Web services (REST)

Encadré par : **Pr. EL ABDELLAOUI SAID**

Réalisé par :

BOUAMIR ASSIA
ELAKIL HAKIMA
ANEFLOUS AMAL

Outline

Introduction

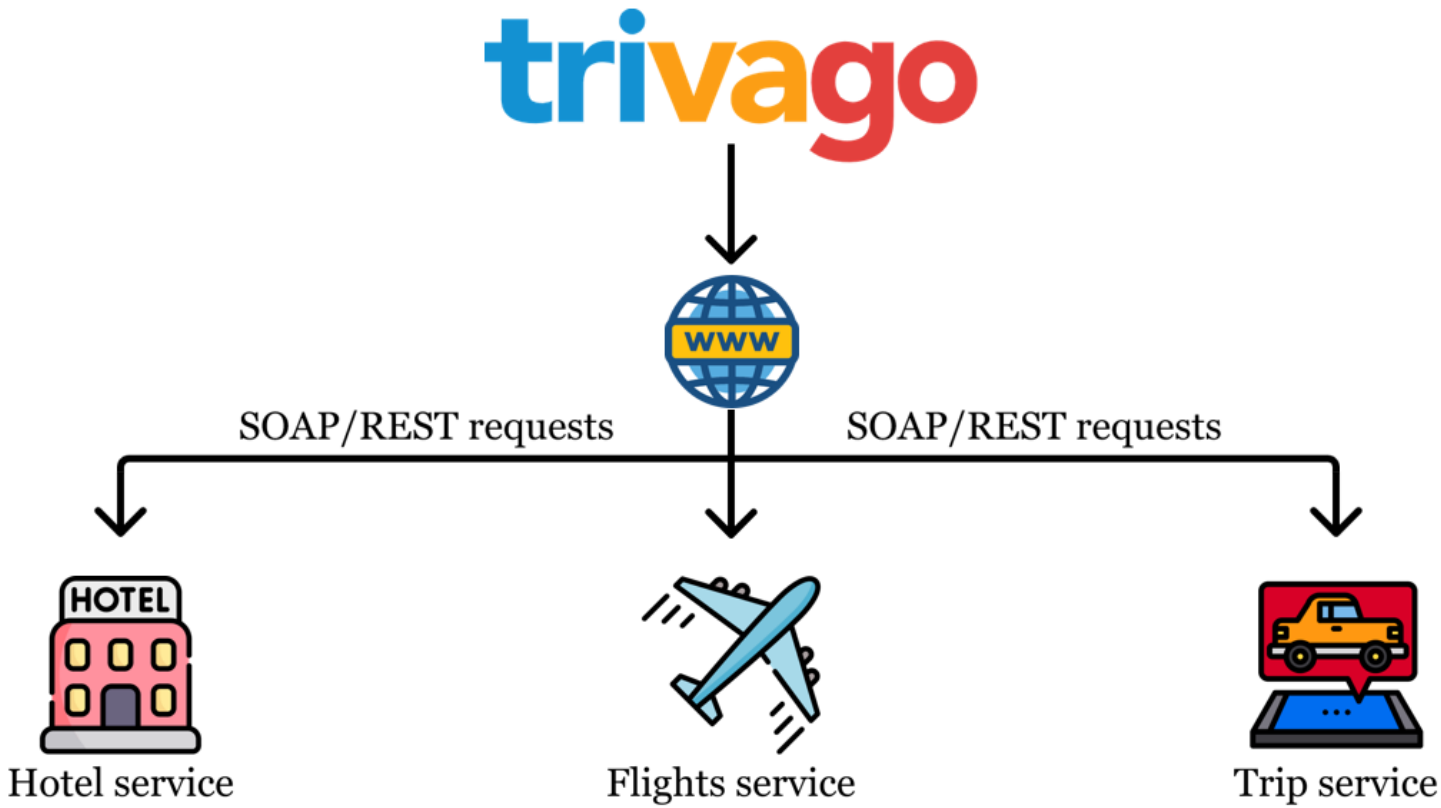
problematique

Définition

Web services

Les services web sont des systèmes de communication qui permettent à des applications logicielles sur Internet de s'échanger des données

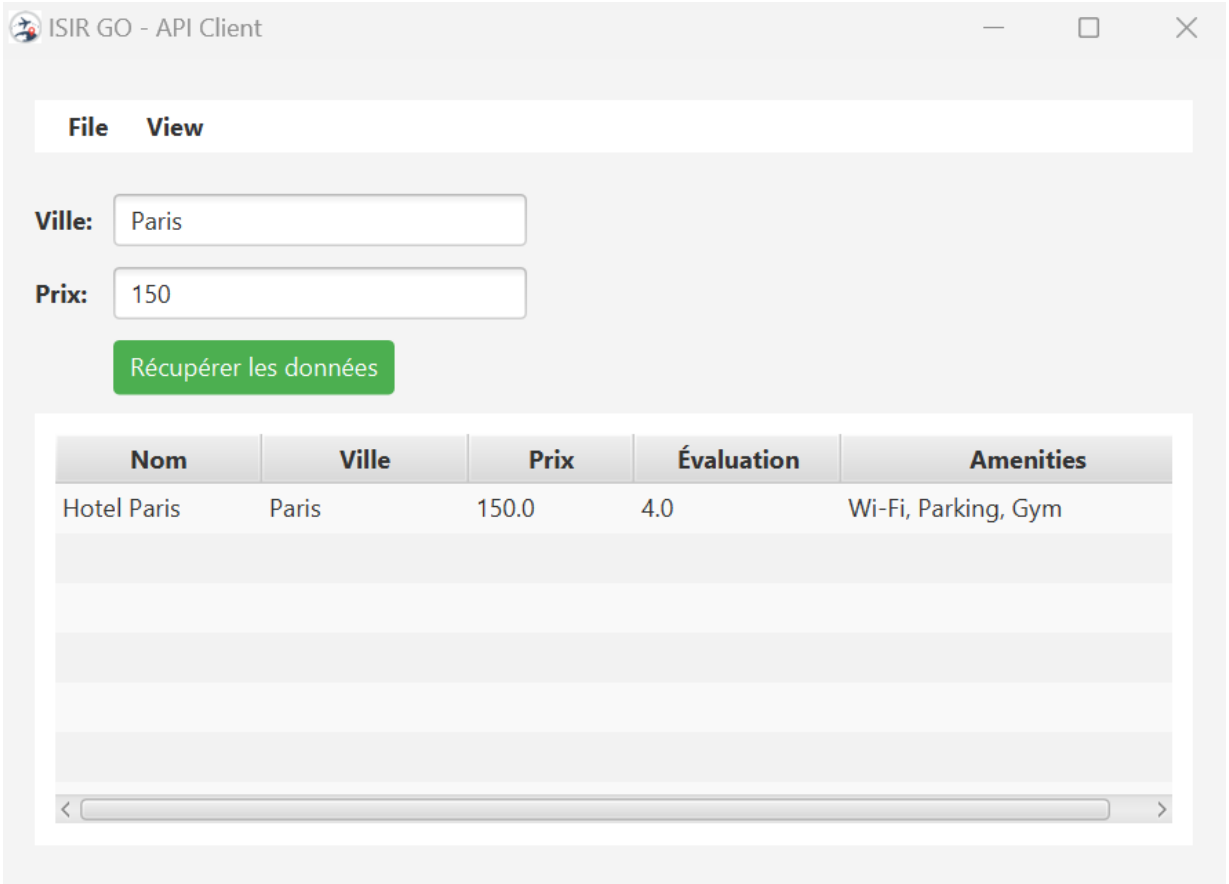
Exemple



Trivago

Motivation

ISIRAGO



Évolution des services Web

Remote Procedure Call (RPC)

1. Protocole de communication qui permet à un programme d'appeler une procédure située sur un autre ordinateur.
2. Flexible : peut être utilisé avec n'importe quel langage de programmation et système d'exploitation.
3. Moins performant que d'autres protocoles, tels que HTTP.

