



Principes d'extractions de données des sources
et
Principe d'insertion des données extraites

AlpesTransport

Mohammed ROUABAH
Ilyes ZEGHDALLOU
William MAILLARD

01/03/2023

1 Principe de l'extraction des données des sources

1.1 Rappels sur les sources de données

Nous disposons des trois sources de données suivantes (avec leurs format de données respectifs) :

1. [SNCF](#)
 - `.csv` pour les référentiels des gares et les tarifs
 - GTFS pour les horaires
2. [STAS](#)
 - `.xml` et GTFS pour les informations sur les horaires et les arrêts des lignes de bus
 - webscrapping sur le site de la [STAS](#) pour récupérer les tarifs
3. [RegionRhôneAlpes](#)
 - GTFS pour les arrêts et les horaires des lignes de bus de la région

1.2 Principe général de l'extraction de données

Du fait de l'hétérogénéité des formats des données, chaque type de fichier de chaque source est associé à un parseur qui extrait les informations pertinentes pour le projets (i.e lié aux : trajets, arrêts, horaires et tarifs). Ces parseurs effectuent les actions ordonnées suivantes :

1. Chargement des données,
2. Nettoyage des données,
3. Transformation des données en un format intermédiaire unique.

1.3 Chargement des données par un parseur

Pour les documents structurés (`.csv` , GTFS , `.xml`) les parseurs utilisent la fonction `read` correspondant au type du document (`read_csv`, `read_gtfs`, `read_xml`) de la librairie python **pandas**.

Cette fonction permet de charger les données dans des *dataframes*, qui sont des tableaux structurés possédants des colonnes nommés. Ces noms de colonnes sont utilisés lors du processus d'extraction de données.

Pour le *webscrapping*, le parseur récupère la page à l'aide d'une requête de type GET au site et en extrait le contenu (dans notre cas le tableau des tarifs), qui peut ensuite être converti en *dataframe* comme le reste des données (en le considérant comme un document xml).

1.3.1 Un exemple avec les documents de la SNCF

```
gares_df.head().transpose()
```

	0	1	2
Code plate-forme	00002-1	00004-1	00006-1
Code gare	2	4	6
Code UIC	87988709	87785006	87784884
Date fin validité plateforme	NaN	NaN	NaN
Intitulé plateforme	Remise à Jorelle	Cerbère	Ur - Les Escaldes
Code postal	93140.0	66290.0	66760.0
Code Commune	10.0	48.0	218.0
Commune	Bondy	Cerbère	Ur
Code département	93.0	66.0	66.0
Département	Seine-Saint-Denis	Pyrénées-Orientales	Pyrénées-Orientales
Longitude	2.487751	3.163403	1.940482
Latitude	48.89317	42.441773	42.457481
Segment DRG	b	c	c
Niveau de service	NaN	1.0	1.0
RG	GARES B IDF LIGNE T4	GARES C LANGUEDOC ROUSSILLON	GARES C LANGUEDOC ROUSSILLON
TVSs	[["TVS_Code": "RULT"]]	[["TVS_Code": "CERT"]]	[["TVS_Code": "URL"]]
SOPs	NaN	NaN	NaN
Gare	("DRG_ON": true, "Etrangere_ON": false, "NbPL": 1, "NbP": 1)	("DRG_ON": true, "Etrangere_ON": false, "NbPL": 1, "NbP": 1)	("DRG_ON": true, "Etrangere_ON": false, "NbPL": 1, "NbP": 1)
Intitulé gare	Remise à Jorelle	Cerbère	Ur - Les Escaldes
Intitulé fronton de gare	Remise à Jorelle	Cerbère	Ur - Les Escaldes
Gare DRG	True	True	True
Gare étrangère	False	False	False
DIC	DOIF	DRG Occitanie Sud	DRG Occitanie Sud
Région SNCF	REGION DE PARIS-EST	REGION LANGUEDOC-ROUSSILLON	REGION LANGUEDOC-ROUSSILLON
Unité gare	NaN	UG Est Occitanie	UG Est Occitanie
UT	BONDY GARE REMISE A JORELLE TRAM TRAIN	CERBERE GARE	UR LES ESCALDES GARE
Nbre plateformes	1	1	1
TVS	RUL	CER	URL
WGS 84	48.89317, 2.487751	42.441773, 3.163403	42.457481, 1.940482

FIGURE 1 – Chargement des données des gares de la sncf

```
tarifs_df.head().transpose()
```

	0	1	2
Région	BOURGOGNE FRANCHE-COMTE	CENTRE	BOURGOGNE FRANCHE-COMTE
Origine	VILLENELVE SUR YO	CHATEAU RENAULT	VILLENELVE SUR YO
Origine - code UIC	87683219	87574665	87683219
Destination	AUXERRE ST GERVAI	CHATEAUDUN	AUXERRE ST GERVAI
Destination - code UIC	87683573	87545756	87683573
Libellé tarif	Abonnement mensuel BFC	BILLET REMI	Tarif Normal
Type tarif	Abonnement tout public	Tarif normal	Tarif normal
Prix	106.4	15.0	11.0

FIGURE 2 – Chargement des données des tarifs de la sncf

1.4 Nettoyage des données par le parseur

Les données ainsi chargés sont ensuite filtrés afin de ne conserver que les informations qui seront insérés dans la WIKIBASE. Lors de cette étape, les parseurs vont aussi nettoyer les données, pour que chaque valeur ait une valeur correcte selon son type.

