

# JAX-RS

**Module SOA**  
**A.U 2019-2020**



# Objectifs



- Comprendre le style d'architecture REST.
- Comprendre les différences entre les services web étendus (SOAP) et les services web REST.



# Plan



- Présentation de REST
- Motivation pour REST
- Principes de REST
- Développement de services web REST java



# Présentation de REST 1/2



- REST est l'acronyme de **REpresentational State Transfert**
- Principe défini dans la thèse de Roy FIELDING en 2000
  - Il est l'un des principaux auteurs de la spécification HTTP
  - Il est le développeur du serveur Web Apache
- REST est un **style d'architecture** inspiré de l'architecture du **Web** pour construire des services web



Un style d'architecture est un ensemble de contraintes qui permettent, lorsqu'elles sont appliquées aux composants d'une architecture, d'optimiser certains critères propres au cahier des charges du système à concevoir.



# Présentation de REST 2/2

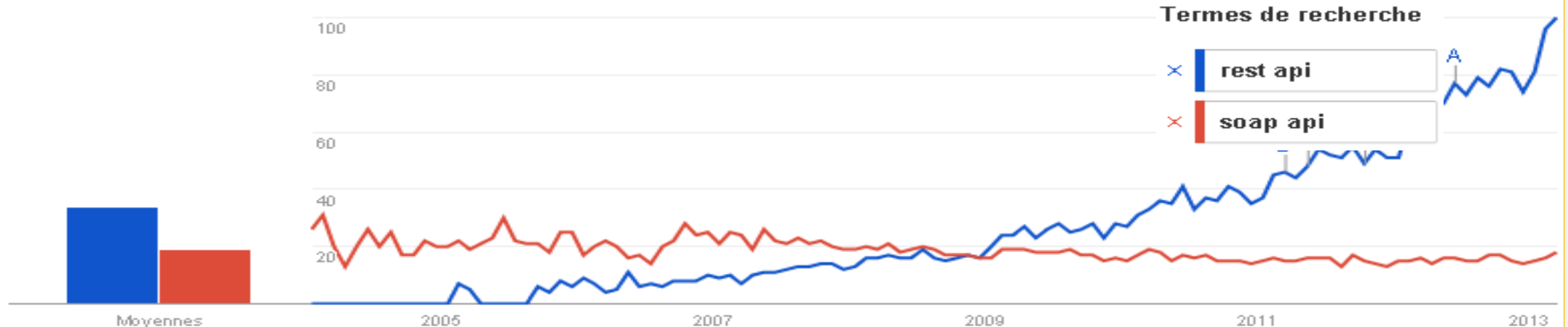


- REST n'est pas:
  - un format
  - un protocole
  - un standard
- Bien que REST ne soit pas un standard, il utilise des standards:
  - HTTP
  - URL
  - XML/HTML

# Motivation pour REST 1/2

- REST est une alternative à SOAP
- En 2006, Google a abandonné son API SOAP au profit d'une API simplifiée REST

## Évolution de l'intérêt pour cette recherche



Source: <http://www.google.com/trends/explore?hl=fr#q=rest%20api%2Csoap%20api&cmpt=q>



# Motivation pour REST 2/2



- REST est léger et simple :
  - Les messages sont courts, faciles à décoder par le navigateur et par le serveur d'application.
- REST est auto-descriptif :

vous pouvez naviguer à travers ses ressources comme vous le feriez avec une page Web. Il y a une URL intuitive unique pour chaque ressource. On peut facilement en déduire la structure des ressources sans avoir besoin de beaucoup de documentation.
- REST est stateless :
  - Consommation de mémoire inférieure
- REST peut être géré en cache
  - mise en cache possible donc meilleure montée en charge

# Principes de REST 1/7

## URI

http://weather.com/tunis

## Représentation

```
Metadata:  
Content-type:  
application/xhtml+xml  
  
Data:  
<!DOCTYPE html PUBLIC "...  
    "http://www.w3.org/...  
<html xmlns="http://www...  
<head>  
<title>5 Day Forecaste for  
Oaxaca</title>  
...  
</html>
```



