



---

# DOCUMENTATION TP API REST

---

Auteur : Traoré Aichatou Boucara

Encadrant : Greg Galli



## Table des matières

Informations générales .....	2
Rappel de l'objectif du TP REST .....	3
Listes des actions possibles, avec leur URL REST, les verbes http.....	3
Autorisation d'accès aux services.....	5
Quelques exemples d'utilisation de l'API avec CURL .....	6
Les codes de statu http générés.....	17

## Informations générales

<b>Framework</b>	Grails 3.3
<b>Langage de programmation</b>	Groovy
<b>Type Sécurité REST implémentée</b>	Authentification : Si un utilisateur s'authentifie un token de durée 86400 secondes lui est généré automatiquement lui permettant d'accéder aux services offerts par L'API .
<b>Environnement de développement</b>	Intelli J
<b>Plugins Grails utilisés pour la sécurité</b>	<b>org.grails.plugins:springsecurity-core:3.2.0.M1</b> <b>org.grails.plugins:springsecurity-rest:2.0.0.M2</b>
<b>Base de données</b>	H2 , 5 bibliothèques contenant chacune 5 livres donc 25 livres au total initialise la base de données au lancement du projet
<b>Outil de test de l'API</b>	Postman , CURL
<b>Outil d'hébergement de code</b>	Github
<b>Trois utilisateurs de l'API</b>	Utilisateur d'identifiants : username toto , password pwd1 username aicha, password 0000 username traore, password 0001
<b>46 tests des services disponibles sur postman dans le dossier test du projet</b>	6 tests pour les services sur la collection de livres de la ressource bibliothèque, 12 sur la ressource bibliothèque, 11 sur la ressource livre , 14 sur les cas d'erreurs gérés et 3 pour l'authentification des 3 utilisateurs
<b>Format service traité</b>	XML et JSON à préciser dans le header de la requête

## Rappel de l'objectif du TP REST

Le but est de construire une API REST complète couvrant tous les besoins possibles sur un modèle de donnée simple, composé de deux entités liées par une composition.

- Entité bibliothèque : Constituée à partir des propriétés suivantes
  - Nom
  - Adresse
  - Année de construction
  - Collection de livres
- Entité Livre : constituée à partir des propriétés suivantes
  - Nom
  - ISBN
  - Date de parution
  - Auteur

## Listes des actions possibles, avec leur URL REST, les verbes http

URL	Verbes Http	Format Services Fonction et Réponses
/api/livre/livre-id	GET, PUT, DELETE	<b>GET</b> : consultation du livre sous le format XML/JSON <b>PUT</b> : Modification de la ressource livre et renvoie les informations sur le livre modifié sous le format(XML /JSON) <b>DELETE</b> : Suppression de la ressource livre Pas de réponse de contenu. Si succès renvoie un statut <b>204</b>
/api/livres	POST, GET	<b>GET</b> : consultation de l'ensemble des ressources livres sous le format XML/JSON <b>POST</b> : Ajout d'une ressource livre et renvoie les informations sur le livre crée sous le format JSON/XML

/api/bibliotheque/BU-id/livres/	GET, POST	<p><b>GET</b> : Consultations de l'ensemble des livres d'une bibliothèque sous le format JSON/XML</p> <p><b>POST</b> :ajout du livre dans la collection de livre de la bibliothèque et renvoie les informations sur le livre ajouté sous le format JSON/XML</p>
/api/bibliotheque/BU-id/livre/livre-id/	GET, PUT, DELETE	<p><b>GET</b> : Consultation du livre dans la bibliothèque sous le format JSON/XML</p> <p><b>PUT</b> : Modification du livre dans la collection de livre de la bibliothèque et renvoie les informations sur la ressource livre modifié dans la bibliothèque sous le format JSON/XML</p> <p><b>DELETE</b> : Suppression du livre dans la bibliothèque, pas de réponse de contenu. Si succès renvoie un statu <b>204</b></p>
/api/bibliotheque/BU-id/	GET, PUT, DELETE	<p><b>GET</b> : Consultation de la bibliothèque sous le format JSON/XML</p> <p><b>PUT</b> : Informations sur la ressource bibliothèque modifiée sous le format JSON/XML</p> <p><b>DELETE</b> : suppression de la bibliothèque si succès renvoie un statu <b>204</b></p>





```
{
  "id": 2,
  "livres": [
    {
      "id": 7
    },
    {
      "id": 8
    },
    {
      "id": 10
    },
    {
      "id": 6
    },
    {
      "id": 9
    }
  ],
  "nom": "Bibliotheque de la sorbone",
  "anneConstructor": 1989,
  "adresse": "Paris"
}
```