1.4.1 Un exemple avec les documents de la SNCF

```
gare_region_df.head().transpose()
```

	11	87
Code plate-forme	00062-1	00422-1
Code gare	62	422
Code UIC	87783175	87734707
Intitulé plateforme	Saint-Flour - Chaudes-Aigues	Lavoite-sur-Loire
Code postal	15100.0	43800.0
Code Commune	187.0	119.0
Commune	Saint-Flour	Lavoite-sur-Loire
Code département	15.0	43.0
Département	Cantal	Haute-Loire
Longitude	3.106286	3.90541
Latitude	45.035912	45.12145
TVSs	[["TVS_Code", "SFC"]]	[["TVS_Code", "LVL"]]
SOPs	NaN	NaN
Gare	["DRG_ON": true, "Etrangere_ON": false, "NbPit...]	["DRG_ON": true, "Etrangere_ON": false, "NbPit...]
Intitulé gare	Saint-Flour - Chaudes-Aigues	Lavoite-sur-Loire
Intitulé fronton de gare	Saint-Flour - Chaudes-Aigues	Lavoite-sur-Loire
Gare DRG	True	True
Gare étrangère	False	False
DTG	DRG AURA-BFC	DRG AURA-BFC
Région SNCF	REGION AUVERGNE	REGION AUVERGNE
Unité gare	UG Auvergne	UG Auvergne

FIGURE 3 – Nettoyages des données des gares de la sncf

```
tarifs_region_df = tarifs_df[tarifs_df["Région"] == "AUVERGNE RHONE-ALPES"]
```

```
tarifs_region_df.head().transpose()
```

	32	35	38	41
Région	AUVERGNE RHONE-ALPES	AUVERGNE RHONE-ALPES	AUVERGNE RHONE-ALPES	AUVERGNE RHONE-ALPES
Origine	BRION MONTREAL CL	BRION MONTREAL CL	BRION MONTREAL CL	BRION MONTREAL CL
Origine - code UIC	87131961	87131961	87131961	87131961
Destination	BELLEGARDE GARE R	LYON PART DIEU	LYON PART DIEU	BELLEGARDE S/V LY
Destination - code UIC	87698407	87723197	87723197	87742726
Libellé tarif	Abonnement TER ilico Hebdo	Abonnement TER ilico Mensuel	Billet Tarif Normal Régional	Billet Tarif Normal Régional
Type tarif	Abonnement tout public	Abonnement tout public	Tarif normal	Tarif normal
Prix	18.9	18.7	18.8	6.9

FIGURE 4 – Nettoyages des données des tarifs de la sncf

1.4.2 Les données nettoyés d'un fichier gtfs de la sncf

Le format gtfs est une archive *.zip* de fichier *.txt*, dont les données sont reliées entre elles par des id, qui établissent la liaison entre les fichier. Voici ces liens :

- route.txt est lié à trips.txt par route_id
- trips.txt est lié à stop-times.txt par trip_id
- stop-times.txt est lié à stop.txt par stop_id

	0	1	2	3	4
route_id	FRLine:00F2577A-6A87-43E0-95F3-07351E4B2F6	FRLine:00F208C-C8BC-4521-A792-ECC3AB85811	FRLine:0128E1D5-9183-4D58-B1CF-F5A5A6A4037	FRLine:0202671B-7107-429E-A37B-473C53E0254C	FRLine:022877D9-D121-4D0B-B808-FE217331866b
route_long_name	Bessing - Sarreguemines	Saint-Etienne - Roanne	Marseille - Toulon - Hyeres	Montpellier Saint-Roch - Avignon Centre	Angers St Laud - Cholet

FIGURE 5 – données du fichier routes.txt

	0	1	2	3
route_id	FRLine:85d3579c-996b-4671-89af-839855ede78az	FRLine:85d3579c-996b-4671-89af-839855ede78az	FRLine:85d3579c-996b-4671-89af-839855ede78az	FRLine:85d3579c-996b-4671-89af-839855ede78az
trip_id	OCSN105330F16460242023-02-08T00:42:08Z	OCSN105342F10746732023-02-08T00:42:08Z	OCSN105347F8289142023-02-08T00:42:08Z	OCSN105347F18739962023-02-08T00:42:08Z
direction_id	1	1	1	1

