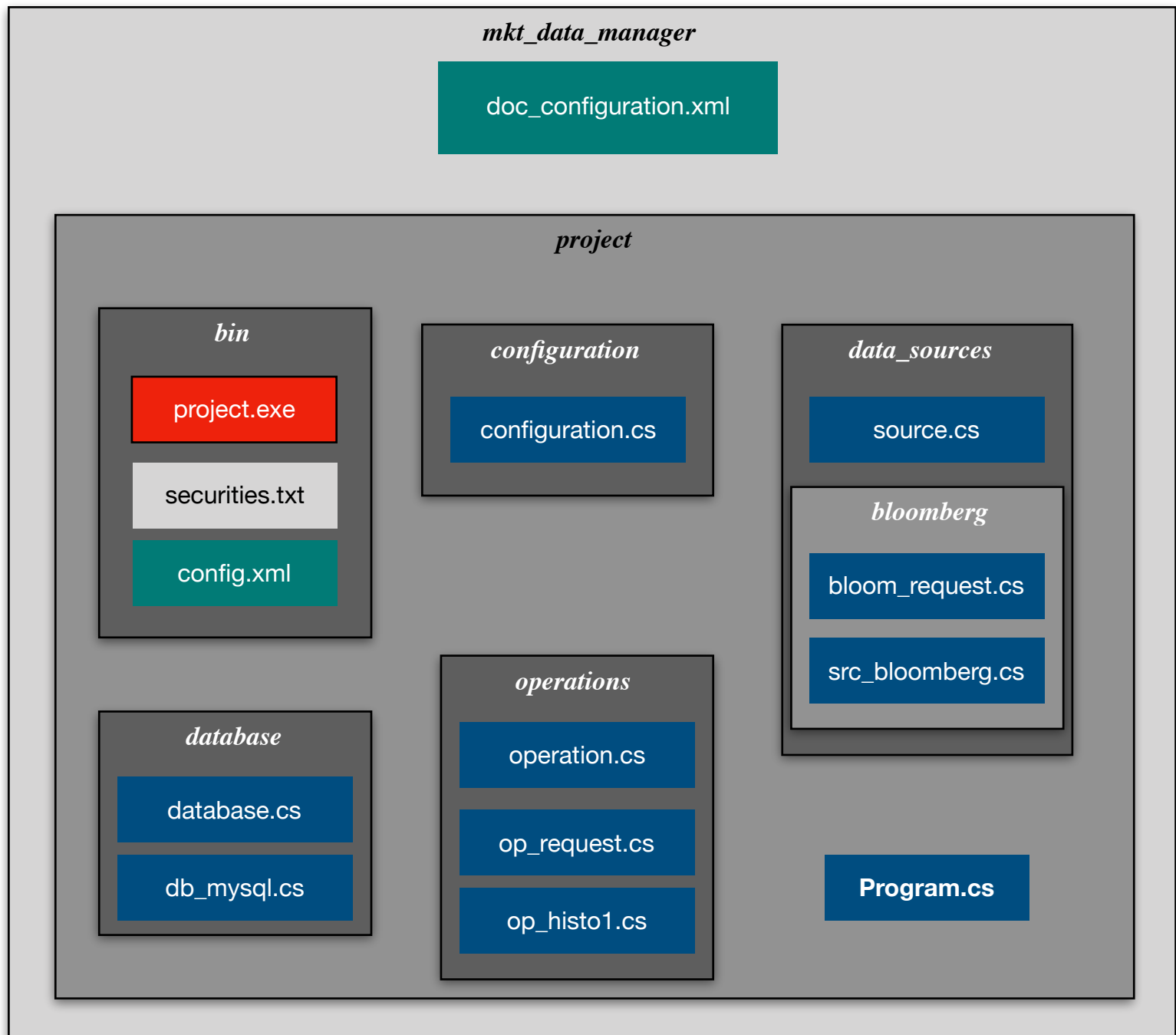


# Guide maintenance/dev

## Architecture du projet

---



Folder

Sql file

XML file

C# file

Executable

## Quelques détails sur le code

---

project.exe

Ce fichier est l'exécutable du projet, il suffit de cliquer dessus pour lancer le script.

securities.txt

Ce fichier contient tous les tickers dont on veut récupérer les données. Ils doivent être notés ligne par ligne. Il faudra écrire « IBM US Equity » pour le ticker IBM et non pas uniquement « IBM » sinon le script ne fonctionnera pas.

doc\_configuration.xml

config.xml

Il ne faut pas modifier le fichier doc\_configuration.xml. C'est un template, une documentation, qui permet de comprendre comment configurer un projet. Le fichier qui contient la configuration effective du projet est le fichier config.xml. On y mets les opérations que l'on souhaite exécuter et les informations de la source (bloomberg) et des bases de données.

Program.cs

Ce fichier est le 'main' du projet. Il fait appel aux classes qui vont permettre d'effectuer les opérations que l'on souhaite.

configuration.cs

Ce fichier contient la classe CONFIGURATION qui va permettre de lire le fichier config.XML et ainsi d'instancier les classes SOURCE, OPERATION et DATABASE.

source.cs

Contient la classe SOURCE qui prend en paramètre : le nom de l'opération que l'on souhaite, la liste des tickers dont on veut récupérer les données, les champs de l'API

Bloomberg que l'on souhaite récupérer pour chacun des tickers, et enfin les paramètres de la requête qu'on va effectuer (date de début, date de fin)

src\_bloomberg.cs

bloom\_request.cs

Ces fichiers contiennent des classes qui permettent de se connecter à l'API de Bloomberg et de faire des requêtes sur celle-ci afin de récupérer les informations dont on a besoin.

database.cs

db\_mysql.cs

Ces fichiers contiennent des classes qui permettent de se connecter et d'insérer dans la base de données qui va accueillir les informations issues de Bloomberg.

operation.cs

op\_request.cs

op\_histo1.cs

operation.cs : Contient une classe OPERATION qui contient une méthode qui permet d'exécuter une opération.

op\_request.cs : Contient le code qui permet de récupérer les informations de l'API de Bloomberg pour les envoyer dans la base de données.

op\_histo1.cs : spécialisation de OP\_REQUEST qui ajoute un ID formé par la concaténation de de ticker et date.

### **Option update :**

Pour qu'elle fonctionne, il faut créer une table qui porte le même nom que la table où se trouvent les données, en ajoutant "\_update" à la fin.

Cette table doit contenir deux champs : security (clé primaire) et date (date de dernière mise à jour, moins deux jours par sécurité).

La fonction update récupère la date de dernière mise à jour de chaque security et prend la date minimale. Cette date sert de date de début pour la requête.

Enfin la fonction update met à jour la table update avec la date d'aujourd'hui moins deux jours.

Pour les tickers qui n'ont jamais été insérés en base, il faut lancer le script en créant un fichier securities.txt qui contient uniquement que les tickers n'ayant jamais été insérés, puis par la suite on pourra lancer le script avec un fichier securities.txt avec l'ensemble de tous les tickers

## Bases de données

---

**Database name :** XXXX

**Tables :** name, name\_update

### name

Field	Type	Key
ID	varchar(40)	PRIMARY
ticker	varchar(8)	
date	datetime	
last	varchar(40)	
last_adj	varchar(40)	
volume	varchar(40)	
short_int	varchar(40)	

### name\_update

Field	Type	Key
ticker	varchar(8)	PRIMARY
date	DATETIME	