

TP1 – WEB SERVICES – 17,5%

MISE EN SITUATION

Vous avez été approché par la compagnie Yannick-Bray, pour réaliser les services qui permettront d'offrir certaines fonctionnalités pour les développeurs. L'idée est d'augmenter le nombre d'applications cliente pouvant utiliser l'information à propos des produits et de leur inventaire.

MANDAT

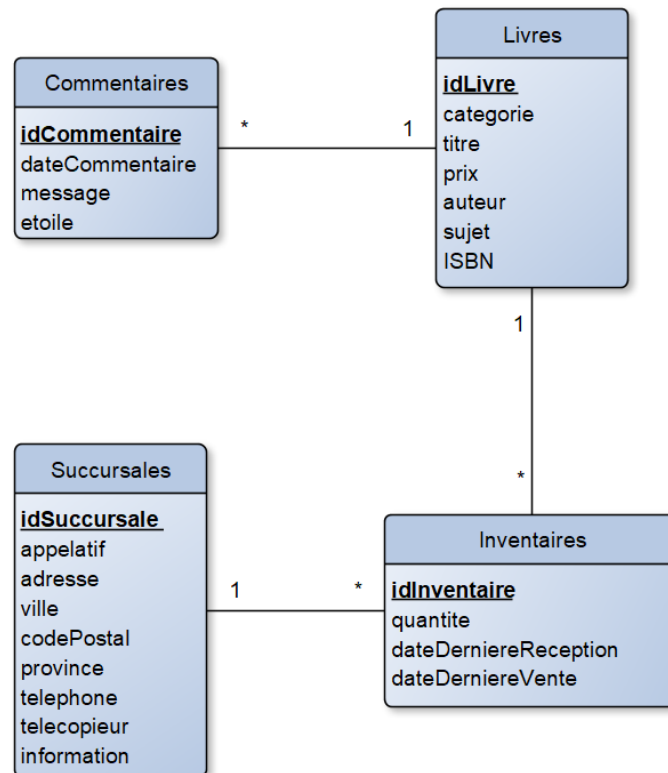
En équipe de 3, vous devez réaliser à l'aide du langage node.js et des modules nécessaires, les Web Services décrits dans la suite du document. Vous devez remettre tous les fichiers nécessaires à l'exécution de votre serveur. Vous devez remettre à la racine de votre projet un fichier **README.md**, contenant tous les prénoms, noms, matricules et la lettre d'équiper des membres de l'équipe. Ainsi que la documentation à suivre pour mettre en fonction le serveur sur un poste de travail sans **node.js**.

Votre travail devra être remis avant 23 h 59, le 16 octobre 2020 sur Léa dans une archive zip (une remise par équipe) :

- Le dossier node_modules ne doit pas être dans l'archive

BASE DE DONNÉES

Vous devez créer une base de données qui permettra de sauvegarder les informations du système. Votre base de données lors de la remise doit contenir des données initiales permettant lors de la correction de tester chacun de vos services. Voici les données et les relations entre les ressources du système. À vous de voir en équipe comme le traduire dans une base de données NoSQL (Mongo). Assurez-vous que votre serveur valide les données en entrées, des modules existent.



SERVICES WEB

Pour chacun des services web demandés, vous devez développer les routes associées dans votre application serveur. La note sera attribuée à l'équipier associé à la route.

SUCCURSALES

Ajout d'une succursale		Équipier C
URL	/succursales	
Paramètres d'URL	_body : Permet de retourner ou non la représentation de l'objet ajouté dans la réponse	
Corps de requête	La représentation JSON de la succursale à ajouter	
Type de réponse	Objet ou Aucun contenu	
Succès	<ul style="list-style-type: none"> Code http de création avec succès La représentation JSON de la succursale ajoutée HEADER location contenant l'URL de la succursale ajoutée 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