- Consultation de tous les ressources bibliothèques sous le format JSON

Sample Request

Lecture\_all\_Bibliotheque\_JSON

```
curl --request GET \
  --url http://localhost:8080/api/bibliotheques \
  --header 'accept: JSON' \
  --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcm9udG91Y2lwYywiOiJINHNJQUFBQUFBBS1ZTTU4VVF5Ui
```



```
[
  {
    "id": 1,
    "livres": [
      {
        "id": 5
      },
      {
        "id": 4
      },
      {
        "id": 27
      },
      {
        "id": 2
      },
      {
        "id": 28
      },
      {
        "id": 26
      }
    ],
    "nom": "SaintJean",
    "anneConstructor": 1990,
    "adresse": "Avenue François Mitterrand"
  },
  {
    "id": 2,
    "livres": [
      {
        "id": 10
      },
      {
        "id": 6
      },
      {
        "id": 9
      }
    ]
  }
]
```

- Création d'une bibliothèque et consultation de la bibliothèque créer sous le format XML

```
sample Request  
Création_Bibliotheque_reponse_JSON  
  
curl --request POST \  
--url 'http://localhost:8080/api/bibliotheques?nom=BU%20des%20lettres&adresse=carlone&annee=1999' \  
--header 'accept: JSON' \  
--header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcmVuZGljaWwYmwiOiJINHNJQUFBQUBBQUBBS1ZTTVU4VVVSUkiLCJ1bmEiOiJkaXNpdC9yZW50dGVudC9kYXN0IiwiaWF0IjoxNTUxMTE0MDAwfQ.'
```

```
Sample Response
```

```
{
  "id": 8,
  "livres": null,
  "nom": "BU des lettres",
  "anneConstructor": 1970,
  "adresse": "carlone"
}
```

- Modification du nom d'une bibliothèque et consultation de la ressource modifiée sous le format JSON

```
curl --request PUT \  
--url 'http://localhost:8080/api/bibliotheque/2?nom=Lettre' \  
--header 'accept: JSON' \  
--header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcmVudC5yYXZlbnR1eWwiOiJJNHhjQUFBQUBS1ZTTTU4VVFSUi'
```

- Suppression d'une bibliothèque

DELETE\_BU

```
curl --request DELETE \
  --url http://localhost:8080/api/bibliotheque/3 \
  --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcmVuY2lwYWwioiJINHNJQUFBQUFBQUFBBS1ZTTVU4VVFUSi
```

- Les requêtes sur la collection d'une ressource bibliothèque
  - Consultation d'un livre d'une ressource bibliothèque identifiée dans l'URL de la requête sous le format JSON

#### Sample Request

Lecture\_UnLivre\_BU\_JSON

```
curl --request GET \
--url http://localhost:8080/api/bibliotheque/1/livre/4/ \
--header 'accept: JSON' \
--header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcmVuY2lwYWwioiJJNHhjQUFBQUBBQUBS1ZTUVU4VWFSUjE
```

```
Sample Response
```

```
{
  "id": 4,
  "bu": {
    "id": 1
  },
  "auteur": "Honore de Balzac",
  "isbn": "1234",
  "nom": "Mamadou",
  "dateApparution": "2010-05-09T22:00:00Z"
}
```