Simple Object Access Protocol (SOAP)

1. Un protocole léger basé sur XML utilisé pour l'échange d'informations dans les environnements d'applications
2. SOAP permet à la requête de l'utilisateur d'interagir avec d'autres langages de programmation.
3. Il utilise le format XML des données pour transférer des messages sur le protocole HTTP décentralisés et distribués

Representational State Transfer (REST)

1. Approche architecturale pour le développement de services Web
2. Modèle client-serveur sans état
3. Transfert d'informations dans divers formats via HTTP
   - JSON, HTML, XLT, texte brut
   - JSON est le format le plus couramment utilisé

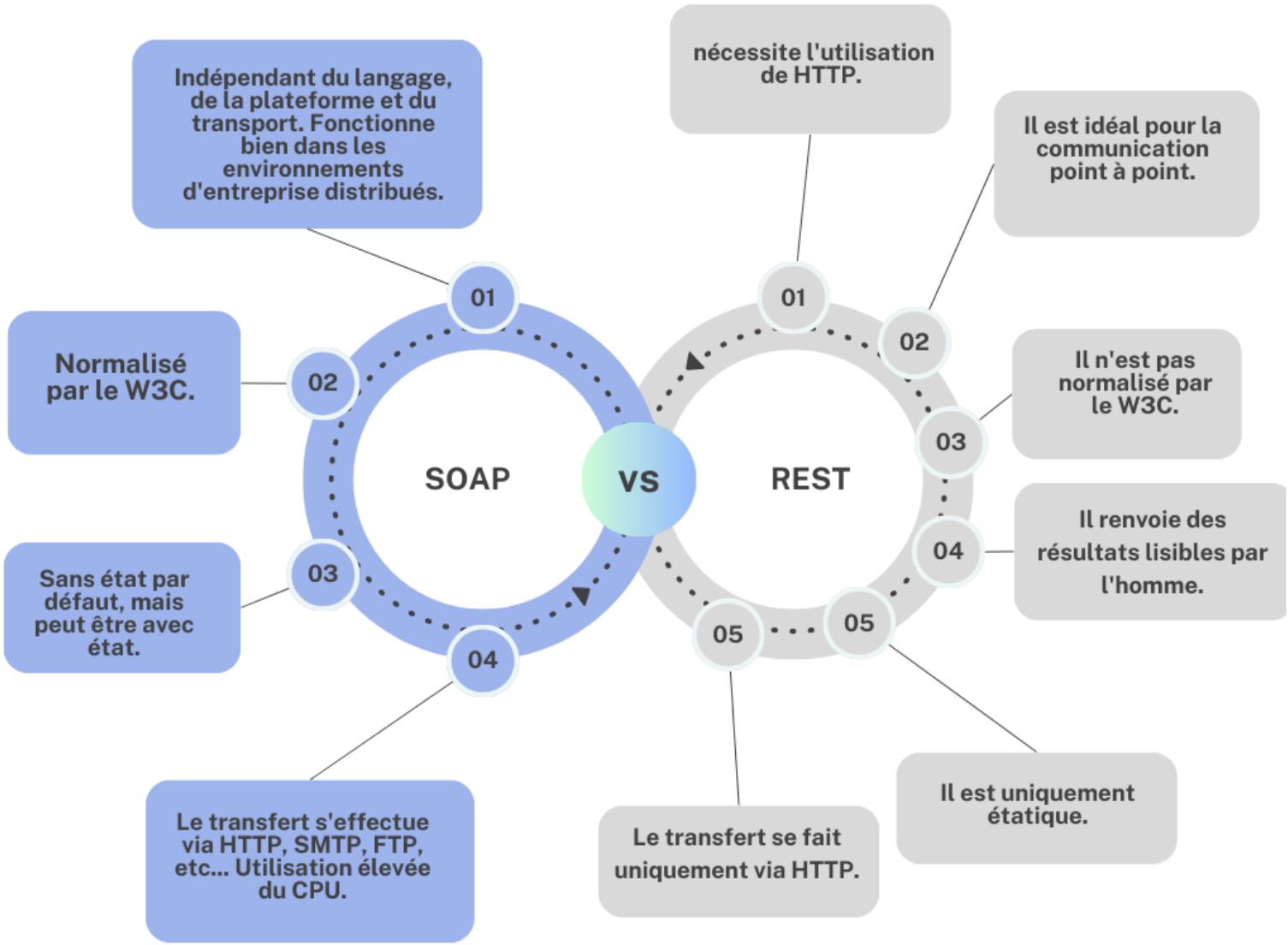
GraphQL

1. GraphQL est un langage de requête flexible qui permet aux clients de spécifier les données dont ils ont besoin de manière précise.

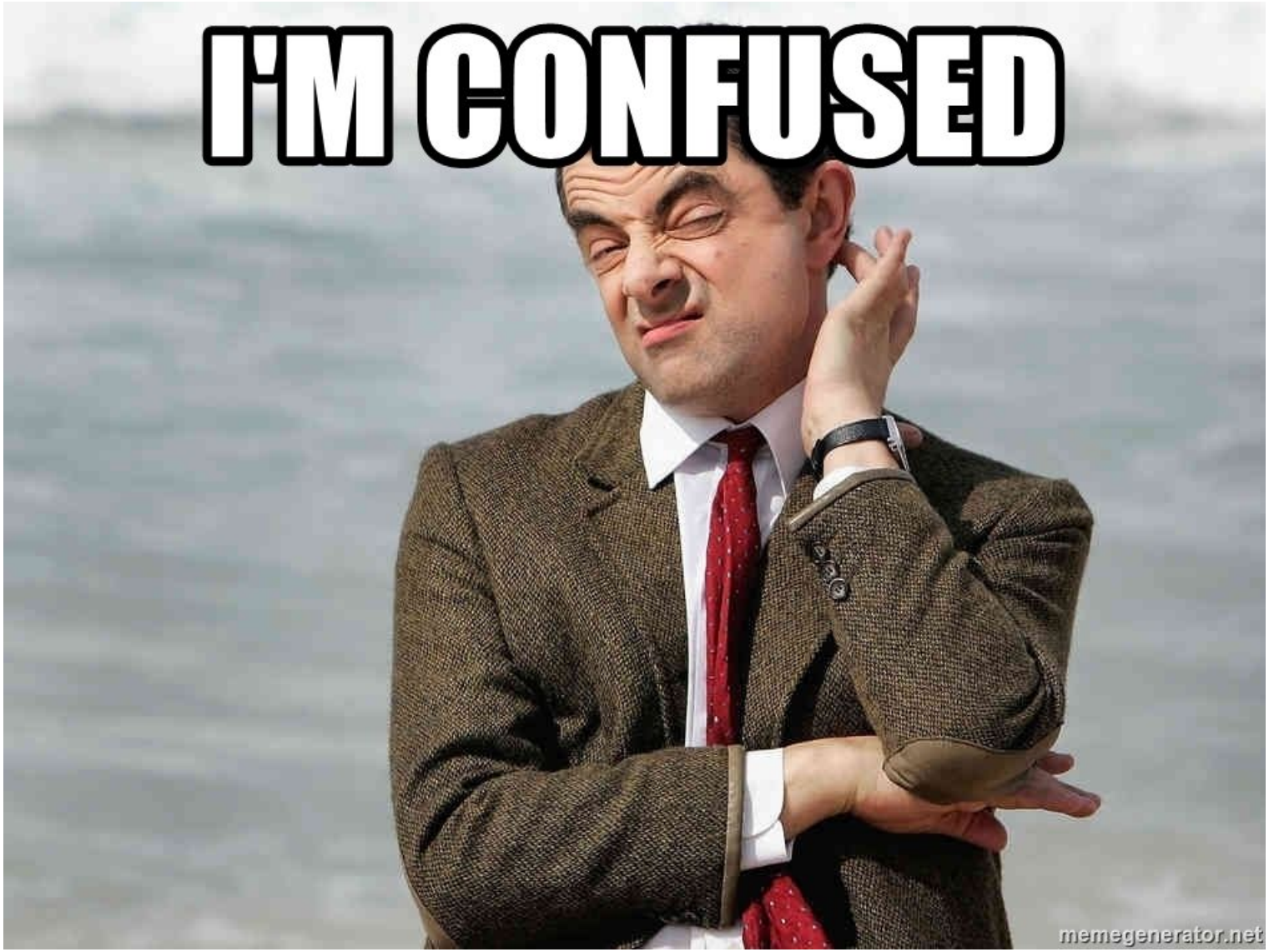
2. Sa structure hiérarchique simplifie la récupération efficace de données complexes en une seule requête.

3. Avec un seul point d'entrée et une documentation automatique, GraphQL offre une solution adaptable pour la gestion d'API.

Comparaison soap et REST



Attention!!



Web Services vs Web Applications

Site web

Services web

Une application Web est accessible via un navigateur Web sur la machine du client.

Un service Web est un système logiciel permettant l'interaction entre différentes machines via un réseau.

Une application Web est une entité complète avec une interface graphique.

Les services Web, la plupart du temps, n'ont pas d'interface utilisateur propre, étant utilisés comme composants dans une application.

Il utilise des langages comme HTML, CSS et JavaScript pour créer une interface utilisateur interactive.

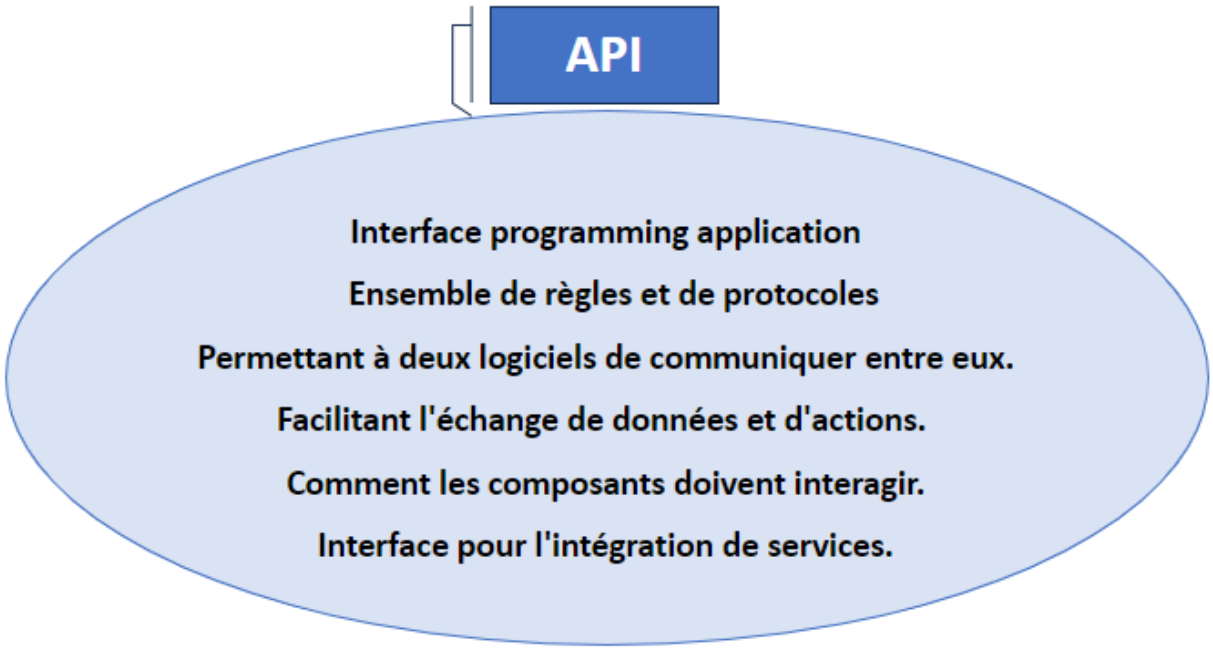
Les services Web utilisent des protocoles standardisés, établissant ainsi un langage commun pour la communication entre différentes applications,