## Ressource

La météo de Tunis







# Principes de REST 2/7



- Une ressource
- Un identifiant de ressource
- Une représentation de la ressource
- Interagir avec les ressources
  - Requêtes HTTP : GET, POST, PUT et DELETE





# Principes de REST 3/7



## ■ Ressources (Identifiant)

- Identifié par une URI

Exemple : `http://localhost:8080/libraryrestwebservice/books`

## ■ Méthodes (Verbes)

- pour manipuler la ressource
- Méthodes HTTP : GET, POST, PUT and DELETE

## ■ Représentation

- donne une vue sur l'état de la ressource
- informations transférées entre le client et le serveur

Exemples : XML, Text, JSON, ...



## Méthodes

# Principes de REST 4/7



- Une ressource quelconque peut subir quatre opérations de base désignées par **CRUD**
  - **Create** (Créer)
  - **Retrieve** (Lire)
  - **Update** (mettre à jour)
  - **Delete** (Supprimer)
- REST s'appuie sur le protocole HTTP pour exprimer les opérations via les méthodes HTTP
  - Create ↔ **POST**
  - Retrieve ↔ **GET**
  - Update ↔ **PUT**
  - Delete ↔ **DELETE**

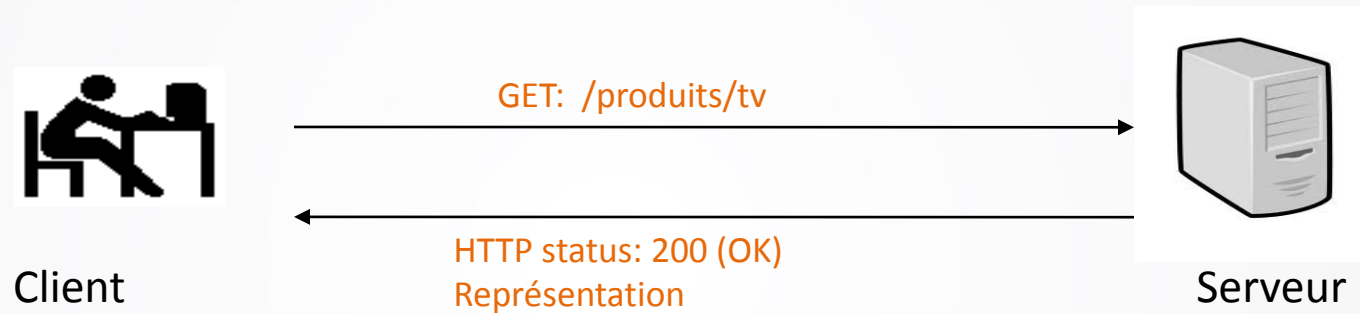


## Méthodes

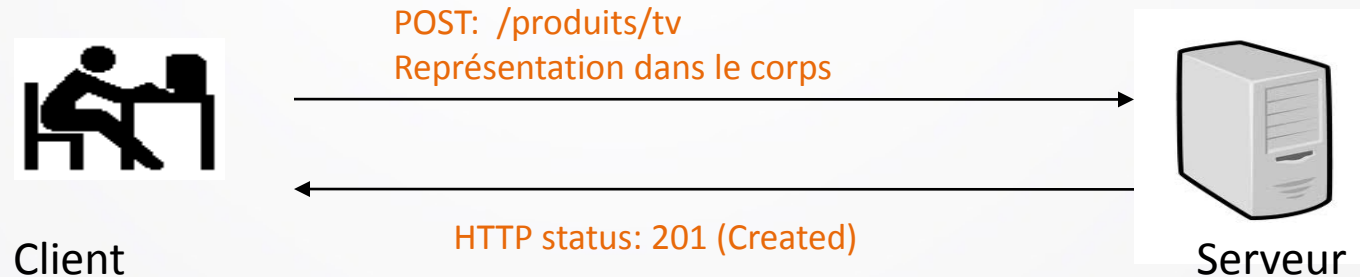
# Principes de REST 5/7



- Méthode **GET** fournit la représentation de la ressource



- Méthode **POST** crée une ressource



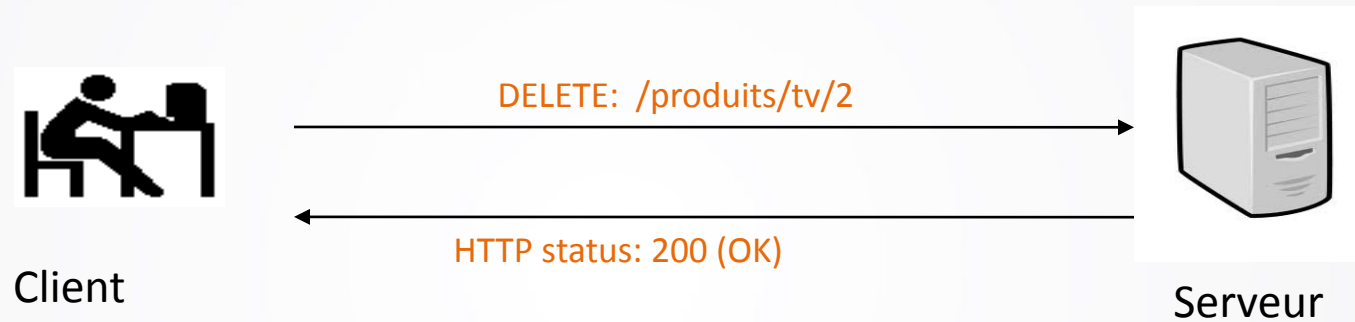


## Méthodes

# Principes de REST 6/7



- Méthode **DELETE** supprime une ressource



- Méthode **PUT** met à jour une ressource





## Représentation

# Principes de REST 7/7



Fournir les données suivant une représentation pour:

- le client (GET): format de sortie
- le serveur (PUT et POST): format d'entrée

La représentation d'une ressource peut prendre différents formats:

- XML
- JSON
- Text
- ...

Le format d'entrée (PUT et POST) et le format de sortie (GET) d'un service Web d'une ressource peuvent être différents



# WADL 1/2



- **Web Application Description Language**
- est un langage de description XML de services de type REST
- est une spécification W3C initiée par SUN
