



Notre projet se divise en deux parties, d'un côté les clients et de l'autre l'API. Il y a trois possibilités pour la consommer, c'est-à-dire trois moyens de récupérer ou modifier ses données.

Le premier est une application Console, appelée Application HTTP Client.

Le deuxième est une application Mobile codée en MAUI.

Le troisième est le Swagger, une documentation pour les Web API.

Toutes les applications clientes communiquent avec l'API, déployée dans un conteneur Docker, et lui envoient des requêtes HTTP, qui permettent de récupérer, ajouter, supprimer, modifier des données, et plus encore. L'application Console utilise pour se faire la classe `HttpClient`.

De son côté, l'API utilise les Controller de chaque classe afin de satisfaire les requêtes de l'utilisateur. Ces Controller vont assembler certaines données s'ils en ont besoin, et utiliser les méthodes de l'interface `IDataManager`. Cette dernière appelle celles de ses filles, les siennes n'étant pas implémentées. Celles-ci vont alors permettre de modifier, ajouter, supprimer ou récupérer les données de la couche de Persistance.

Celle-ci est dans notre cas composée du Stub et de notre Base de Données, et va rassembler nos données sous diverses formes.

Le fait d'appeler les méthodes de la mère (`IDataManager`), pour utiliser celles de filles (Par exemple `StubData`), va permettre à la partie Persistance d'être interchangeable, et ce sans changer aucune ligne de code dans celle-ci.