Ils sont généralement destinés à être consultés par des utilisateurs pour obtenir des informations, des services, ou interagir avec des applications en ligne.

les services Web sont utiles pour la communication ou le transfert de données entre des applications Web sur différentes plates-formes.

Concept de base de web services

Définition d'une API



Différents types d'API

Il existe de nombreux types d'API, dont les suivants :

- API REST:

  Les API REST sont basées sur le modèle architectural REST (Representational State Transfer). Elles utilisent des méthodes HTTP standard, telles que GET, POST, PUT et DELETE, pour interagir avec les ressources.

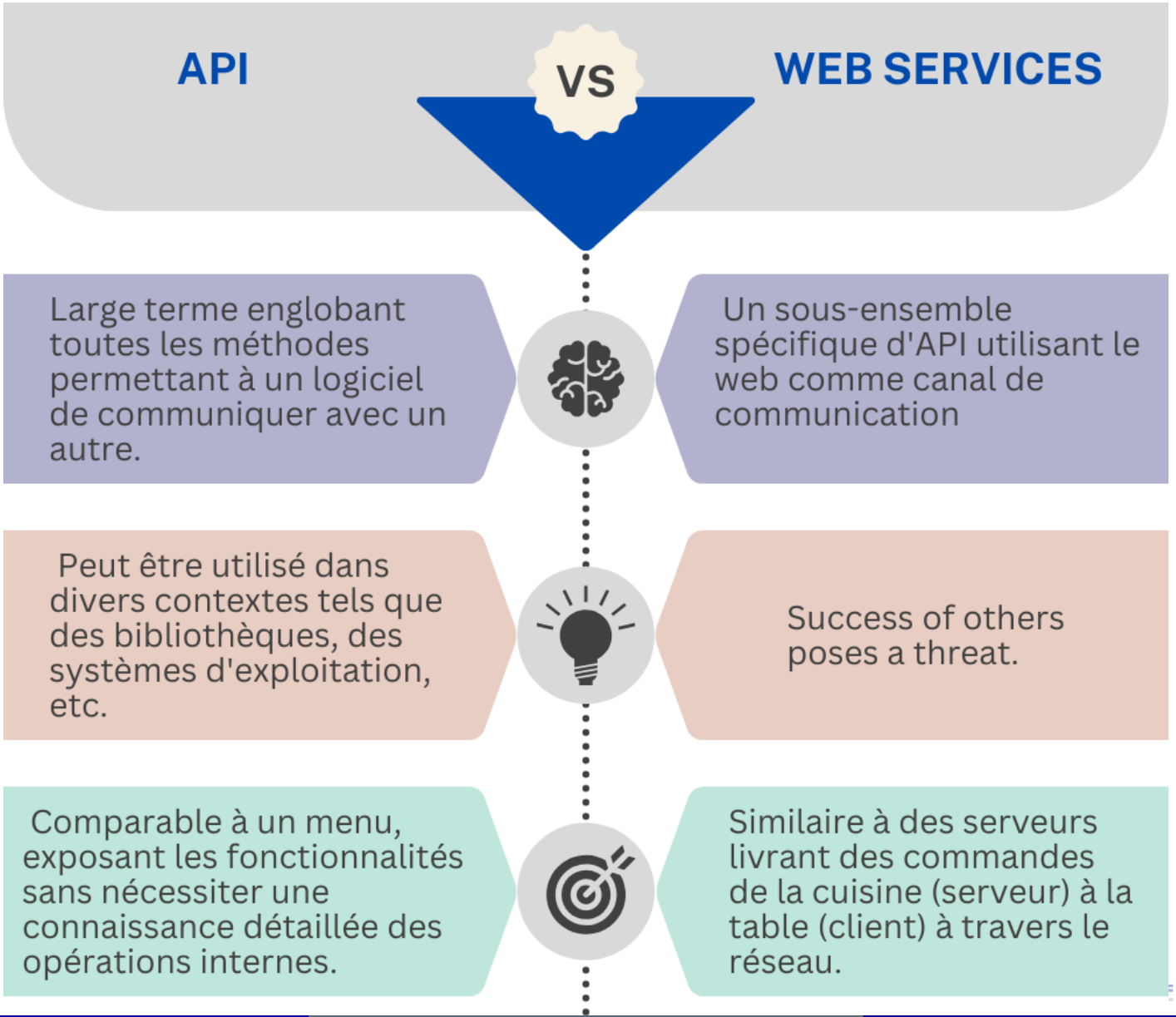
- API SOAP:

  Les API SOAP sont basées sur le protocole SOAP (Simple Object Access Protocol). Elles utilisent un format XML pour représenter les données et les méthodes.

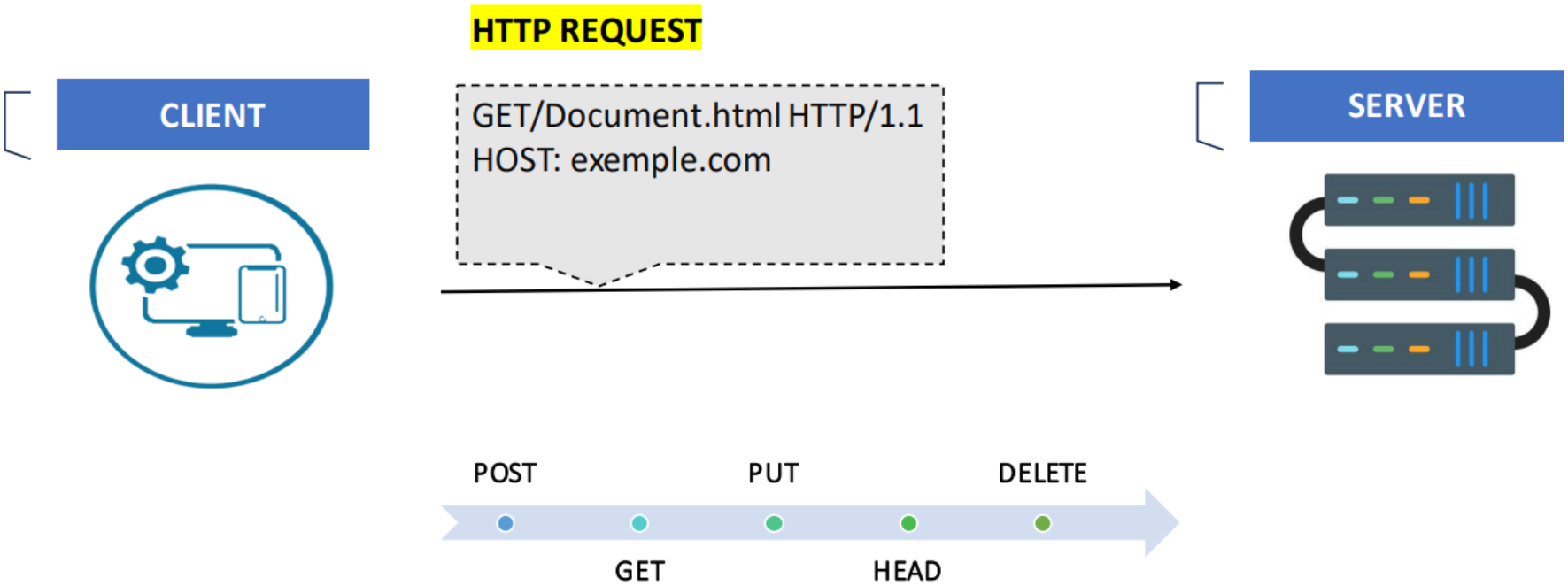
- API GraphQL:

  Les API GraphQL sont un type d'API basé sur des requêtes. Elles permettent aux applications de demander uniquement les données dont elles ont besoin.

