12:13	13:09

contents of the corresponding csv file already present in the folder (`**5**`,` **6**` or `**10**`) and replacing only the previously formatted times.

Another important part of each csv file is the first row. Within this row, each of the cells belonging to a trip column contains the ID of that trip, which is taken from the `trip_id` field of the corresponding record contained in the `trips.txt` file located in the same directory (`**5**`,` **6**` or `**10**`). In fact, each trip in the `trips.tx` file corresponds to the respective time column inside the csv file.

In order to correctly identify the trip_id of a columand to clarify any doubts that may arise during the course of this part of the tutorial, it is necessary to provide a description of the fields contained within the 'trip.txt' files located within the directories labeled **5**, **6**, and **10**. Each trip is required to have the following set of information:

```
route_id,service_id,trip_id,trip_headsign,direction_id,shape_id,wheelchair_accessible
```

- route_id: this field can assume only two values because each train line has two routes. One identifies the travel in one direction (e.g. the travel from Bolzano to Verona Porta Nuova) and the other identifies the travel in the opposite direction (e.g. the travel from Verona Porta Nuova to Bolzano)
- service_id: information about this field can be found in the following section
- trip_id: for each train line, the id of a trip is structured as follows: trip_headsign + $ + start date of validity of the timetable (yyyyMMdd) + end date of validity of the timetable (yyyyMMdd). E.g. RV3461$2020121320210612
- trip_headsign: text that appears on signage identifying the trip's destination to riders. E.g. RV3461
- direction_id: indicates the direction of travel for a trip. In this case, the assumed value (0 or 1) depends on the `route_id` field, since, as described before, each train line has two routes, one for each direction
- shape_id: identifies a geospatial shape that describes the vehicle travel path for a trip. As for the previous field, the assumed value depends on the `route_id` field
- wheelchair_accessible: indicates wheelchair accessibility

So, to identify the trip_id of a column, it is essential to use the `trip_headsign` field, which is equal to the value that can be found at the top of each trip column within the timetable pdf files.

Service identifier

In the scope of this tutorial, the most important field to specify is `service_id` that identifies a set of dates when a trip is available for the corresponding route. This field is foreign key mapped to its reference inside the calendar.txt file (service_id) or calendar_dates.txt file (service_id). This information is fetched from the download PDF. For example, service_id WES_SEPGIU is a service offered from September to June on weekdays (MON-SAT) as indicated in the `calendar.txt` file:

```
service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date,end_date
WES_SEPGIU,1,1,1,1,1,1,0,20220912,20230623
```

The same service is not offered on weekdays that occur on festivals (for e.g May 1 Labor day) as indicated in the `calendar_dates.txt` file:

```
service_id,date,exception_type
WES_SEPGIU,20230501,2
```

For more information about GTFS specification please refer to:

```
https://gtfs.org/
```

Execute

In order to generator GTFS for **Verona-Bolzano** line run the following command:

```
root@229c03ad87f7:/cache-generator/smartplanner# java -cp smart-planner.jar
it.sayservice.platform.smartplanner.utils.CSVGTFSGenerator 5
```

For **Trento-Bassano** line run the following command:

```
root@229c03ad87f7:/cache-generator/smartplanner# java -cp smart-planner.jar
it.sayservice.platform.smartplanner.utils.CSVGTFSGenerator 6
```

For **Trento-Malè** line run the following command:

```
root@229c03ad87f7:/cache-generator/smartplanner# java -cp smart-planner.jar
it.sayservice.platform.smartplanner.utils.CSVGTFSGenerator 10
```

The generated GTFS .zip files can be found in respective folders: `**5**`,` **6**` or `**10**`.

Error handling

There can be errors in case of missing tripIds inside the .csv file or trips.txt file. The output GTFS .zip file must be validated using the GTFS validation tools. Furthermore, the consistency of schedules can be checked using the ScheduleViewer tool provided with the GTFS Validation tools or at runtime after cache generation.

7. Opentripplanner (OTP) Graph Generation

The Smart Planner engine is realized on top of an open source project OpentripPlanner, enhancing its basic functionality with the real-time information, planning flows customization, and implementing the required smart mobility and environment sustainability features. The open trip planner (OTP) is a core multimodal engine based on java which identifies the extent of the area needed by the GTFS dataset to build a routing graph based on search algorithm. It exposes API which is then used by smartplanner engine to grab the necessary data coverage(trip, route information) during the cache generation phase.

Configuration

Copy all the GTFS .zip files generated in previous steps inside the **otp/trentino** folder.

Build graph

In order to build the routing graph, run from the **otp** root.

```
root@229c03ad87f7:/cache-generator/otp# java -Xmx4G -Duser.timezone=Europe/Rome -jar otp.jar
--basePath . --build trentino
```

The generated graph can be found inside the `trentino` folder. The OTP instance is already up and running in the launched container environment on port 7575. The instance will update automatically after the execution of build command since it is set to run in autoReload mode so any changes made to the graph object will be applied automatically at run time. The container shell must print the following log trace after the reloading of the graph.

```
otp           | 14:08:26.680 INFO (GrizzlyServer.java:121) Grizzly server running.
otp           | 14:09:02.537 INFO (InputStreamGraphSource.java:189) Router ID 'trentino' graph input
modification detected, force reload.
otp           | 14:09:02.539 INFO (InputStreamGraphSource.java:213) Loading graph...
```

The running instance of open trip planner engine (OTP) on port 7575 will be used by smartplanner engine to make API calls to obtain route trip information during cache generation phase.