FIGURE 6 – données du fichier trips.txt

	0	1	2	3
stop_id	StopAreaOCE80142893	StopPointOCETrain TER-80142893	StopAreaOCE80142901	StopPointOCETrain TER-80142901
stop_name	Appenweier	Appenweier	Legelshurst	Legelshurst
stop_lat	48.541964	48.541964	48.558617	48.558617
stop_lon	7.973221	7.973221	7.913533	7.913533
parent_station	NaN	StopAreaOCE80142893	NaN	StopAreaOCE80142901

FIGURE 7 – données du fichier *stops_times.txt*

	0	1	2
trip_id	OCSN105330F1646024-2023-02-08T00:42:00Z	OCSN105330F1646024-2023-02-08T00:42:00Z	OCSN105342F1074673-2023-02-08T00:43:00Z
arrival_time	07:28:00	07:33:00	17:22:00
stop_id	StopPointOCENavette-87571240	StopPointOCENavette-87571000	StopPointOCENavette-87571240

FIGURE 8 – données du fichier *stop.txt*

1.5 Transformation en un format intermédiaire

Après avoir été chargé et nettoyé, les données sont ensuite transformer dans un format commun avant de pouvoir être insérer dans la wikibase. Cela permet d'être plus modulaire et de faciliter l'insertion de données provenant de sources diverses.

Le format intermédiaire choisi est un dictionnaire python comme ci-dessous :

```
new_product = {
    'entity': 'item',
    'label': 'Nom du produit',
    'description': 'Description du produit',
    'property': [
        {
            'entity': 'property',
            'type': 'string',
            'label': 'Prix',
            'value': '10,99 €'
        },
        {
            'entity': 'property',
            'type': 'url',
            'label': 'Image',
            'value': 'http://example.com/image.jpg'
        }
    ]
}
```

FIGURE 9 – Principe

1.6 Extraction de données des autres sources

1.6.1 Extraction de données STAS

Pour les données de la STAS, la première étape consiste à parser les fichiers *.xml* en format *.csv* afin de les rendre exploitables. Cela permet d'extraire les données brutes des fichiers *.xml* et de les transformer en un format plus facile à traiter.

Ensuite, il est nécessaire de fusionner les différentes sources de données pour obtenir une source complète. Cette étape peut impliquer la suppression de doublons et la normalisation des

données pour assurer la cohérence de l'ensemble.

0	1	1	M1	BELLEVU...	3	005BAA	nan	E1-0-1-31	1	Eglise c	0	nan
1	1	1	M1	BELLEVU...	3	005BAA	nan	E1-0-1-31	10	Eglise co...	0	nan
2	1	1	M1	BELLEVU...	3	005BAA	nan	E1-0-1-31	100	Bellevue	1	nan
3	1	1	M1	BELLEVU...	3	005BAA	nan	V1-0-1-31	1000	Bellevue	1	nan
4	1	1	M1	BELLEVU...	3	005BAA	nan	V1-0-1-31	1001	Bellevue	1	nan
5	1	1	M1	BELLEVU...	3	005BAA	nan	V1-0-1-31	1002	Bellevue	1	nan
6	1	1	M1	BELLEVU...	3	005BAA	nan	V1-0-1-31	1003	Bellevue	1	nan
7	1	1	M1	BELLEVU...	3	005BAA	nan	V1-0-1-31	1004	Bellevue	1	nan
8	1	1	M1	BELLEVU...	3	005BAA	nan	V1-0-1-31	1005	Bellevue	1	nan
9	1	1	M1	BELLEVU...	3	005BAA	nan	V1-0-1-31	1006	Bellevue	1	nan

FIGURE 10 – Source de données

Pour les tarifs, On extrait le tableau des tarifs à partir de la page web, puis on insère son contenu dans un dataframe avant de le transformer dans le format intermédiaire.

Titre de transport	Pour qui ?	Prix	Détails
1 mois tout public	Vous utilisez la STAS plusieurs fois par semaine.	47,00 €	Voir détail
1 Voyage	Vous utilisez la STAS de temps en temps.	1,60 € Tarif applicable au 01/03/2023	Voir détail
1 mois -20 ans	Vous utilisez la STAS plusieurs fois par semaine et vous avez moins de 20 ans.	10,00 €	Voir détail
1 mois Retraité 60 ans et +	Vous utilisez la STAS plusieurs fois par semaine et vous êtes retraité de 60 ans et +	10,00 €	Voir détail
Formule Liberté	Vous avez une carte OÛRAI et vous utilisez la STAS occasionnellement, vous souhaitez bénéficier du tarif le plus avantageux sans contrainte.	1,20 € (prix du voyage, Inscription gratuite) Tarif applicable au 01/03/2023	Voir détail
Voyage Souplesse	Vous avez une carte OÛRAI et vous voyagez occasionnellement, vous souhaitez bénéficier du tarif le plus avantageux.	1,20 € Tarif applicable au 01/03/2023	Voir détail
1 mois demandeur d'emploi	Vous utilisez la STAS plusieurs fois par semaine et vous êtes demandeur d'emploi sous certaines conditions.	10,00 €	Voir détail