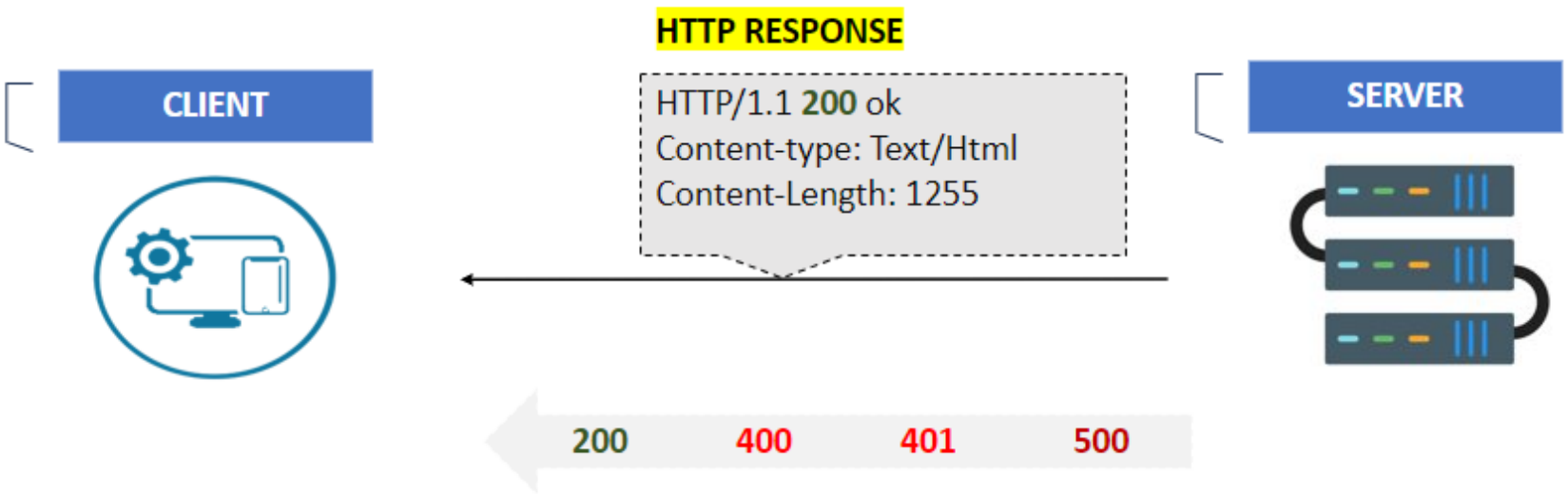
API vs Web Service



Client serveur Architecture



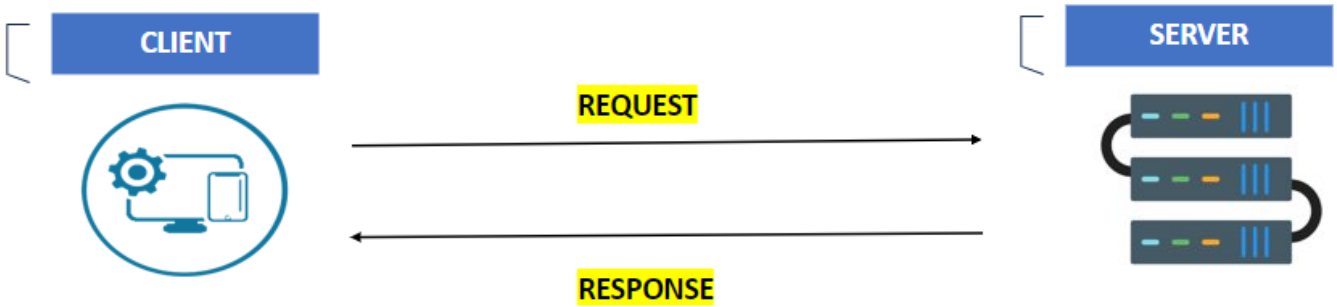
Client serveur Architecture



Communication Protocols: HTTP, HTTPS

- HTTP

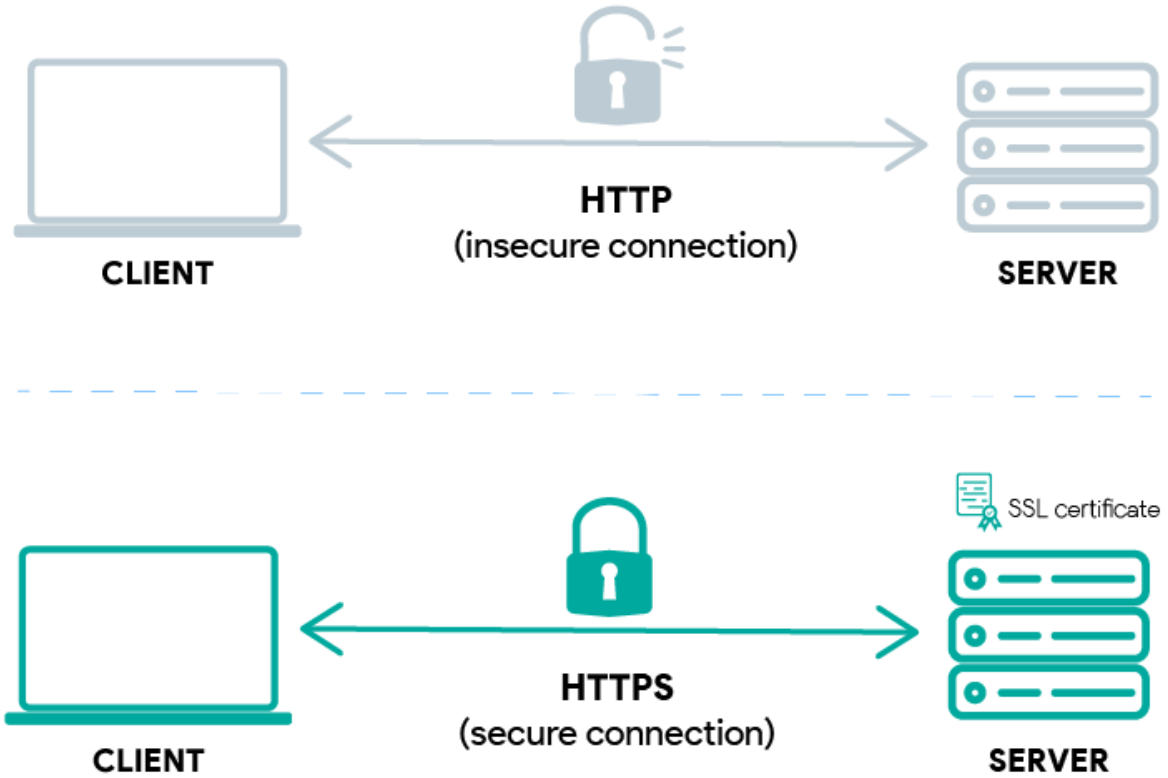
  HTTP est un protocole qui permet aux navigateurs Web de télécharger des pages Web et d'autres ressources à partir de serveurs Web.



Communication Protocols: HTTP, HTTPS

- HTTPS

  HTTPS est une version sécurisée de HTTP qui chiffre les communications entre le navigateur Web et le serveur Web.



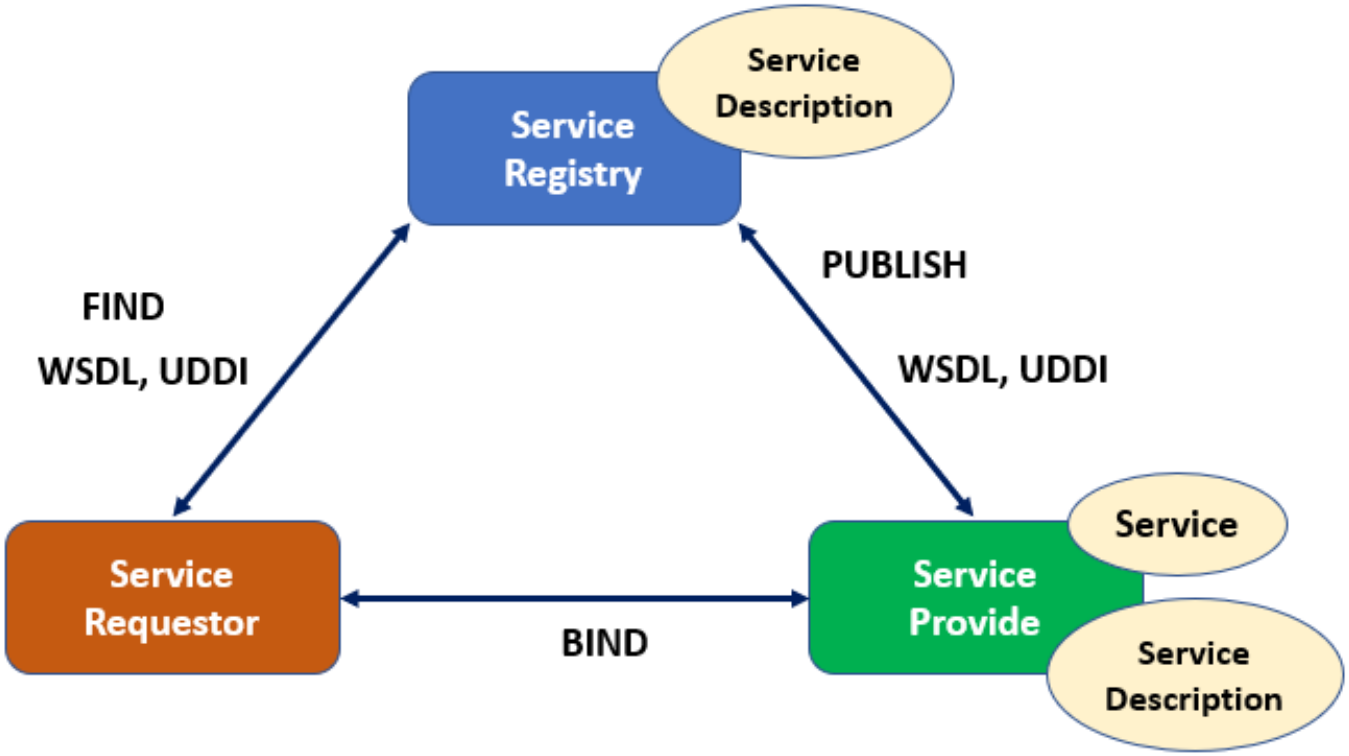
les composants de web services

Qu'est-ce que WSDL ?

