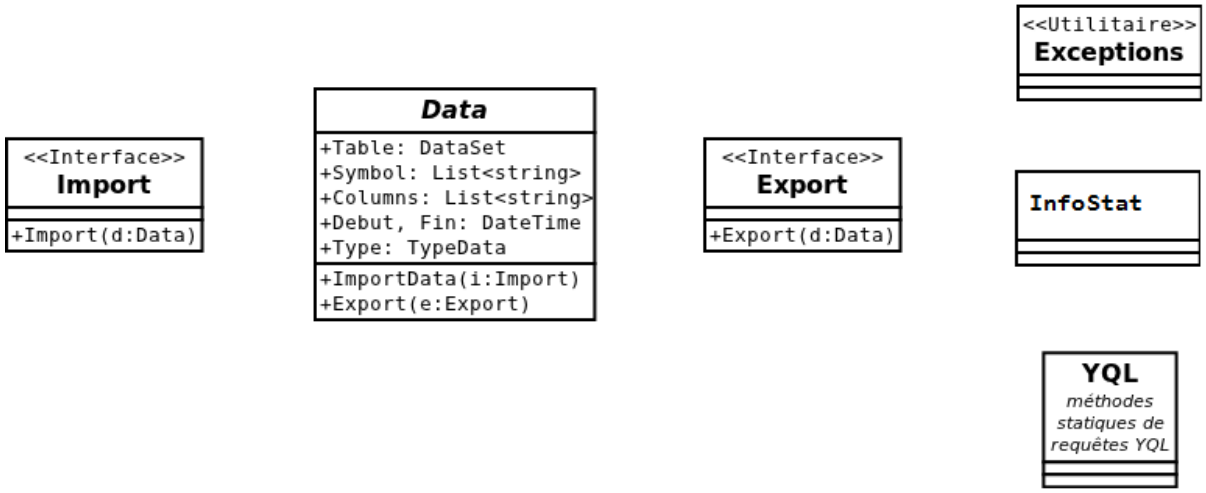
APIFIMAG : DOCUMENTATION TECHNIQUE

ARCHITECTURE DE L'API

DIAGRAMME GLOBAL



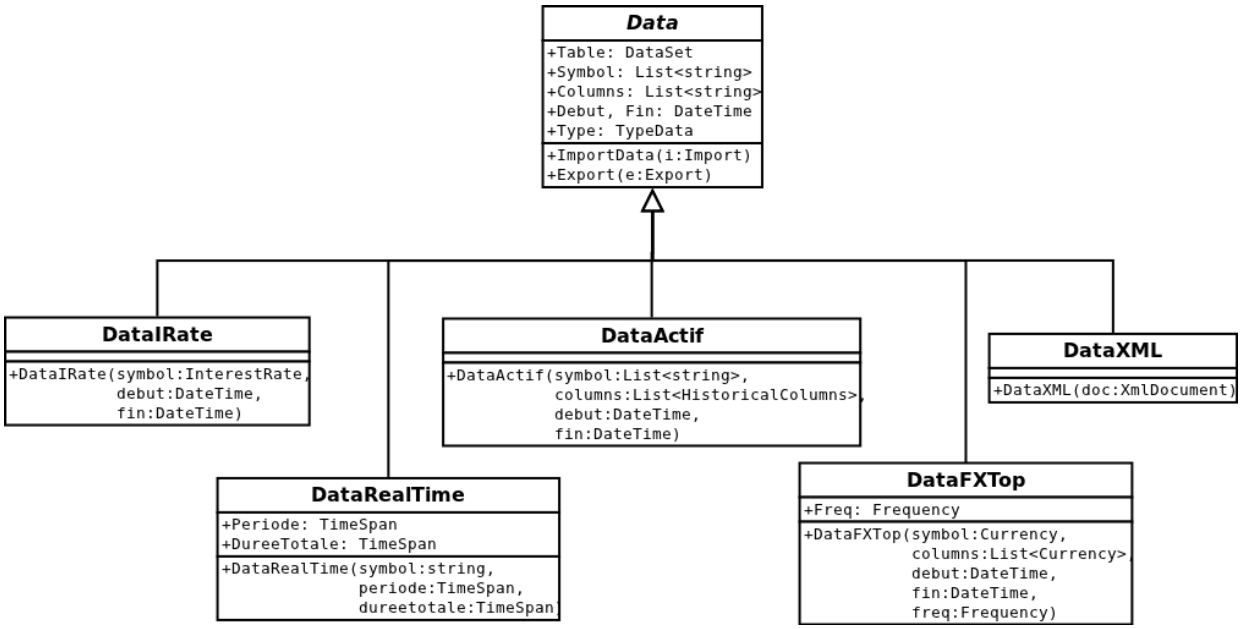
Les éléments principaux de l'API sont une classe de stockage de données `Data`, et deux interfaces, `Import` et `Export` dont la relation est gérée avec le patron de conception Visiteur.