FIGURE 11 – Tableau des tarifs sur la page web de la stas

```

<div id="result-fares">
  <table class="table fare-table table-bordered table-striped fare-table-mono">
    <caption>Tous les profils - Toutes les fréquences</caption>
    <thead>
      <tr>
        <th class="col-classic text-center" scope="col">Titre de transport</th>
        <th class="col-classic text-center" scope="col">Pour qui ?</th>
        <th class="col-classic text-center" scope="col">Prix</th>
        <th class="col-classic text-center" scope="col">Détails</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td></td>
        <td class="hidden-sm hidden-xs cw-visible-print content_adm"></td>
        <td class="text-center hidden-sm hidden-xs cw-visible-print content_adm"></td>
        <td class="text-center hidden-print"></td>
      </tr>
      <tr>
        <td></td>
        <td class="hidden-sm hidden-xs cw-visible-print content_adm"></td>
        <td class="text-center hidden-sm hidden-xs cw-visible-print content_adm"></td>
        <td class="text-center hidden-print"></td>
      </tr>
    </tbody>
  </table>
</div>

```

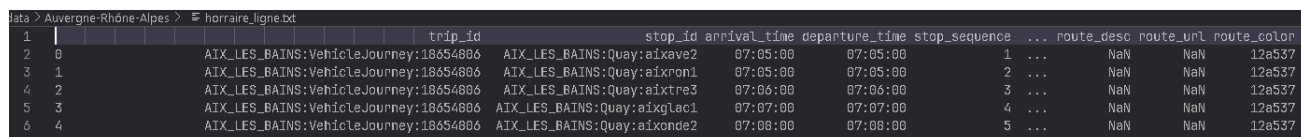
FIGURE 12 – Structure html du tableau des tarifs extrait

Enfin, la dernière étape consiste à rendre cette nouvelle source de données compatible avec le modèle de données choisi. Cela implique de mapper les propriétés de la source de données

aux propriétés correspondantes du modèle de données, et de s'assurer que les données sont correctement structurées pour répondre aux exigences du modèle.

1.7 Extraction de données Rhônes-Alpes

Pour les données de Rhône-Alpes dans un premier temps on récupère les données dans un format gtfs, ensuite nous effectuons une fusion ainsi qu'une mise en relation des données en elle-même pour les rendre exploitable, il en résulte un fichier txt contenant les données a exploité. A ce moment la les données peuvent être envoyer dans la wikibase.



	trip_id	stop_id	arrival_time	departure_time	stop_sequence	route_desc	route_url	route_color
0	AIX_LES_BAINS:VehicleJourney:18654886	AIX_LES_BAINS:Quay:aixave2	07:05:00	07:05:00	1	NaN	NaN	12a537
1	AIX_LES_BAINS:VehicleJourney:18654886	AIX_LES_BAINS:Quay:aixron1	07:05:00	07:05:00	2	NaN	NaN	12a537
2	AIX_LES_BAINS:VehicleJourney:18654886	AIX_LES_BAINS:Quay:aixtre3	07:06:00	07:06:00	3	NaN	NaN	12a537
3	AIX_LES_BAINS:VehicleJourney:18654886	AIX_LES_BAINS:Quay:aixglac1	07:07:00	07:07:00	4	NaN	NaN	12a537
4	AIX_LES_BAINS:VehicleJourney:18654886	AIX_LES_BAINS:Quay:aixonde2	07:08:00	07:08:00	5	NaN	NaN	12a537

FIGURE 13 – Données après extraction dans un dataframe

2 Principe de l'insertion de données dans la wikibase

Une fois le *preprocessing* des données complétés, nous disposons de ces dernières dans un format homogène (i.e dictionnaire python). On va ensuite lire ces structures de données et insérer les items et propriétés qu'elles contiennent dans la *wikibase* à l'aide de la librairie **wiki-baseIntegrator**.

Cela implique de réaliser l'association entre les données extraites avec entités (item, propriété, valeur) correspondantes dans la Wikibase, et de créer une requête d'insertion pour chaque élément de données.

Pour l'envoi des données dans la wikibase, plusieurs solutions on été testées, notamment en utilisant un bot afin de rentrer les données. Dans un premier temps le bot se connecte a l'api de la wikibase, ensuite des requêtes sont effectués pour créé les items, puis les propriété qui lui sont associées son insérés et reliés cet item.