- Langage XML pour décrire les services web
- Un service web est décrit comme :
  - Un ensemble de points de communication (ports)
- Un point de communication est composé de deux parties:
  - Définitions abstraites des opérations et des messages
  - Liaison concrète à un protocole réseau (et une adresse de point de terminaison correspondante) et un format de message
- Pourquoi cette séparation?
  - Améliorer la réutilisabilité (comme nous le verrons dans la référence à UDDI pour le document WSDL)

voilà un exemple

UDDI



UDDI defines a way to publish and find information about Web services.

Format de données


{JSON}
</>
<XML>

XML

- Une structure arborescente.
- Les balises sont utilisées pour encadrer un contenu il y a:
  - une balise ouvrante
  - une balise fermante
- il permet de structurer l'information sous une forme plus robuste.
- Déployable.

JSON

- JSON est un format de données léger et lisible par l'homme qui est utilisé.
- pour représenter des données structurées.
- JSON est basé sur un ensemble de règles simples qui sont faciles à apprendre et à utiliser.
- Les données JSON sont représentées sous forme d'objets, de tableaux et de chaines de caractères.

MIME

- Types MIME les plus courants

text/plain	Un fichier texte
text/plain;charset=utf-8	Un fichier texte encodé en UTF-8
text/html	Un fichier HTML
application/x-www-form-urlencoded	Le format de données pour la soumission d'un formulaire HTML
text/xml ou application/xml	Un fichier XML
text/json ou application/json	Un fichier JSON
image/jpeg	Une image au format jpeg
application/octet-stream	Un flux d'octets sans type particulier. Il s'agit du format par défaut si l'en-tête `Content-type` est absent.

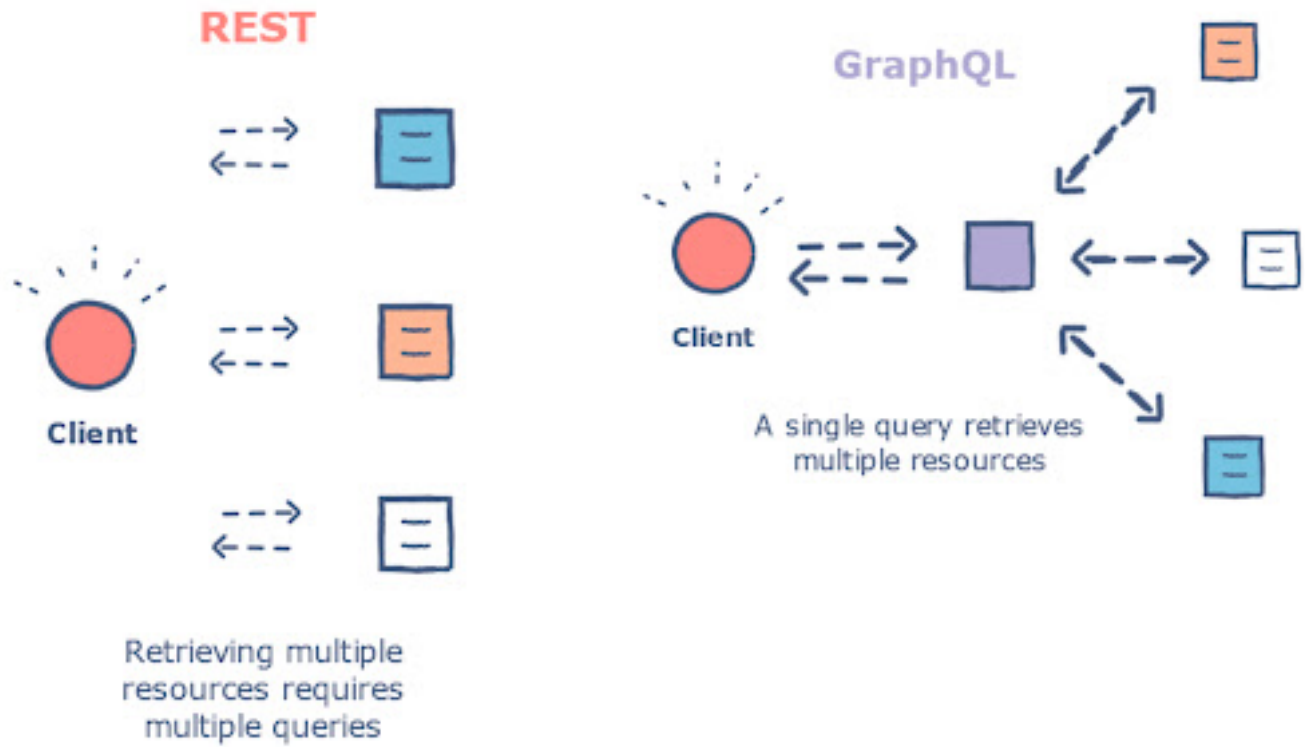
REST

Pourquoi REST ?

Il existe plusieurs raisons pour lesquelles REST est un choix populaire pour les API Web. Voici quelques-unes des principales raisons :

1. REST est simple et léger. Il s'appuie sur des protocoles standards tels que HTTP et JSON, ce qui le rend facile à comprendre et à mettre en œuvre.
2. REST est flexible. Il peut être utilisé pour une variété d'applications, de la simple exposition de données à la mise en œuvre de services complexes.
3. REST est évolutif. Il peut être facilement mis à l'échelle pour répondre aux besoins croissants des applications.

L'architecture REST vs GraphQL:

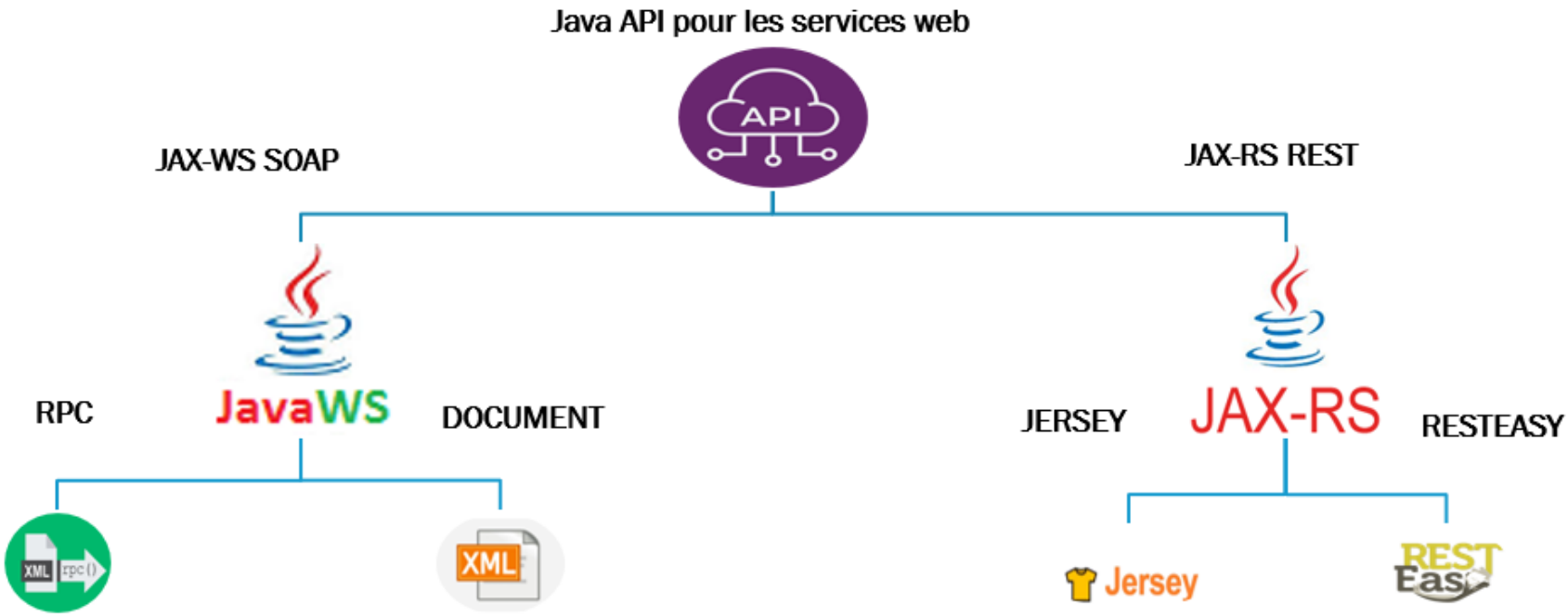


REST → *L'utilisation* :

1. Utiliser dans le developpement des applications orientée ressources (ROA).
2. Les applications respectant l'architecture REST dites RESTFul.

