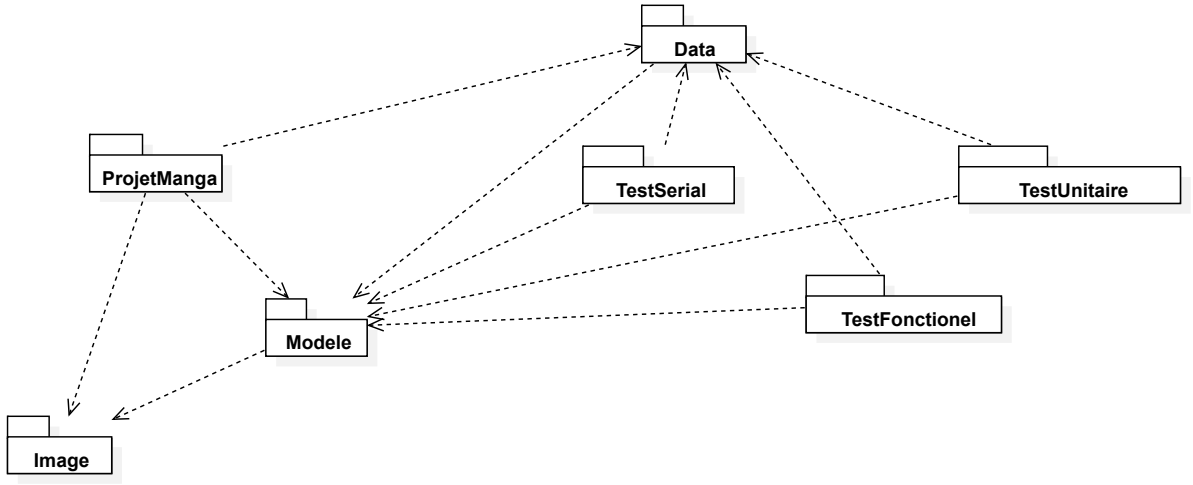
Louis Perret
Simon Dallet
3H-4H
IUT 2020/2021

Documentation Projet Tutoré

Sommaire

Diagramme de paquetages avec persistance

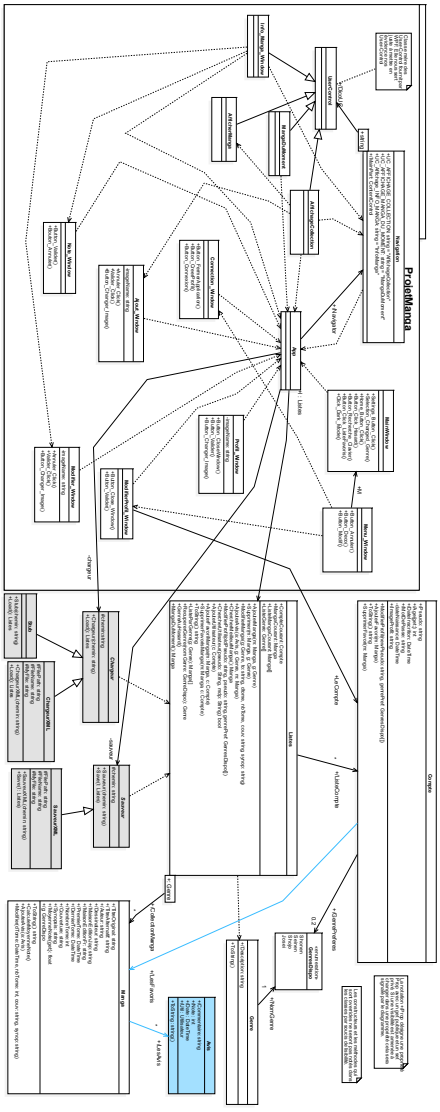


Data : Bibliothèque de classes contenant nos classes pour la persistance
TestSerial : Permet de tester notre chargement et sauvegarder

