

Ce document :

- ☞ doit être ouvert avec un lecteur PDF supportant les liens hypertextes ;
- ☞ est amené à évoluer, les modifications pouvant être suivies sur son **dépôt Github**.

Table des matières

I	Les tags	2
II	Principes généraux des API	3
2.1	Proposition de principe général	4
2.2	Avantages futurs en terme de gestion de projet	4
2.3	L'API « tags »	5

I LES TAGS

II

PRINCIPES GÉNÉRAUX DES API

2.1 Proposition de principe général

Les différents modules du portail peuvent chacun renvoyer à des acteurs, pratiques et enjeux spécifiques, dépendre de bases de données qui leurs sont propres ou encore être écrits dans des langages différents. Cette hétérogénéité fonctionnelle et technique ne doit cependant pas obombrer le fait que les modules sont amenés à manipuler les mêmes concepts et objets documentaires, sociaux et scientifiques (planche, spécimen, détermination, collection, unité de transcription, récolteur/teuse, utilisateur/trice, sous-partie d'intérêt d'une planche, etc.). Ainsi, s'il est prévu que les modules déploient des outils métier particuliers justifiant leurs existences autonomes, il est cependant vital qu'ils puissent consommer les données produites par leurs pairs. Au surplus, la couverture fonctionnelle d'un module pourra être naturellement amenée à évoluer durant la vie du projet (pour hypothèse : l'utilité d'une extension des Herbonautes pourrait être révélée d'ici un ou deux ans suite à une première série de tests, laquelle conduirait à la collecte d'un nouveau type de données dont l'exploitation serait également pertinente dans les autres modules). Pour ces raisons, nous préconisons que chaque module propose une API *exposant l'intégralité des données qu'il permet de récolter*. Ces APIs ne devraient pas être conçues en s'adaptant aux spécifications émanant des autres modules dénotant les besoins respectifs de ceux-ci à un temps donné, mais devraient plus simplement poursuivre le but d'exposer des données « brutes » structurées dans les termes métier des modules auxquels elles sont rattachées, sans présupposer des usages et attentes possibles en d'autres lieux du portail (voir la figure [1.1](#)).

2.2 Avantages futurs en terme de gestion de projet

Ce découplage des APIs des modules fournisseurs de données par rapport aux besoins des modules tiers consommateurs de données possède des vertus sur le plan de la gestion de projet. Ainsi, si un module fournisseur M_1 attend qu'un module consommateur M_2 spécifie les termes et structures sous lesquels il souhaite pouvoir accéder à un sous-ensemble des données de M_1 , l'API proposée par M_1 pourra se trouver inadéquate pour une future demande identique émanant d'un autre module M_3 . Dans ce cas, l'API définie par M_1 pour M_2 posera des problèmes assimilable à une dette technique pour toute extension à d'autres modules consommateurs. Si au contraire les modules fournisseurs exposent chacun une API complète « brute », la gestion des APIs ne constitue plus un ensemble de sous-projets bloquants transversaux supposant une co-validation de plusieurs partenaires, mais une simple extension fonctionnelle de chacun des modules, une

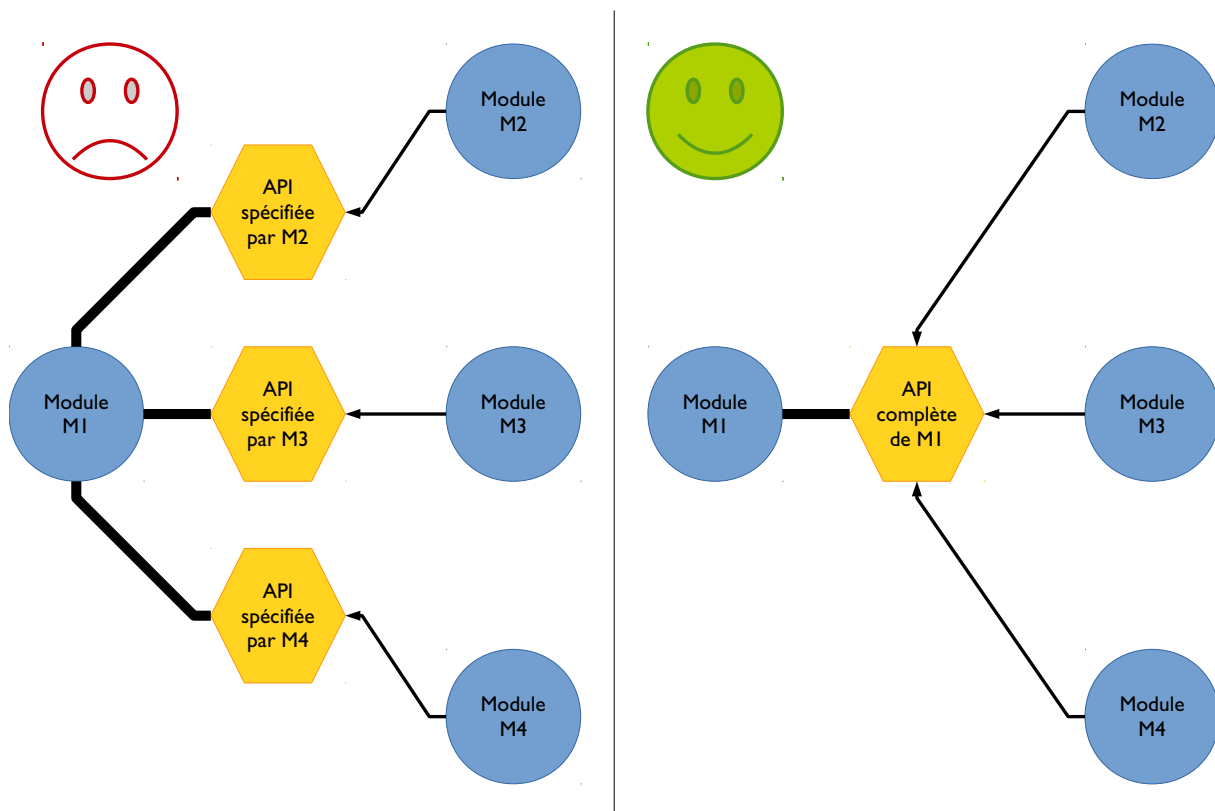


Figure 1.1 Une API complète par module

simple modalité supplémentaire des fonctions relevant des aspects « *retrieval* » des opérations « CRUD ». Les modules développés par DICEN reposeront exclusivement sur ce principe. Les clients Web légers du collaboratoire et de la visite virtuelle seront alors les premiers « modules » consommateurs des APIs définies par le composant serveur exposant la base augmentée (voir ? ?).

Remarquons que ces APIs complètes « brutes » devront s'accompagner d'une documentation publique à destination des développeurs de l'écosystème e-ReColNat précisant la sémantique et la syntaxe des ressources structurées qu'elles permettent de récupérer (voir *infra*).

Enfin, sur le plan scientifique, concevoir chaque module comme un fournisseur de données complet autorise à découvrir et inventer de nouveaux usages des données en cours de route. À ce jour, nous avons identifié des usages intéressants sur le plan communautaire ou scientifique à la quasi totalité des données produites par les Herbonautes au sein du collaboratoire.

2.3 L'API « tags »

Note : afin de spécifier nos APIs REST, nous utilisons le langage de description **RAML**, et préconisons son utilisation pour toute tâche similaire.

> Dépôt Github des APIs e-ReColNat+ <