Mise à jour complète d'une succursale		Équipier C
URL	/succursales/:idSuccursale	
Paramètres d'URL	_body : Permet de retourner ou non la représentation de l'objet ajouté dans la réponse	
Corps de requête	La représentation complète JSON de la succursale à modifier	
Type de réponse	Objet ou Aucun contenu	
Succès	<ul style="list-style-type: none"> Code http de succès La représentation JSON de la succursale suite aux modifications 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

Sélection d'une succursale		Équipier A
URL	/succursales/:idSuccursale	
Paramètres d'URL	embed : Sur la collection d'inventaires fields : Permet de retourner seulement les attributs spécifiés	
Corps de requête	Aucun	
Type de réponse	Objet	
Succès	<ul style="list-style-type: none"> Code http de succès Représentation JSON d'une succursale 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

CATÉGORIES

Sélection de toutes les catégories		Équipier B
URL	/categories	
Paramètres d'URL	Aucun	
Corps de requête	Aucun	
Type de réponse	Collection sans metadata	
Succès	<ul style="list-style-type: none"> Code http de succès Représentation JSON d'une collection de toutes les catégories 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

LIVRES

Ajout d'un livre		Équipier A
URL	/livres	
Paramètres d'URL	Aucun	
Corps de requête	La représentation JSON du livre à ajouter	
Type de réponse	Objet ou Aucun contenu	
Succès	<ul style="list-style-type: none"> Code http de création avec succès La représentation JSON du livre ajouté HEADER location contenant l'URL du livre ajouté 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

Sélection de tous les livres		Équipier A
URL	/livres	
Paramètres d'URL	categorie :Retourne seulement les livres de la catégorie spécifiée	
Corps de requête	Aucun	
Type de réponse	Collection avec metadata	
Succès	<ul style="list-style-type: none"> Code http de succès Représentation JSON d'une collection de tous les livres 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

Sélection d'un livre		Équipier B
URL	/livres/:idLivre	
Paramètres d'URL	embed : Sur la collection d'inventaires fields : Permet de sélectionner seulement les attributs spécifiés	
Corps de requête	Aucun	
Type de réponse	Objet	
Succès	<ul style="list-style-type: none"> Code http de succès Représentation JSON d'un livre 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

Mise à jour partielle d'un livre		Équipier B
URL	/livres/:idLivre	
Paramètres d'URL	_body : Permits de retourner ou non la représentation de l'objet ajouté dans la réponse	
Corps de requête	La représentation JSON partielle du livre à modifier	
Type de réponse	Objet ou Aucun contenu	
Succès	<ul style="list-style-type: none"> Code http de succès Représentation JSON du livre mis à jour 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

Suppression d'un livre		Équipier C
URL	/livres/:idLivre	
Paramètres d'URL	Aucun	
Corps de requête	Aucun	
Type de réponse	Aucun	
Succès	<ul style="list-style-type: none"> Code http pour aucun contenu 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

Ajouter un commentaire à propos d'un livre		Équipier B
URL	/livres/:idLivre /commentaires	
Paramètres d'URL	_body : Permits de retourner ou non la représentation de l'objet dans la réponse	
Corps de requête	La représentation JSON du commentaire à ajouter	
Type de réponse	Objet ou Aucun	
Succès	<ul style="list-style-type: none"> Code http de création avec succès La représentation JSON du commentaire ajouté HEADER location contenant URL du commentaire ajouté. 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

Sélection des inventaires d'un livre		Équipier C
URL	/livres/:idLivre /inventaires	
Paramètres d'URL	Aucun	
Corps de requête	Aucun	
Type de réponse	Collection sans metadata	
Succès	<ul style="list-style-type: none"> Code http de succès Représentation JSON des inventaires d'un livre 	
Échec	<ul style="list-style-type: none"> Les codes d'erreur pouvant être générés Le format de réponse d'erreur étudié en classe. 	

CONSIGNES SUPPLÉMENTAIRES

SÉRIALISATION

Vous devez respecter les bonnes pratiques étudiées en classe pour la sérialisation de vos objets en JSON.

LINKING

COLLECTION SANS METADATA

Lorsqu'une collection sans *metadata* est demandée, vous devez retourner une collection JSON contenant les éléments de la collection.

