

# Programmation Web – Avancé

JavaScript & Node.js

Introduction aux RESTful APIs



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

*Presentation template  
by [SlidesCarnival](#)*



# Introduction aux RESTful APIs

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



## RESTful API's

---

- **RE**presentational **State** Transfer : style architectural pour construire des applications web extensibles : **client** – **serveur** séparés
- Définition d'une **interface uniforme** permettant à des applications de coopérer
- **Stateless** : pas d'enregistrement du contexte d'une session sur le serveur pour comprendre une requête d'un client



## RESTful API's

---

- Mise à disposition d'**opérations** sur des **ressources** via :
  - des **URI**
  - des **méthodes HTTPS**
  - des **représentations** des ressources compréhensibles tant par les **clients** que les **serveurs**



## RESTful API's

---

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



## RESTful API's

---

- Manipulation de ressources d'une API via des représentations des ressources à indiquer par un client
- Media type = format de donnée représentant une ressource : **JSON**, **XML**, **HTML**, **TXT**, **JPEG**, **PNG**...
- Quasi toujours du **JSON** dans ce cours



## 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

URI	HTTP	Fonction
<b>films</b>	<b>GET</b>	READ ALL : Lire toutes les ressources de la collection
<b>films?minimumDuration=value</b>	<b>GET</b>	READ ALL FILTERED : Lire toutes les ressources de la collection selon le filtre donné
<b>films/{id}</b>	<b>GET</b>	READ ONE : Lire la ressource identifiée
<b>films</b>	<b>POST</b>	CREATE ONE : Créer une ressource basée sur les données de la requête
<b>films/{id}</b>	<b>DELETE</b>	DELETE ONE : Effacer la ressource identifiée
<b>films/{id}</b>	<b>PUT</b>	UPDATE ONE : Remplacer l'entièreté de la ressource par les données de la requête