A part, nous avons une classe YQL qui fournit sous forme de méthodes statiques toutes les requêtes YQL nécessaires à notre projet, notamment lors de la construction de l'arbre des actifs et des acquisitions en temps réel.

La classe `InfoStat`, s'occupe du téléchargement des informations statiques des sociétés.

Une classe utilitaire fournit les exceptions pour la programmation défensive.

CLASSE DATA



`Data` est une classe abstraite servant à stocker les données importées.

Pour chaque type de données que l'on veut récupérer il existe une sous-classe dotée des attributs nécessaires et implémentant un constructeur adéquat.

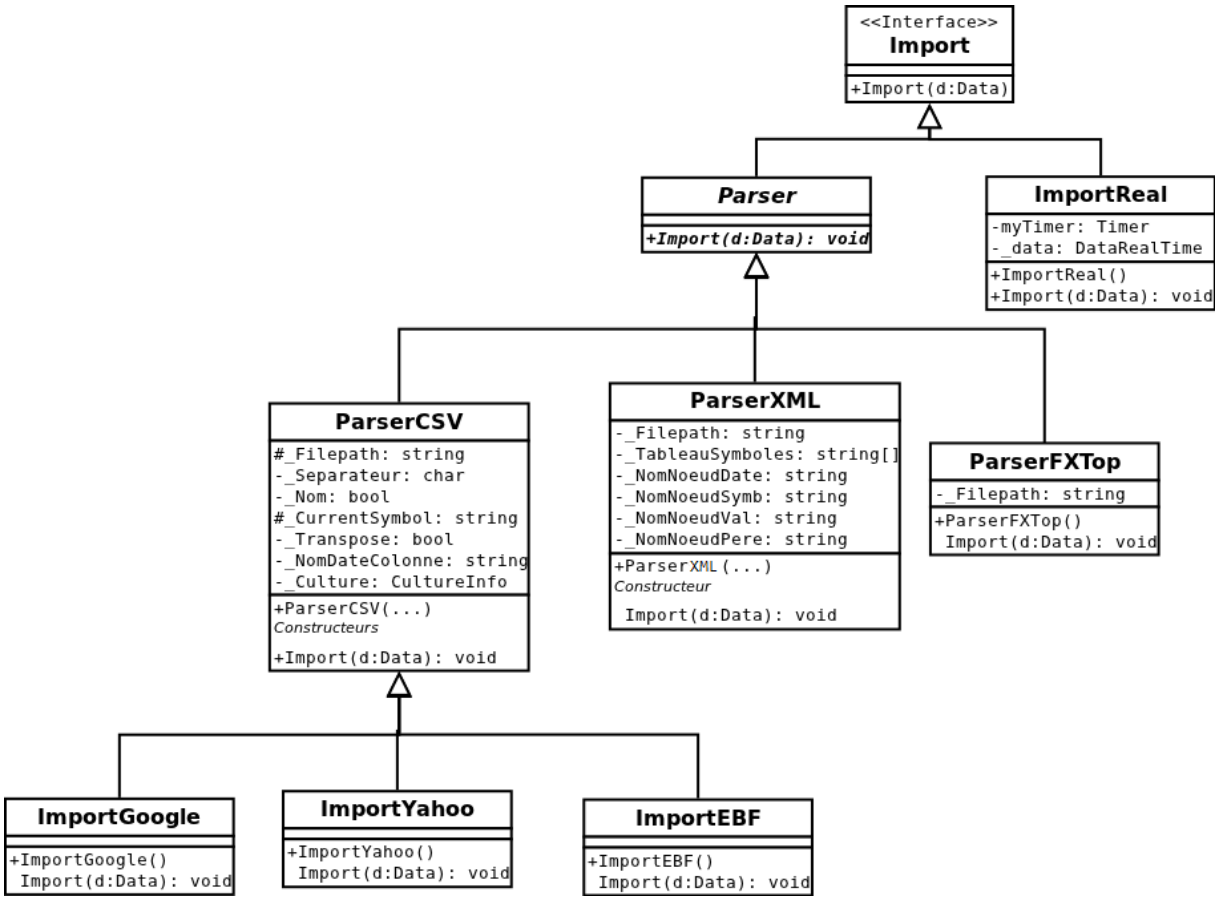
Il est donc facile de rajouter d'autres types de données sans perturber l'API dans sa globalité, il suffit de créer une nouvelle sous-classe héritant de Data.