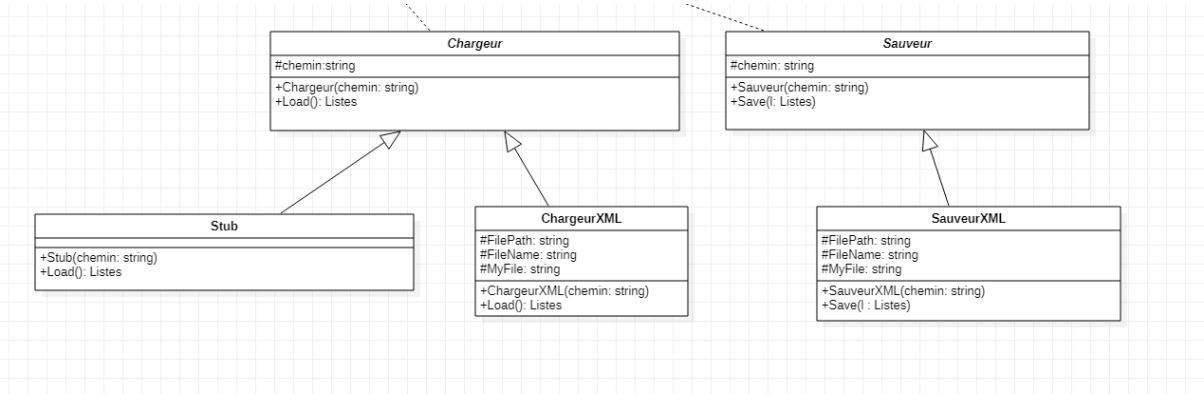
Nous avons décidé de séparer Data des autres paquetages dans un assemblage différent car cela nous permet de regrouper plusieurs manière de persister nos données indépendamment de notre modèle et de notre vue et ainsi de pouvoir choisir la manière de persister que l'on veut.
Pour la description des autres paquetages : se référencer à notre documentation conception orientée objet

Diagramme de classes avec persistance et nos partie ajoutées

Ce qui est en gris concerne la partie persistance de notre application
Et tout ce qui est en bleu concerne nos ajouts personnels



Description de notre persistance



Ceci est la partie de notre diagramme de classes concernant le chargement et la sauvegarde.
Nos classes Chargeur et Sauveur sont des classes abstraites et contiennent chacune un constructeur et une méthode qui leur est propre. Cette stratégie de conception nous permet donc avec l'héritage de spécialiser quel type de chargement ou de sauvegarde nous voulons. Si on veut passer par une base de données, par exemple, il nous suffira de généraliser ces deux classes avec leur méthode respective qui nous permettra de charger et sauvegarder avec une base de données.
De plus, grâce à l'héritage il nous ai facile de pouvoir effectuer du polymorphisme en prenant les classes mères et en changeant seulement le type dynamique pour changer notre manière de persister.

Classe Sauveur
Cette classe abstraite va permettre de sérialiser notre instance de Listes.
Elle possède une méthode qui reçoit un objet Listes en paramètre pour le serialiser.

Classe Chargeur
Cette classe va permettre de charger nos données à partir d'un objet Listes

Classe Stub
Cette classe simule un faux chargement de données qui nous est utile pour réaliser nos tests fonctionnels, unitaires et de notre binding de notre application.

ChargeurXML
Classe qui est spécialisée dans le chargement de nos données à partir d'un fichier XML.

SauveurXML
Classe qui est spécialisée dans la sauvegarde de nos données sous format xml.

Description de nos parties personnalisées

Nos parties ajoutées concernent la classe Avis qui permet de représenter la note avec le commentaire laissé par un utilisateur sous un manga.
Notre classe manga possède donc bien évidemment une collection d'Avis qui représente l'ensemble de ses avis. Cette collection nous permet de faire la moyenne des notes du manga qui est représentée par son attribut MoyenneNote. Cet attribut nous est utile avec la méthode MangaDuMoment dans notre classe Listes qui retourne le manga ayant la meilleure moyenne de notes.
Notre dernière partie ajoutée est la propriété LesFavoris contenue dans Compte qui représente la collection des manga préférés d'un compte.

Pour plus d'explication sur ces classes se référencer sur notre documentation conception orientée objet