COLLECTION AVEC METADATA

Lorsqu'une collection avec metadata doit être retournée, vous devez retourner une réponse semblable à celle-ci :

```
{
  "_metadata": {
    "hasNextPage": true,
    "page": 12,
    "limit": 20,
    "totalPages": 50,
    "totalDocument": 1000
  },
  "_links": {
    "prev": "http://localhost:5000/...",
    "self": "http://localhost:5000/...",
    "next": "http://localhost:5000/..."
  },
  "results": [{}, {}, {}, {}]
```

REPRÉSENTATION DES DOCUMENTS

Toutes les représentations de document doivent contenir au minimum un attribut **href** qui rend possible la bonne pratique du **linking**. Les données sensibles ne doivent pas être retournées au client. Vous devez utiliser le `_id` pour générer le href de chacune des ressources.

PARAMÈTRES D'URL

Comme nous l'avons étudié en classe, lors de certaines requêtes nous pouvons ajouter certains paramètres à l'URL de la requête pour demander au serveur de faire un traitement particulier. Voici les quatre possibilités pour ce travail pratique. Vous devez implémenter les paramètres lorsque demandé dans la description de la fonctionnalité. N'oubliez pas que les paramètres d'URL sont situés après le ? dans l'URL. Exemple :

`www.monservice.com/ressource?param1=valeur1¶m2=valeur2`

EMBED

Format	?embed={ressource associée}
Comportement	La représentation doit comporter les informations de la ressource associée

FIELDS

Format	?fields={liste des attributs} {liste des attributs} : Doit être une liste d'attributs de la ressource. Chacun des attributs est séparé par une virgule (attr1, attr2, attr3).
Comportement	La représentation de la ressource doit être retournée seulement avec les attributs demandés dans le paramètre fields.

_BODY

Format	?_body={true false}
Comportement	Si la valeur du paramètre _body est false la représentation de la ressource n'est pas retournée, dans tous les autres cas, l'objet est retourné.

LIMIT, SKIP ET PAGE

Format	?limit={nombre}&skip={départ}
Format	?page={noPage}
Comportement	La valeur du paramètre limit représente le maximum de résultat à retourner dans la réponse. La valeur du paramètre skip et la position de départ des résultats à retourner dans la réponse. La valeur du paramètre page représente la page qui doit être retournée.

CODE D'ERREUR

Lorsqu'une erreur doit être retournée, vous devez aussi retourner une réponse JSON dans le format suivant :

```
{
  "developerMessage" : "",
  "userMessage" : "",
  "errorCode" : "",
  "moreInfo" : ""
}
```

TEST D'INTÉGRATION

Chaque équipe doit remettre un fichier provenant d'une collection Postman en format JSON, permettant de tester vos services web de manière individuelle. Vous devez tester le bon fonctionnement et les erreurs de chacune des routes. Le fichier doit se nommer **tests.json** et être à la racine de votre projet.

PONDÉRATION

Ce travail compte pour **17,5%** de la note final du cours, les points seront distribués de cette manière :

Élément de correction	
Équipier A	/25
Sélection d'une succursale	7,5
Ajout d'un livre	7,5
Sélection de tous les livres	10
Équipier B	/25
Sélection de toutes les catégories	5
Sélection d'un livre	10
Mise à jour partielle d'un livre	5
Ajout d'un commentaire pour un livre	5
Équipier C	/25
Ajout d'une succursale	7,5
Mise à jour complète d'une succursale	7,5
Suppression d'un livre	2,5
Sélection des inventaires d'un livre	7,5
Architecture du projet (pour l'ensemble de l'équipe)	
Base de données et données initiales	/7.5
Sérialisation adéquate	
Division du code pertinente	
Documentation de mise en route suffisante	
Test d'intégration	
Professionalisme (pour chacun des équipiers)	
Respect des consignes	/12.5
Qualité du code	
Qualité des commentaires	
Gestion d'adéquat des erreurs	
Total individuel	/37.5
Total équipe	/7.5
Total	/45