`DataXML` permet de parser un document XML de configuration pour automatiser une acquisition de données historiques sur des actifs : elle récupère les symboles, les colonnes et les dates voulues puis construit l'objet `Data`.

Des types énumérés sont utilisés pour définir :

- Le type de Data : (historique, taux de change, temps réel, taux d'intérêts)
- Données Historiques : Les différentes colonnes d'informations qu'il est possible de demander (par exemple high, low, close ...)
- Taux de change : La fréquence (daily, monthly, yearly), et la liste exhaustive des devises.
- Taux d'intérêts interbancaires : Les différents taux (EURIBOR, EONIA ...)

INTERFACE IMPORT

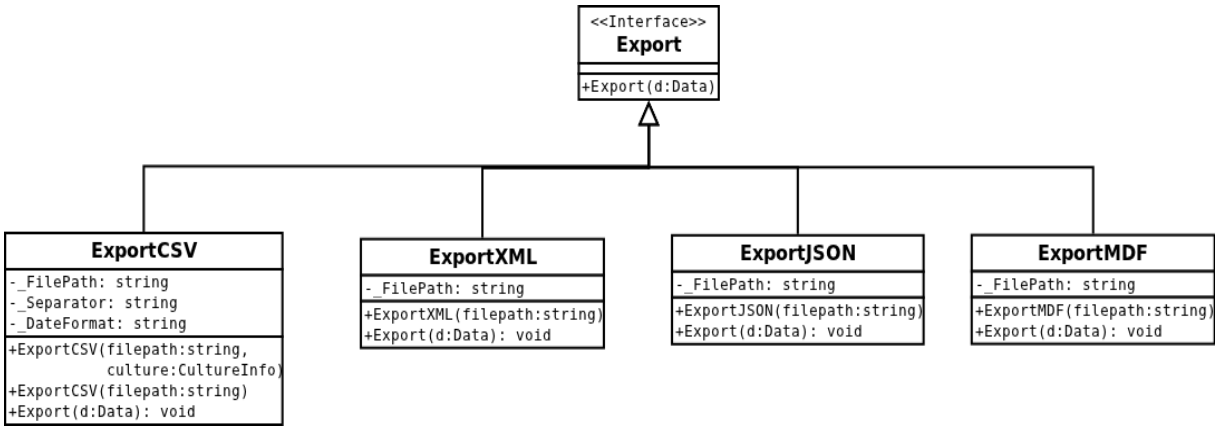


Nous avons ici une construction en héritages successifs due aux différents niveaux de spécialisation des classes, afin de faciliter l'ajout de sources d'import.
En effet, il est ici possible d'ajouter une source d'import en créant une sous-classe au bon endroit, selon le besoin ou non d'un parseur.

Nous n'utilisons pas le parseur XML car nos sources (Google Finance, Yahoo Finance, Euribor EBF) nous fournissent des fichiers .csv à parser, cependant il est mis à disposition dans l'éventualité où quelqu'un voudrait utiliser une source fournissant des fichiers XML.

