# Spécifications API WS REST

Méthodes de l’API

**Méthode book() du contrôler api**

* Obtention d’un livre
* GET via
* <http://localhost:8081/tpwsrest/api/book/1>
* Entrée : une requête de type GET, un header « Accept: application/json » et l’ID du livre que l’on veut obtenir
* Sortie : le livre en JSON ou XML en fonction du header (JSON par défaut) avec ID de sa libraire parente
* Code : 200 si obtention du livre ou 404 sinon
* Gestion des erreurs : livre avec cet ID introuvable (404)
* Création d’un livre
* POST via
* <http://localhost:8081/tpwsrest/api/book?name=newBook&releaseDate=2017-04-01T22:00:00Z&isbn=666&author=JCVD&library.id=2>”
* Entrée : une requête de type POST, un header « Accept: application/json » et les paramètres du livre à créer avec l’ID de la librairie à rattacher
* Sortie : un message indiquant l’échec ou le succès de l’insertion du livre
* Code : 201 si livre créé, ou 400 sinon
* Gestion des erreurs : si livre non créé ou librairie à rattacher introuvable (400)
* Mise à jour d’un livre
* PUT via <http://localhost:8081/tpwsrest/api/book/5?name=updatedBook&releaseDate=2017-04-01T22:00:00Z&isbn=666&author=JCVD&library.id=2>”
* Entrée : une requête de type PUT, un header « Accept: application/json », l’ID et les paramètres du livre à modifier
* Sortie : un message indiquant l’échec ou le succès de la mise à jour du livre
* Code : 200 si livre mis à jour, ou 400 sinon
* Gestion des erreurs : si livre non mis à jour ou librairie à rattacher introuvable (400)
* Suppression d’un livre en cascade
* DELETE via
* <http://localhost:8081/tpwsrest/api/book/1>
* Entrée : une requête de type DELETE, un header « Accept: application/json » et l’ID du livre que l’on veut supprimer
* Sortie : un message indiquant l’échec ou le succès de la suppression du livre
* Code : 200 si suppression du livre ou 404 sinon
* Gestion des erreurs : livre avec cet ID introuvable (404)

**Méthode library() du contrôler api**

* Obtention d’une librairie
* GET via
* <http://localhost:8081/tpwsrest/api/library/1>
* Entrée : une requête de type GET, un header « Accept: application/json » et l’ID de la bibliothèque que l’on veut obtenir
* Sortie : la libraire en JSON ou XML en fonction du header (JSON par défaut) et les ID de ses livres
* Code : 200 si obtention de la bibliothèque ou 404 sinon
* Gestion des erreurs : librairie avec cet ID introuvable (404) ou requête non compréhensible (400)
* Création d’une librairie
* POST via
* <http://localhost:8081/tpwsrest/api/library?name=newBiblio&address=LibanStreet&yearCreated=1999>"
* Entrée : une requête de type POST, un header « Accept: application/json » et les paramètres de l’application à créer
* Sortie : un message indiquant l’échec ou le succès de l’insertion de la librairie
* Code : 201 si librairie créée, ou 400 sinon
* Gestion des erreurs : librairie non créée (400) ou requête non compréhensible (400)
* Mise à jour d’une librairie
* PUT via
* <http://localhost:8081/tpwsrest/api/library/3?name=libUpdated&address=Alesiastreet&yearCreated=1945>"
* Entrée : une requête de type PUT, un header « Accept: application/json », l’ID et les paramètres du livre à modifier
* Sortie : un message indiquant l’échec ou le succès de la mise à jour de la librarie
* Code : 200 si libraire mise à jour, ou 400 sinon, ou 404
* Gestion des erreurs : si librairie à modifier introuvable (404) ou requête non compréhensible (400)
* Suppression d’une librairie en cascade
* DELETE via
* [http://localhost:8081/tpwsrest/api/library/2](http://localhost:8081/tpwsrest/api/book/1)
* Entrée : une requête de type DELETE, un header « Accept: application/json » et l’ID de la librairie que l’on veut supprimer
* Sortie : un message indiquant l’échec ou le succès de la suppression du de la librairie
* Code : 200 si obtention de la librairie ou 400 sinon, ou 404
* Gestion des erreurs : librairie avec cet ID introuvable (404) ou requête non compréhensible (400)