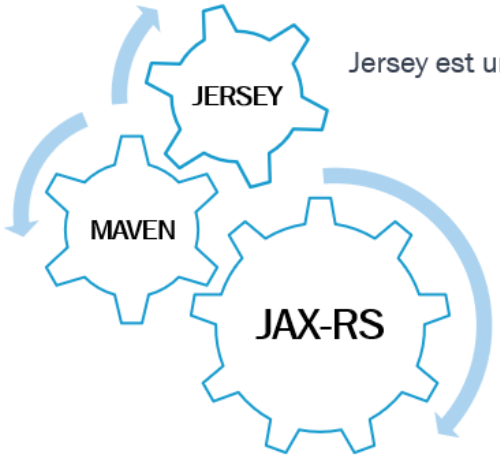
Developpement d'un service web REST avec java

Présentation JAX-RS (RESTful)



PRÉSENTATION DE JAX-WS

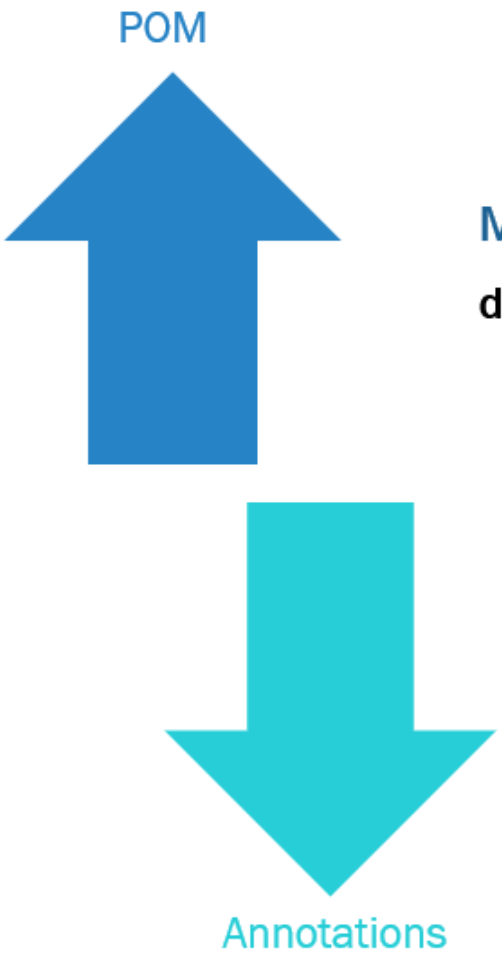
En utilisant Maven, un gestionnaire de dépendances et un outil de gestion de projet, vous pouvez configurer facilement un projet Java pour développer des APIs RESTful avec JAX-RS et Jersey.



Jersey est un Framework JAX-RS open-source populaire.

JAX-RS est l'API Java pour les services web RESTful, permettant de créer des APIs web en utilisant le style architectural REST.

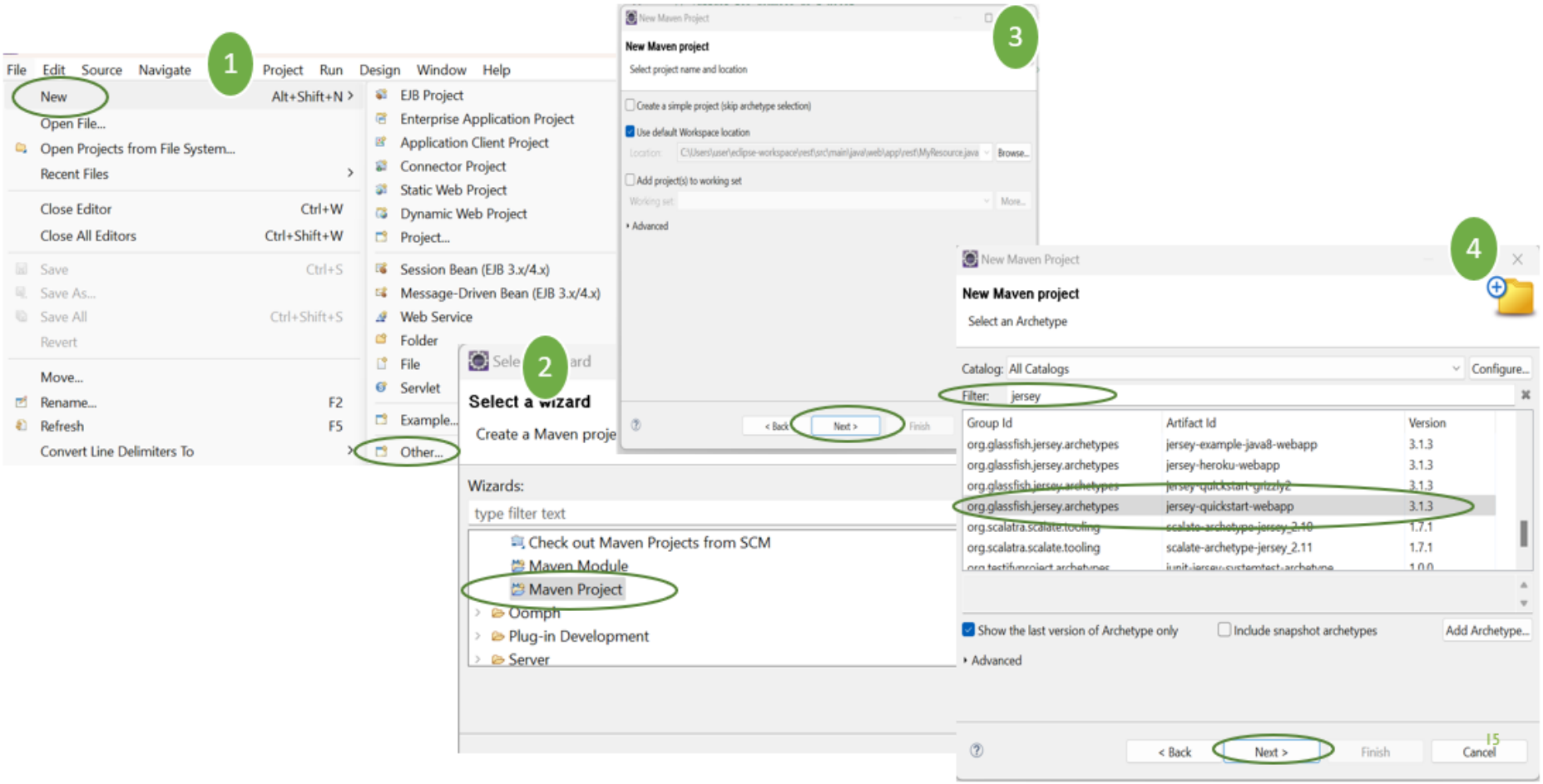
PRÉSENTATION DE JAX-WS



MAVEN simplifie la gestion des dépendances et des configurations.

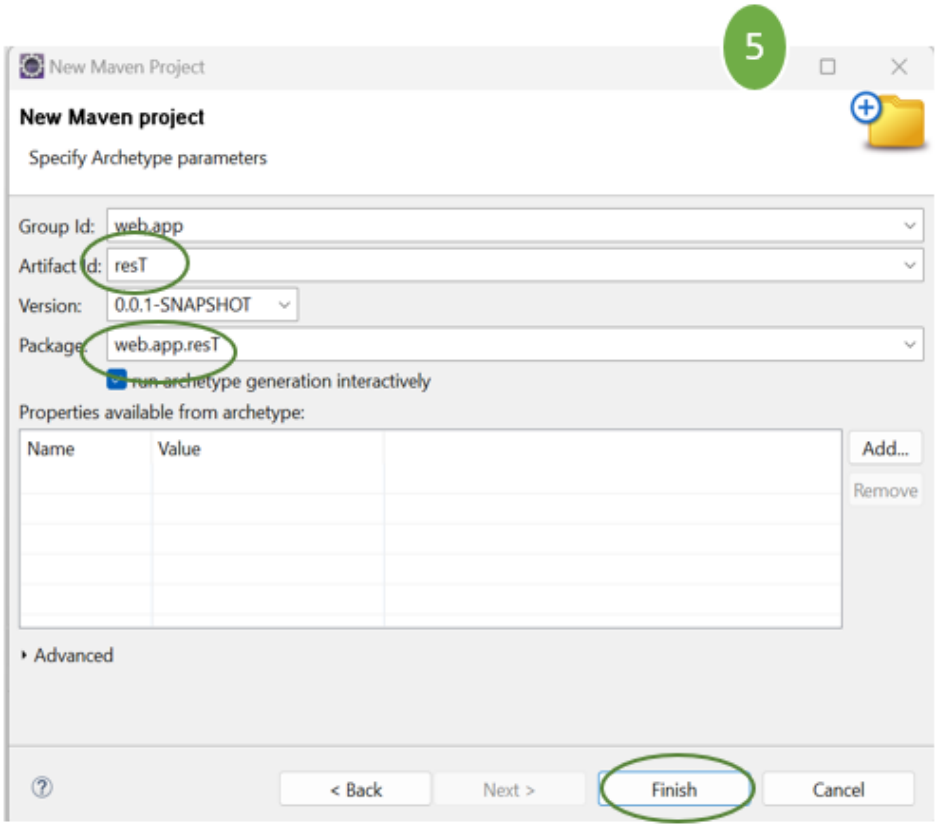
JERSEY facilite le développement des services web RESTful en Java.