- Modification d'un livre parmi la collection de livre d'une ressource bibliothèque et consultation du livre modifié sous le format JSON

Sample Request

```
Modification_collection

curl --request PUT \
  --url 'http://localhost:8080/api/bibliotheque/1/livre/2?nom=ami' \
  --header 'accept: JSON' \
  --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcmVudGkiOiJINHNJQUFBQUFBQUFB1ZTTU4VVF5U1I'`
```

- Consultation de la collection de livre contenu dans la bibliothèque dont l'identifiant est précisé dans l'URL de la requête sous le format JSON

#### Sample Request

Lecture\_collectionLivres\_BU\_JSON

```
curl --request GET \  
  --url http://localhost:8080/api/bibliotheque/2/livres \  
  --header 'accept: JSON' \  
  --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcm1uY2lwYWwiOiJINHNJQUFBQUFBQ1ZTTTU4VVFSUi
```

```
[  
  {  
    "id": 7,  
    "bu": {  
      "id": 2  
    },  
    "auteur": "Boccace",  
    "isbn": "1236",  
    "nom": "Decameron",  
    "dateApparution": "2017-11-04T08:54:12Z"  
  },  
  {  
    "id": 9,  
    "bu": {  
      "id": 2  
    },  
    "auteur": "Emily Bronte",  
    "isbn": "1238",  
    "nom": "Les Hauts de Hurlevement",  
    "dateApparution": "2017-11-04T08:54:12Z"  
  },  
  {  
    "id": 8,  
    "bu": {  
      "id": 2  
    },  
    "auteur": "Jorge Luis Borges",  
    "isbn": "1237",  
    "nom": "Fictions",  
    "dateApparution": "2017-11-04T08:54:12Z"  
  },  
]
```

- Ajout d'une nouvelle ressource livre dans la collection de ressource de la bibliothèque dont l'identifiant est précisé dans l'URL de la requête

**Sample Request**

```
Ajout_livre_BU_JSON

curl --request POST \
      --url 'http://localhost:8080/api/bibliotheque/1/livres?nom=vale&dateApparution=2017-10-13%' \
      --header 'accept: JSON' \
      --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcmVuY2lwYWwiOiJJNHNJQUFBQUBQUBS1ZTTVU4VVFSUI'
```

```
{
  "id": 28,
  "bu": {
    "id": 1
  },
  "auteur": "Selfi",
  "isbn": "vale",
  "nom": "vale",
  "dateApparution": "2017-10-13T06:22:20Z"
}
```

- Suppression d'une ressource livre dans la collection de livre de la ressource bibliothèque dont l'identifiant est précisé dans l'URL de la requête

**Sample Request**

```
DELETE_Livre_BU

curl --request DELETE \
      --url http://localhost:8080/api/bibliotheque/1/livre/1 \
      --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcmVudWwioiJINHNjQUFBQUBBQUBS1ZTTVU4VWFVSUoiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9'`
```

- Les requêtes sur les ressources livres
  - Consultation de la ressource livre dont l'identifiant est précisé dans l'URL de la requête sous le format JSON

- Les requêtes sur les ressources livres
  - Consultation de la ressource livre dont l'identifiant est précisé dans l'URL de la requête sous le format JSON

Sample Request

```
lecture_Livre_reponse_JSON
```

```
curl --request GET \
  --url http://localhost:8080/api/livre/2 \
  --header 'accept: JSON' \
  --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcmVuY2lwYywiOiJINHNJQUFBQUFBQS1ZTTU4VVF5Ui
```

Sample Response

```
{
  "id": 2,
  "bu": {
    "id": 1
  },
  "auteur": "Hans Christian Andersen",
  "isbn": "1232",
  "nom": "ami",
  "dateApparution": "2017-11-04T08:54:12Z"
}
```