**Méthode books() du contrôler api**

* Obtention de tous les livres
* GET via
* <http://localhost:8081/tpwsrest/api/books>
* Entrée : une requête de type GET, un header « Accept: application/json »
* Sortie : La liste de tous les livres en JSON ou XML en fonction du header (JSON par défaut)
* Code : 200 si obtention des livres ou 400 sinon
* Gestion des erreurs : requête non compréhensible (400)
* Suppression de tous les livres en cacades
* DELETE via
* <http://localhost:8081/tpwsrest/api/books>
* Entrée : une requête de type DELETE, un header « Accept: application/json »
* Sortie : un message indiquant l’échec ou le succès de la suppression des livres
* Code : 200 si suppression des livres
* Gestion des erreurs : requête non compréhensible (400)

**Méthode librairies() du contrôler api**

* Obtention de toutes les librairies
* GET via
* [http://localhost:8081/tpwsrest/api/librairies](http://localhost:8081/tpwsrest/api/books)
* Entrée : une requête de type GET, un header « Accept: application/json »
* Sortie : La liste de toutes les librairies en JSON ou XML en fonction du header (JSON par défaut)
* Code : 200 si obtention des librairies ou 400 sinon
* Gestion des erreurs : requête non compréhensible (400)
* Suppression de toutes les librairies en cacades
* DELETE via
* <http://localhost:8081/tpwsrest/api/librairies>
* Entrée : une requête de type DELETE, un header « Accept: application/json »
* Sortie : un message indiquant l’échec ou le succès de la suppression des librairies en cascade
* Code : 200 si suppression des librairies en cascade
* Gestion des erreurs : requête non compréhensible (400)

**Méthode booksInLibrary() du contrôler api**

* Obtention de tous les livres d’une bibliothèque
* GET via
* <http://localhost:8081/tpwsrest/api/library/1/books>
* Entrée : une requête de type GET, un header « Accept: application/json » et l’id de la bibliothèque dont on veut les livres
* Sortie : La liste de tous les livres en JSON ou XML en fonction du header (JSON par défaut) de la librairie demandée
* Code : 200 si obtention des livres ou 404 sinon, 400
* Gestion des erreurs : librairie avec cet ID introuvable (404), requête non compréhensible (400)

**Méthode bookInLibrary() du contrôler api**

* Obtention d’un livre en particulier dans une bibliothèque
* GET via
* <http://localhost:8081/tpwsrest/api/library/1/book/2>
* Entrée : une requête de type GET, un header « Accept: application/json », l’ID de la bibliothèque et la position du livre à l’interieur
* Sortie : Le livre en JSON ou XML en fonction du header (JSON par défaut) dans la librairie demandée
* Code : 200 si obtention du livre ou 400 sinon, 404.
* Gestion des erreurs : librairie avec cet ID introuvable (404), livre avec cet id introuvable (404)
* Création d’un livre dans une bibliothèque en particulier
* POST via
* <http://localhost:8081/tpwsrest/api/library/2/book/10?name=newBook&releaseDate=2017-04-01T22:00:00Z&isbn=666&author=JCVD&library.id=2>
* Entrée : une requête de type POST, un header « Accept: application/json » ,l’ID de la libraire à rattacher et les paramètres du livre à créer
* Sortie : un message indiquant l’échec ou le succès de l’insertion du livre dans la bibliothèque
* Code : 201 si livre créé, ou 400 sinon, ou 404
* Gestion des erreurs : si livre non créé (400) ou librairie à rattacher introuvable (404)