configuration

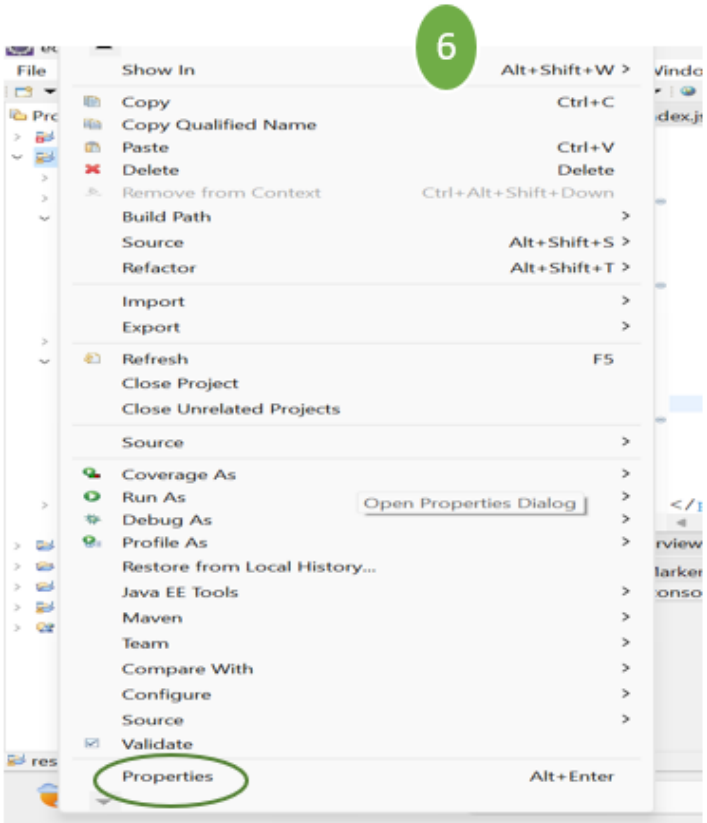


configuration

Ajouter le Group ID et Le Artifact ID le nom de projet

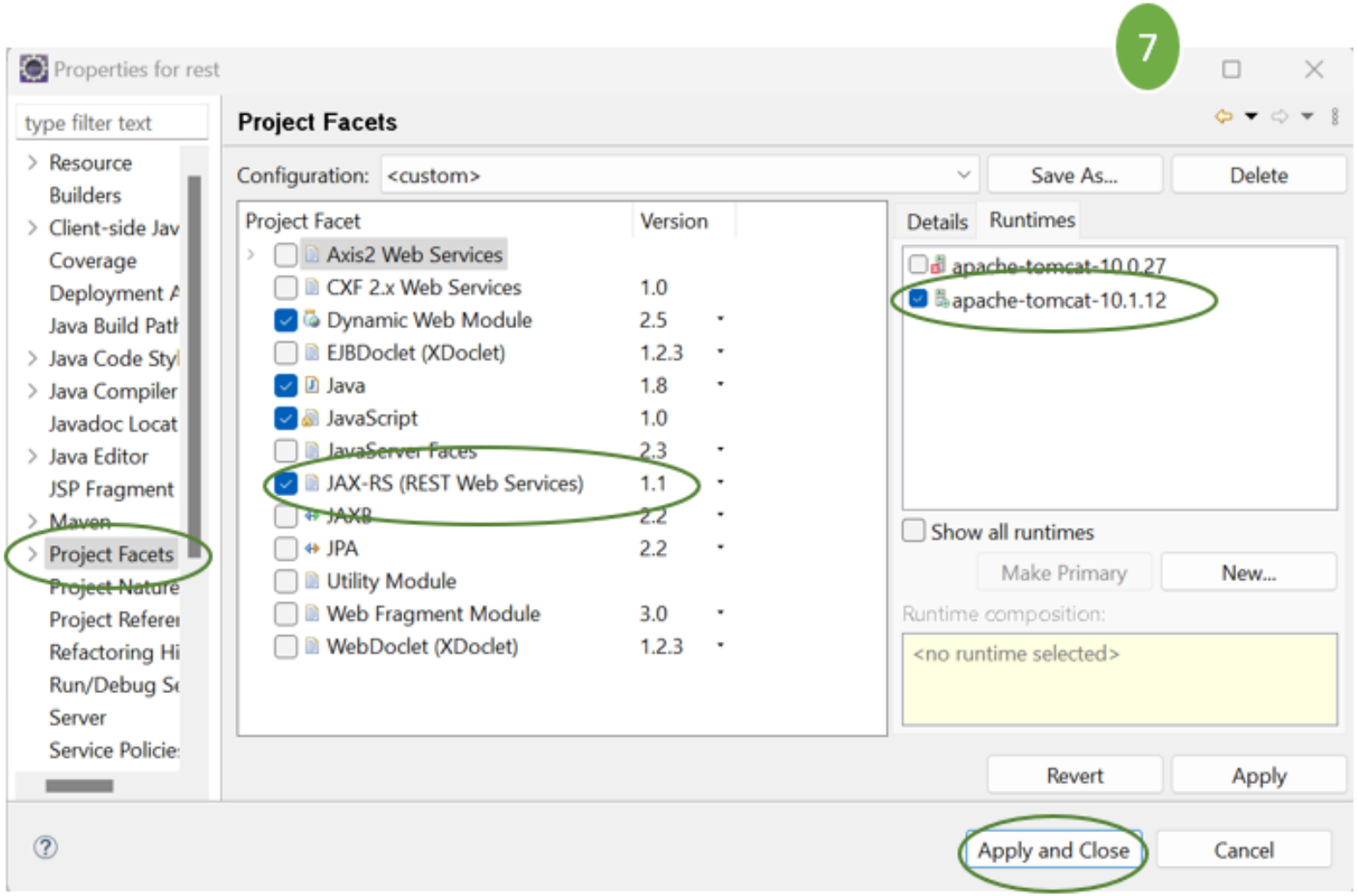


Selectionner le projet ->double clique à droite-> properties



configuration

Déclencher le serveur et jax-rs



configuration

Configuration de la servlet

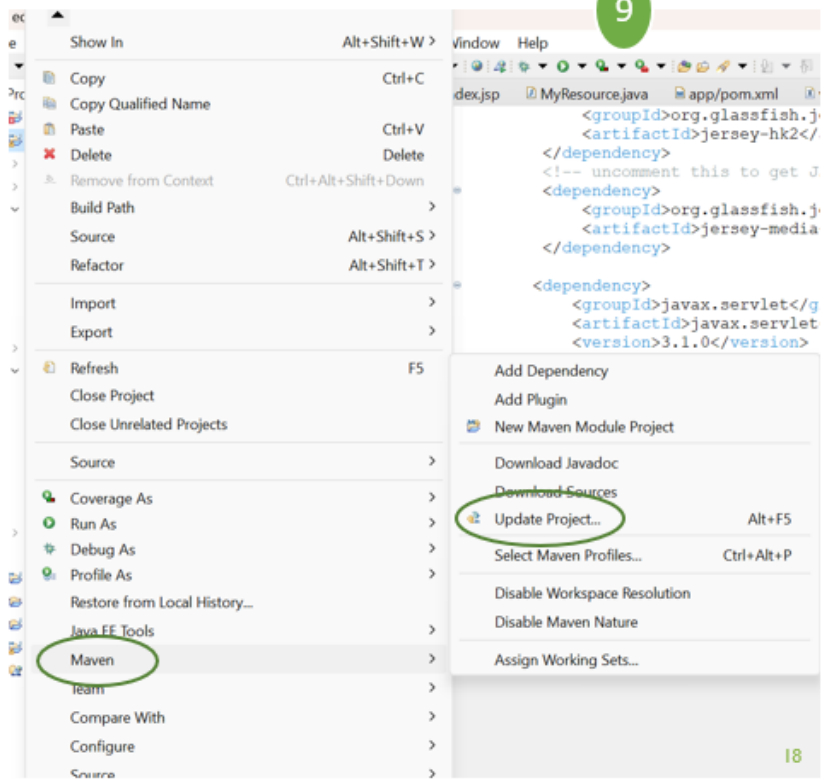
Ajouter l'independance dans le fichier POM.xml

8

```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
    <scope>provided</scope>
</dependency>
```

