



Université Abderrahmane Mira de Béjaïa  
Faculté des Sciences Exactes  
Département d'Informatique

# Le REST

## (Cours 5)

Dr H. EL BOUHISSI

Septembre 2023

# Objectifs du cours

---

- 1 Découvrir le les Service Web REST
- 2 Comprendre les briques de base des Services REST

## REST - Définition

---

REST acronyme de REpresentational State Transfert : Concept introduit en 2000 dans la thèse de Roy FIELDING (un des créateurs du protocole HTTP).

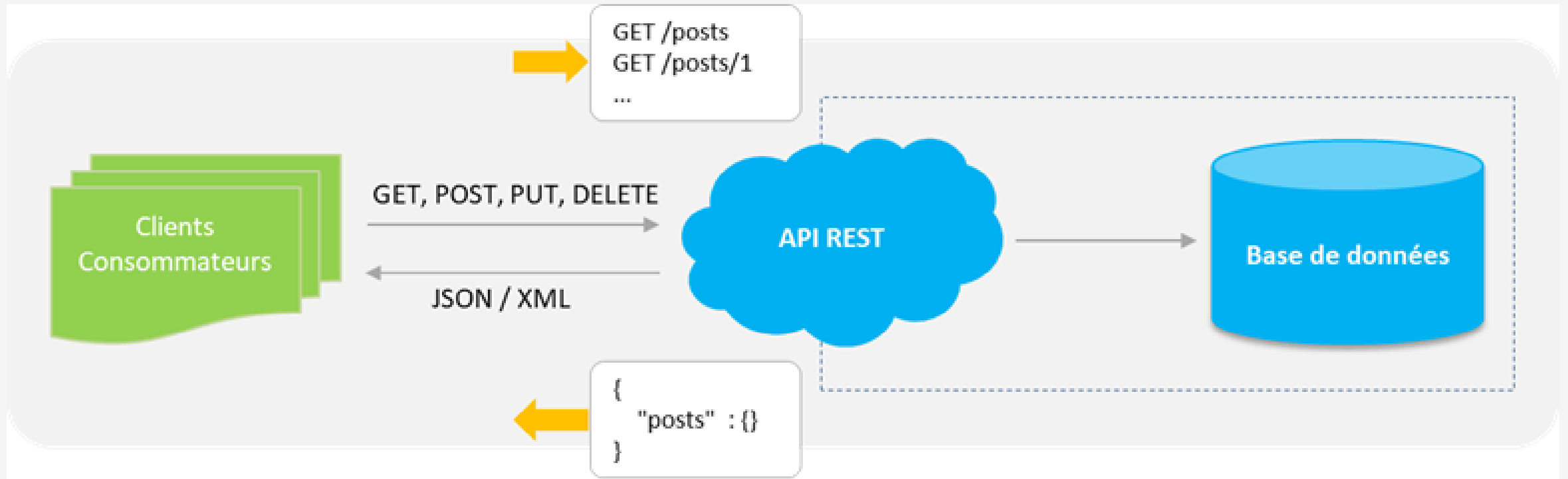
REST est une manière de construire une application pour les systèmes distribués comme le World Wide Web.

Ce n'est pas:

- Un format
- Un protocole
- Un standard

**REST n'est pas un standard: Pas de recommandation du W3C**

# Les 5 règles à suivre pour implémenter REST



Style d'architecture inspiré de l'architecture WEB et repose sur le HTTP.

Permet l'envoi de messages sans enveloppe SOAP et dans un encodage libre (XML, JSON, binaire, simple texte).

# Les 5 règles à suivre pour implémenter REST

---

## **Règle n°1 : l'URI comme identifiant des ressources**

REST se base sur les URI (Uniform Resource Identifier) afin d'identifier une ressource. Il est nécessaire de prendre en compte la hiérarchie des ressources et la sémantique des URL pour les éditer. Par exemple : une ressource : « liste de livres »

**Non OK : <http://mywebsite.com/book> OK : <http://mywebsite.com/books>**

## **Règle n°2 : les verbes HTTP comme identifiant des opérations**

La seconde règle d'une architecture REST est d'utiliser les verbes HTTP existants plutôt que d'inclure l'opération dans l'URI de la ressource. Ainsi, généralement pour une ressource, il y a 4 opérations possibles (CRUD) :

**Créer (create) → POST**

**Afficher (read) → GET**

**Mettre à jour (update) → PUT**

**Supprimer (delete) → DELETE**

# Les 5 règles à suivre pour implémenter REST

---

## **Règle n°3 : les réponses HTTP comme représentation des ressources**

Il est important d'avoir à l'esprit que la réponse envoyée n'est pas une ressource, c'est la représentation d'une ressource. Ainsi, une ressource peut avoir plusieurs représentations dans des formats divers : HTML, XML, CSV, JSON, etc.

C'est au client de définir quel format de réponse il souhaite recevoir via l'entête Accept. Il est possible de définir plusieurs formats.

## **Règle n°4 : les liens comme relation entre ressources**

Les liens d'une ressource vers une autre ont tous une chose en commun : ils indiquent la présence d'une relation.