`ImportReal` s'occupe des acquisitions en temps réel en utilisant les méthodes de la classe YQL.
`ParserFXTop` récupère les taux de change en parsant la page html spécifique au site `fxtop.com`

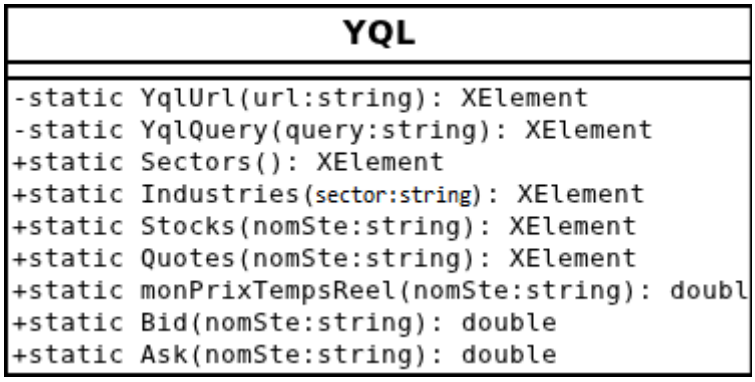
INTERFACE EXPORT



Au niveau de l'export la construction est la plus simple possible, une classe par format voulu, implémentant l'interface d'export.
Les méthodes pour les formats JSON et MDF ont été récupérées directement des projets précédents.

N'importe qui peut donc ajouter un format d'exportation à l'API facilement.

CLASSE YQL

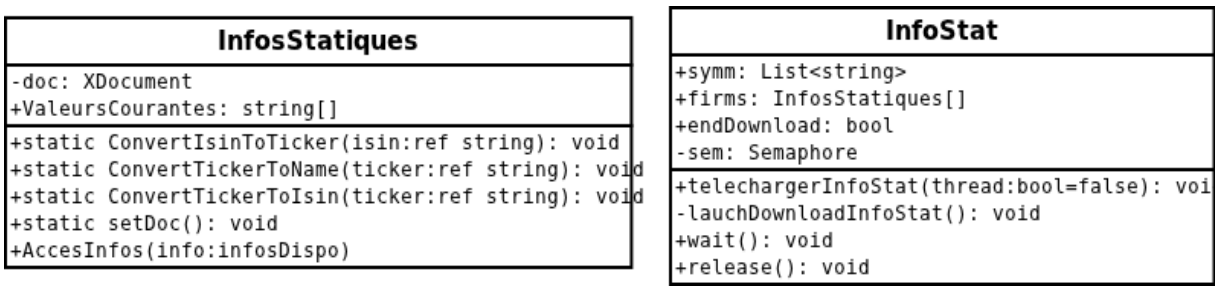


Cette classe a été quasiment intégralement récupérée du projet XDataFi.

Les méthodes publiques `Sectors`, `Industries`, `Quotes` et `Stocks` permettent de construire l'arbre des actifs dans l'interface, classés par secteur, et de récupérer les informations statiques sur les sociétés cotées en bourse.

Pour les acquisitions en temps réel, ce sont les méthodes `monPrixTempsReel`, `Bid` et `Ask` qui fournissent les bonnes requêtes YQL.

INFORMATIONS STATIQUES SUR LES SOCIETES



Nous avons récupéré ces classes du projet XDataFi, elles sont indépendantes du reste de l'API.

Elles permettent de récupérer les informations statiques des sociétés cotées en bourse afin de les afficher dans l'interface.

Pour effectuer ces téléchargements en parallèle des acquisitions de données, un thread utilisant un sémaphore est utilisé.

`InfoStat` est un singleton disposant d'une liste d'instances d'`InfosStatiques`, une par société.

On utilise un type énuméré correspondant aux différentes informations qu'il est possible d'extraire (nom de la société, secteur, nombre d'employés ...)

GESTION DES EXCEPTIONS

En cas de mauvaise utilisation l'API lève des exceptions :

- Acquisition dont la date de fin est antérieure à la date de début
- Erreur de connexion à l'url générée lors d'une importation
- Erreur de typage (exemple : on demande des taux de changes en renseignant « GOOG »)
- Valeur négative pour une donnée positive