Synthèse : web2

# Chapitre 1 : RESTful API : définition & convention

## Qu’est-ce qu’une application rest :

* REST : **REpresentational State Transfer**
  + Un style d’architecture pour construire des APIs web où **le client et le serveur sont séparés**
* Les requêtes sont stateless
  + Chaque requête du client contient toutes les informations nécessaires, le serveur ne garde pas de « contexte » session entre les requêtes

## Ressources, URl & méthodes http

* On manipule des ressources (ex : films, drinks, users, …)
  + Chaque ressource est accessible via une Url unique
* Les méthodes http standard sont utilisées pour déterminer l’opération (CRUD) :
  + GET pour lire (« READ »)
  + POST pour créer (« Create »)
  + PUT/PATCH pour mettre à jour (« Update »)
  + DELETE pour supprimer (« DELETE »)

## Représentation des ressources :

* Les ressources sont souvent représentées en JSON dans cet enseignement
* Le client envoie ses données dans le body de la requête (typiquement JSON) pour les requêtes POST, PUT, PATCH

## Express & TypesScript : structure d’une API

* Middleware dans Express :
  + **Ce sont des fonctions qui interceptent les requêtes avant qu’elles atteignent les** routes ou après, pour divers traitements : journalisation, parsing JSON, vérification, etc.
* Différents types :
  + Application-level middleware : applique à toutes les requêtes
  + Router-level middleware : appliqués à un routeur particulier
  + Error-handling middleware : middleware spécialisé pour capturer et répondre aux erreurs
* Définition des routes :
  + On définit des routes via app.METHOD(path, handler) ou router.METHOD(path, handler)
  + Routes regroupées selon les ressources avec l’usage de Router() pour organiser le code