

Programmation Web – Avancé

JavaScript & Node.js

Partie 21 : RESTful API

Version 2020



Attribution –
Partage dans les
Mêmes Conditions
4.0 International
(CC BY-SA 4.0)

*Presentation template
by [SlidesCarnival](#)*



Introduction aux RESTful API

Comment faire coopérer des applications au travers d'une interface
uniforme ?



RESTful API's

- **RE**presentational **S**tate **T**ransfer :
style architectural pour construire des applications web extensibles
- Définition d'une interface uniforme permettant à des applications de coopérer



RESTful API's

- 1 collection de données = des ressources
- 1 élément d'une collection = 1 ressource
- 1 adresse unique pour chaque ressource = 1 URL
- Opérations possibles (% CRUD) sur une ressource = méthode HTTP



RESTful API's : une convention

- Adresse unique d'une ressource par combinaison de :
 - URL racine

```
www.ilovejs.com/api
```

- Identifiant unique du type de la ressource

```
users
```

```
www.ilovejs.com/api/users/10
```



RESTful API's : une convention

- Opérations possibles (% CRUD) sur une ressource données = méthode HTTP
 - GET = READ
 - POST = CREATE
 - DELETE = DELETE :)
 - PUT = UPDATE



RESTful API's : un exemple de CRUD API

URL	Méthode	Fonction
users	GET	READ : Lire toutes les ressources de la collection
users/10	GET	READ : Lire la ressource identifiée
users	POST	CREATE : Créer une ressource basée sur les données de la requête
users/10	DELETE	DELETE : Effacer la ressource identifiée
users/10	PUT	UPDATE : Remplacer l'entiéreté de la ressource par les données de la requête
users/10	PATCH	UPDATE : Remplacer une partie de la ressource par les données de la requête



Gérer les opérations sur les ressources via Express

- Utilisation des routes : **app.get()**, **app.post()**, **app.delete()**....
- Utilisation des paramètres au sein d'une route : **request.params**

```
router.get("/users/:userId/books/:bookId", function (req, res) {  
  res.send(`Parameters in path: userID: ${req.params.userId}  
    bookId:${req.params.bookId}`);  
});
```




Consommer les ressources via le frontend

- Exemple de requête côté client :
 - Clic sur un lien : **GET**
<http://www.ilovejs.com/api/users/10>
 - Envoi des données d'un formulaire : **POST**
<http://www.ilovejs.com/api/users/>
- AJAX