- Consultation de tous les ressources livres existant sous le format JSON

Sample Request

```
Lecture_All_Livre_JSON
```

```
curl --request GET \
  --url http://localhost:8080/api/livres \
  --header 'accept: JSON' \
  --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcmVuY2lwYywiOiJINHNJQUFBQUFBQS1ZTdjA4VVF5Ui
```

```
[
  {
    "id": 1,
    "bu": {
      "id": 1
    },
    "auteur": "Chinua Achebe",
    "isbn": "1231",
    "nom": "Le monde seffondre",
    "dateApparution": "2017-11-04T08:54:12Z"
  },
  {
    "id": 2,
    "bu": {
      "id": 1
    },
    "auteur": "Hans Christian Andersen",
    "isbn": "1232",
    "nom": "Contes",
    "dateApparution": "2017-11-04T08:54:12Z"
  },
  {
    "id": 3,
    "bu": {
      "id": 1
    },
    "auteur": "Jane Austen",
    "isbn": "1233",
    "nom": "Orgueil et prejuges",
    "dateApparution": "2017-11-04T08:54:12Z"
  },
  {
    "id": 4,
    "bu": {
      "id": 1
    },
    "auteur": "Honore de Balzac",

```

- Création d'une nouvelle ressource livre et consultation du livre créé sous le format XML

**Sample Request**

```
Creation_Livre_reponse_xml

curl --request POST \
      --url 'http://localhost:8080/api/livres?bu.id=1&nom=Abid&dateApparution=2017-10-13%2008:22' \
      --header 'accept: XML' \
      --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcmVudWY2YmYwIiwiaWF0IjoiMjAxNjA5MDQyOTUyIn0='
```

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?><livre id="29"><bu id="1" /><auteur>Kingston</auteur></livre>
```



- Sample Request

PUT\_Livre\_reponse\_JSON

```
curl --request PUT \  
  --url 'http://localhost:8080/api/livre/4?bu.id=1&dateApparution=2017-11-04%2009:19:57.15' \  
  --header 'accept: JSON' \  
  --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcm9udWY2IiwiaWF0IjINHNjQUFBQUBQUBS1ZTdjaA4VVF5Ui
```

Sample Response

```
{  
  "id": 4,  
  "bu": {  
    "id": 1  
  },  
  "auteur": "Honore de Balzac",  
  "isbn": "1234",  
  "nom": "Mamadou",  
  "dateApparution": "2010-05-09T22:00:00Z"  
}
```

- Sample Request

```
DELETE_Livre

curl --request DELETE \
  --url 'http://localhost:8080/api/livre/3?bu.id=1' \
  --header 'mytoken: eyJhbGciOiJIUzI1NiJ9.eyJwcm9udG91Y2lwYyI6IjINHNJQUFBQUBBQUFBBS1ZTdjaA4VVF5UiJ9.eyJwcm9udG91Y2lwYyI6IjINHNJQUFBQUBBQUFBBS1ZTdjaA4VVF5UiJ9'
```

## Les codes de statu http générés

Statu	Exemple d'utilisation
200	La requête PUT sur une ressource bibliothèque ou livre réussit avec succès
201	La requête POST sur une ressource bibliothèque ou livre réussit avec succès
204	La requête DELETE sur une ressource bibliothèque ou livre réussit avec succès
400	Une mauvaise URL utilisée pour faire une requête POST, PUT, DELETE sur une ressource bibliothèque ou livre
401	Si un utilisateur précise bien dans le header de la requête le paramètre MyToken ayant pour valeur le token de l'utilisateur déjà expiré alors le serveur génère ce code de statu
404	La requête GET sur une ressource bibliothèque ou livre inexistant
405	Un verbe http utilisé sur un URL non approprié
406	Faire une requête sans précisé dans le header de la requête le paramètre MyToken portant le token du client