## **Règle n°5 : un paramètre comme jeton d'authentification**

Comment authentifier une requête ? La réponse est très simple et est massivement utilisée par des APIs renommées (Google, Yahoo, etc.) : **le jeton d'authentification**.

## Services REST : Fournisseurs

---

**flickr**<sup>TM</sup>

**amazon.com**<sup>®</sup>  
and you're done.<sup>™</sup>

**facebook.**

**Google**<sup>TM</sup>

**YAHOO!**<sup>®</sup>  
<https://developer.yahoo.com/>

**twitter**

 **Zillow.com**<sup>TM</sup>  
Your Edge in Real Estate

# Description de services Web REST

---

Langages : WSDL 2.0, WADL, RSDL (Restful Service Description Language), SERIN (Semantic RESTful Interfaces).

- **WSDL2.0** : Évolution de Web Service Description Language recommandé en 2007 par le W3C. Il permet de spécifier un binding HTTP au lieu de SOAP
- **WADL** (Web Application Description Language) : Un langage de description XML de services de type REST.

Permet une description de services par éléments de type: ressource, méthode, paramètre, requête, réponse.

**Pas assez de Framework qui supportent la description WADL**



## Description de services Web REST : WADL

---

ELEMENTS WADL	DESCRIPTION
<application>	La racine d'une description WADL
<ressources base=.....>	<ul style="list-style-type: none"><li>• Un conteneur pour les ressources que l'application fournit</li><li>• L'attribut base définit l'URI pour les ressource</li></ul>
<resource id="" path="">	<ul style="list-style-type: none"><li>• Décrit la ressource que l'application fournit</li><li>• L'attribut id identifie l'élément ressource</li><li>• L'attribut path fournit une URI relative pour l'identifiant de la ressource.</li></ul>
<method name=" " id="">	<ul style="list-style-type: none"><li>• Fils d'élément ressource ou application</li><li>• L'attribut name définit les méthodes HTTP</li></ul>
<request> <response>	<ul style="list-style-type: none"><li>• Request: décrit une requête à la méthode HTTP sur une ressource</li><li>• Response: décrit la sortie en réalisant la méthode HTTP sur le ressource</li></ul>

# Description de services Web REST : WADL

## EXEMPLE de description WADL pour l'application Yahoo News Search

```
<?xml version="1.0"?>
<application xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wadl.dev.java.net/2009/02 wadl.xsd"
  xmlns:tns="urn:yahoo:yn" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:yn="urn:yahoo:yn"
  xmlns:ya="urn:yahoo:api" xmlns="http://wadl.dev.java.net/2009/02">
  <grammars><include href="NewsSearchResponse.xsd"/> <include href="Error.xsd"/>
</grammars>
  <resources base="http://api.search.yahoo.com/NewsSearchService/V1/">
    <resource path="newsSearch">
      <method name="GET" id="search">
        <request>
          <param name="appid" type="xsd:string" style="query" required="true"/>
          <param name="query" type="xsd:string" style="query" required="true"/>
          <param name="type" style="query" default="all">
            <option value="all"/> <option value="any"/> <option value="phrase"/>
          </param>
          <param name="results" style="query" type="xsd:int" default="10"/>
          <param name="start" style="query" type="xsd:int" default="1"/>
          <param name="sort" style="query" default="rank">
            <option value="rank"/> <option value="date"/>
          </param>
          <param name="language" style="query" type="xsd:string"/>
        </request>
        <response status="200">
          <representation mediaType="application/xml" element="yn:ResultSet"/>
        </response>
        <response status="400">
          <representation mediaType="application/xml" element="ya:Error"/>
        </response>
      </method>
    </resource>
  </resources>
</application>
```

Description des espaces de noms

Description de Grammaire XML utilisée par le serv

Description des ressources Et méthodes HTTP utilisées

Plus facile à comprendre, à interpréter et à écrire qu'un WSDL

# Exemples de service web REST

---

**TP 5 et TP 6**

# REST vs SOAP (Simple Object Access Protocol)

---

SOAP	REST
SOAP est un protocole.	REST est un style d'architecture.
SOAP signifie Simple Object Access Protocol.	REST signifie REPresentational State Transfer.
SOAP ne peut pas utiliser REST car c'est un protocole.	REST peut utiliser les services Web SOAP car il s'agit d'un concept et peut utiliser n'importe quel protocole comme HTTP, SOAP.
SOAP utilise des interfaces de services pour exposer la logique métier.	REST utilise l'URI pour exposer la logique métier.
JAX-WS est l'API java pour les services Web SOAP.	JAX-RS est l'API java pour les services Web RESTful.
SOAP définit les normes à suivre strictement.	REST ne définit pas trop de normes comme SOAP.
SOAP nécessite plus de bande passante et de ressources que REST.	REST nécessite moins de bande passante et de ressources que SOAP.
SOAP définit sa propre sécurité.	Les services Web RESTful héritent des mesures de sécurité du transport sous-jacent.
SOAP autorise uniquement le format de données XML.	REST autorise différents formats de données tels que le texte brut, HTML, XML, JSON, etc.
SOAP est moins préféré que REST.	REST plus préféré que SOAP.