- l'objectif est de pouvoir générer automatiquement les APIs clientes d'accès aux services REST

## Remarques

- Peu d'outils exploite la description WADL
- Apparu bien plus tard

# WADL 2/2

## Exemple

```
<application>
<doc jersey:generatedBy="Jersey: 1.4 09/11/2010 10:30 PM"/>
<resources base="http://localhost:8088/librarycontentrestwebservice/">
  <resource path="/contentbooks">
    <resource path="uribuilder2">
      <method name="POST" id="createURIBooks">
        <request>
          <representation mediaType="application/xml"/>
        </request>
        <response>
          <representation mediaType="*/*/"/>
        </response>
      </method>
    </resource>
    <resource path="uribuilder1">
      <method name="POST" id="createBooksFromURI">
        <request>
          <representation mediaType="application/xml"/>
        </request>
        <response>
          <representation mediaType="*/*/"/>
        </response>
      </method>
    </resource>
    ...
  </resource>
</resources>
</application>
```





# Services Web REST avec Java



- JAX-RS: **J**ava **A**PI for **R**ESTful Web **S**ervices
- Spécification décrivant la mise en œuvre et la consommation des services web REST
- JAX-RS est basé sur les annotations

@Path	définit le chemin de la ressource. Cette annotation se place sur la classe et/ou sur la méthode implémentant le service.
@GET, @PUT, @POST, @DELETE	définit l'action implémentée par le service
@Produces	spécifie le type de la réponse du service
@Consumes	spécifie le type accepté en entrée du service



# Services Web REST avec Java



- Différentes implémentations de JAX-RS sont disponibles:
  - JERSEY (Oracle)
  - CXF (Apache)
  - RESTEasy (JBoss)
  - RESTlet
- Seule l'approche bottom-up est possible
  - Annoter une classe POJO
  - Compiler et déployer



# En résumé



- REST est un style d'architecture
- REST est une alternative aux services web étendus (SOAP)
- REST se base sur le protocole HTTP
- JAX-RS est l'API java permettant de développer et consommer des services web REST