8. Smartplanner Cache

SmartPlanner engine is an open source information system that provides multimodal transportation planning and real-time mobility data, obtained from the service providers and generated by the community of users. The engine provide support for real time trip updates, static and real time transit trip data information, famous transportation modes in metropolitan cities such as car sharing, carpooling, bike sharing, and provide efficient communication channel with the community of users

Structure

The tool structure contains two important folders (src, trentino) for cache-generation phase.

```
|- smart-planner.jar
|- fare
|- opendata
|- gen
|-5
|-6
|-10
|-src
| `- main
|     `-resources
|        ` trentino.yaml
|-trentino
| |- cache
| |  |- annotated
| |     `- 12
| |  |- client
| |      |- 5
| |      |- 6
| |      |- 10
| |      |- 12
| |      |- routesdb_trento.zip
| |      `- routesdb_trento_extended.zip
| |   `- schedules
| |- config
| |- gtfs
`- tn-routemodel.json
```

The `src\main\resources` folder contains the configuration YAML file for smartplanner. The other important folder `trentino` contains subfolders such as `cache`, `client`, and `schedules` which are used during the execution of cache generation. The generated cache at the end of this step will be located inside the `trentino/cache/client` folder:

```
routesdb_trento.zip
routesdb_trento_extended.zip
```

The files are preconfigured, so we proceed with cache generation.

Configure

Copy all the annotated csv files generated in section 5 of this tutorial (*CSV annotation*) from the folder

```
`\annotated.timetable.generator\resources\annotatedtimetable\12`
```

to the destination folder

```
`smartplanner\trentino\cache\annotated\12`
```

Rename the copied file

```
%20C_A-Feriale-2023091120240611-annotated
%20C_R-Feriale-2023091120240611-annotated
```

as

```
_Ca-Feriale-2023091120240611-annotated.csv
_Cr-Feriale-2023091120240611-annotated
```

Copy the GTFS zip files generated in previous steps (5.zip, 6.zip,10.zip, and 12.zip) inside folder

```
`smartplanner\trentino\gtfs`
```

Clean the following folder.

```
smartplanner\trentino\cache\client
```

Execute

The cache generation tool is dependent on running instances of mongodb and otp (opentripplanner) server. Generate cache running the following set of commands from the root `cache-generator\smartplanner`:

```
root@229c03ad87f7:/cache-generator/smartplanner# export OTP_HOME=$PWD
root@229c03ad87f7:/cache-generator/smartplanner# java -cp smart-planner.jar
it.sayservice.platform.smartplanner.utils.CacheGenerator
```

The generated cache files can be copied from the `client` folders to the destination.

Prune (Optional)

In case of error during cache generation, it is required to clean mongo-data volume and `client` folder before successive `cache-generation` attempts. The simple way to prune volumes is to launch the container clean every time or remove the volume forcefully using docker command.

```
..\cache-generator> docker volume ls
DRIVER    VOLUME NAME
local     318357a12890022bd1c3aec146769dc3a91a09c4f5c005d2ec6f3c8802376b5a
local     cache-generator_mongo-data
PS C:\Users\engg_\Desktop\cache-generator>docker volume rm cache-generator_mongo-data
```

Another possible way is to connect to mongo container and clean the `trentino` collection

```
..\..> docker exec -it mongo /bin/bash
root@0df3145e6407:/# mongo
MongoDB shell version v3.6.2
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.2
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
Server has startup warnings:
2023-11-02T13:53:27.466+0000 I STORAGE  [initandlisten]
2023-11-02T13:53:27.466+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly
recommended with the WiredTiger storage engine
2023-11-02T13:53:27.467+0000 I STORAGE  [initandlisten] **       See
http://dochub.mongodb.org/core/prodnotes-filesystem
2023-11-02T13:53:27.521+0000 I CONTROL  [initandlisten]
2023-11-02T13:53:27.521+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the
database.
2023-11-02T13:53:27.521+0000 I CONTROL  [initandlisten] **       Read and write access to data and
configuration is unrestricted.
2023-11-02T13:53:27.521+0000 I CONTROL  [initandlisten]
2023-11-02T13:53:27.522+0000 I CONTROL  [initandlisten]
2023-11-02T13:53:27.522+0000 I CONTROL  [initandlisten] ** WARNING:
```

```
/sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2023-11-02T13:53:27.522+0000 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2023-11-02T13:53:27.522+0000 I CONTROL  [initandlisten]
2023-11-02T13:56:23.324+0000 E -        [main] Error loading history file: FileOpenFailed: Unable to fopen() file
/root/.dbshell: No such file or directory
> use trentino
switched to db trentino
> db.dropDatabase()
{ "ok" : 1 }
>
```

9. Repository

The source code of the cache-generator tool is available on the following link:

https://github.com/smartcommunitylab/pt-data-generator