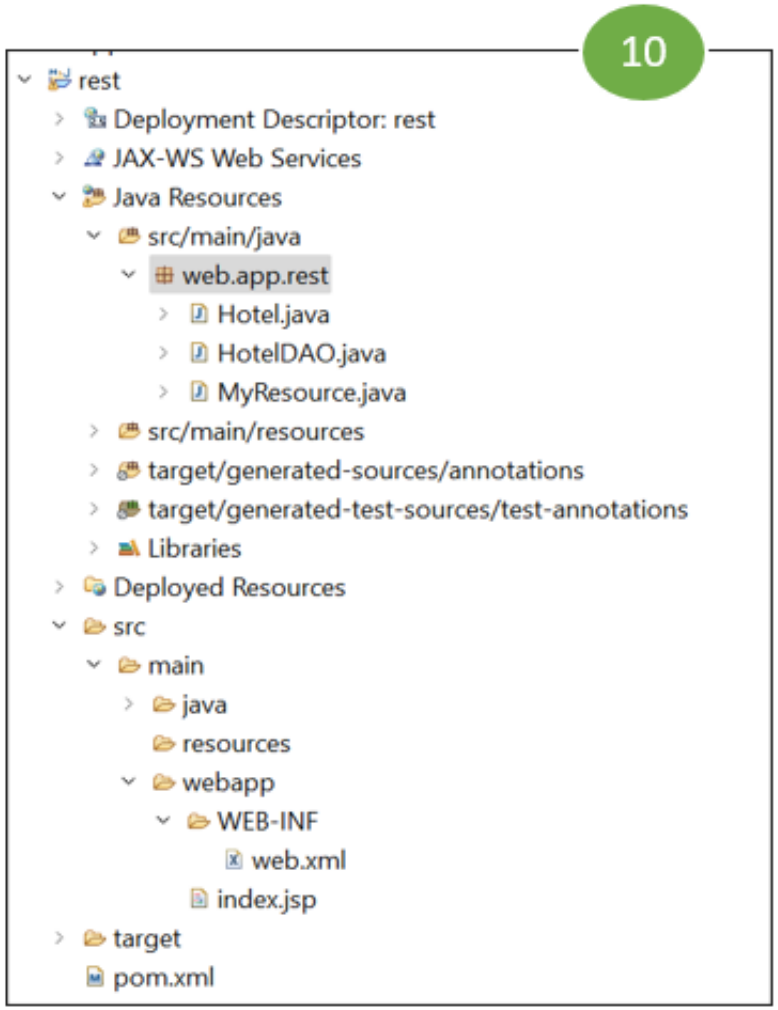
Rafraichir le projet pour appliquer les mises à jour

9

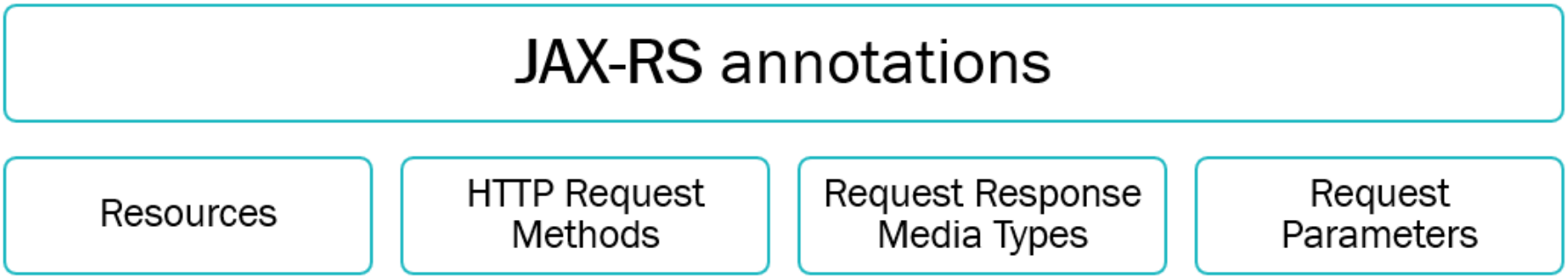


configuration

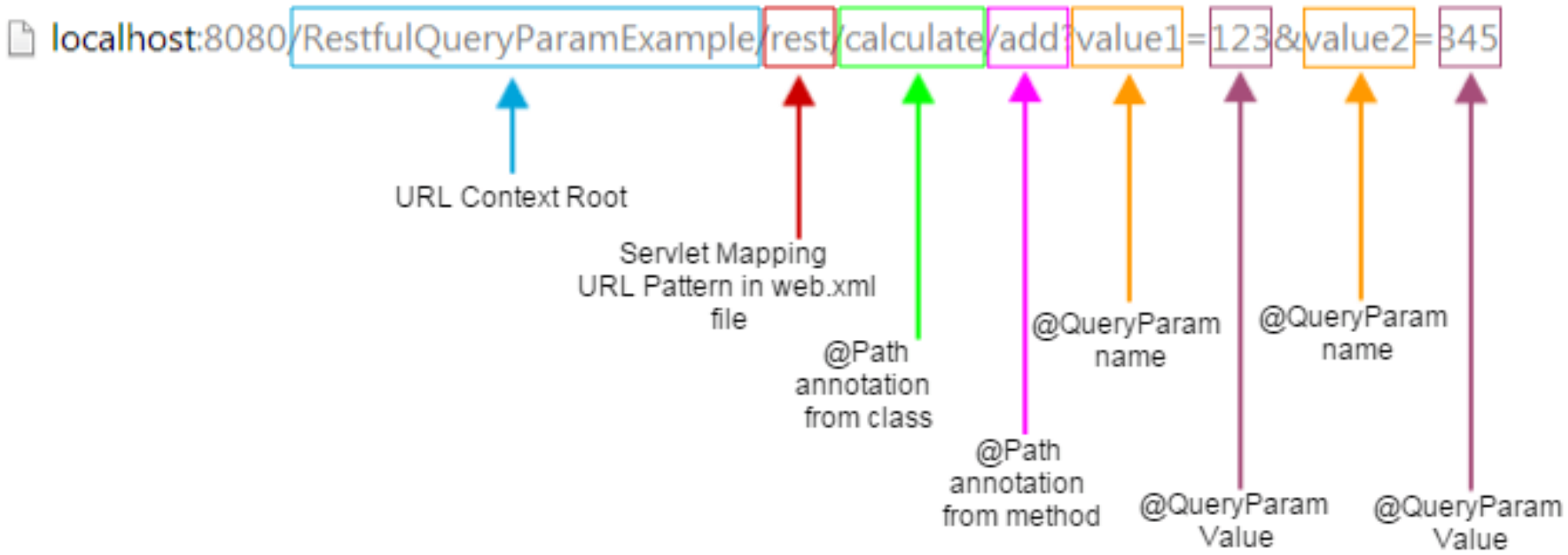
la structure de projet



Implémentation des ressources avec JAX-RS


JAX-RS annotations
Resources
HTTP Request Methods
Request Response Media Types
Request Parameters

Structure URL



Annotation Path

- Permet de rendre une classe accessible par une requête HTTP.
- Elle definit la source des ressources (Root Racine Ressouces).
- La valeur donnee correspond a L'uri relative de la ressource.
- Peut etre utiliser pour annoter les méthodes d'une classe.

Annotation –Les Méthodes HTTP–

- Permettent de mapper une méthode à un type de requete http.
- Ne sont pas utilisable que sur des méthodes.
- Le nom de la méthode n'a pas d'importance, JAX détermine la méthode à exécuter en fonction de le requête.

1

[1] NB: Si aucune méthode Java n'est déclarée pour traiter la méthode HTTP de la requête entrante, alors le serveur répondra automatiquement le code erreur 405 (Method not allowed).

Paramètre de chemin

Comme chaque ressource Web est identifiée par une URI, il est important pour le serveur de pouvoir récupérer dans le chemin les informations qui vont lui permettre de réaliser cette identification dynamiquement. Par exemple, le serveur peut extraire du chemin de la ressource une clé primaire lui permettant d'effectuer une recherche en base de données.

Avec JAX-RS, on déclare des paramètres de chemin entre accolades et on utilise l'annotation javax.ws.rs.PathParam pour récupérer leur valeur dans les paramètres des méthodes :

```
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
@Path("/user/{id}")
public class UserResource {
@GET
public User get(@PathParam("id") long id) { //.... }
}
```

Paramètre de requête

@QueryParam

Comme pour les paramètres de chemin, il est possible de récupérer la valeur des paramètres de la requête comme arguments des méthodes de la ressource JAX-RS grâce à l'annotation **@javax.ws.rs.QueryParam**.

```
@GET
public List<User> search(@QueryParam("name") String name) { // ... }
```

@FormParam

Les données transmises *via* un formulaire HTML peuvent être récupérées comme arguments des méthodes de la ressource JAX-RS grâce à l'annotation **@javax.ws.rs.FormParam**.

Data Binding @Consumes / @Produces

Remarque

Lorsqu'une méthode d'une ressource retourne une instance d'un objet Java, JAX-RS va tenter de créer une réponse au format souhaité en fonction de l'annotation @Produces.

- Il existe un ensemble de règles par défaut permettant de passer d'un objet Java à un document XML ou JSON.
- On appelle l'ensemble de ces règle le data binding.

Génération des Réponses

Concept

Parfois, il n'est pas suffisant de retourner une instance d'un objet Java en laissant à JAX-RS le soin de créer la réponse HTTP. C'est notamment le cas si l'on souhaite retourner un code statut HTTP différent de 200 ou ajouter des en-têtes HTTP dans la réponse. Pour cela, il faut retourner une instance de la classe javax.rs.core.Response. Cette classe suit le design pattern builder et offre un ensemble de méthodes utilitaires pour construire la réponse. Au final, il suffit d'appeler la méthode build() et retourner le résultat.

Génération des Réponses

```
package fr.epsi;
import java.net.URI;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
 @Path("/user")
 public class UserResource {
@GET
@Path("/{name}")
@Produces({MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML})
 public Response get(@PathParam("name") String name) {
User user; // ...
return Response.ok(user).build();
}}
```

Figure: Exemple d'utilisation de la classe javax.rs.core.Response

Thank you!