

# Communication Web - REST -

**Groupe des étudiants : CIR2**

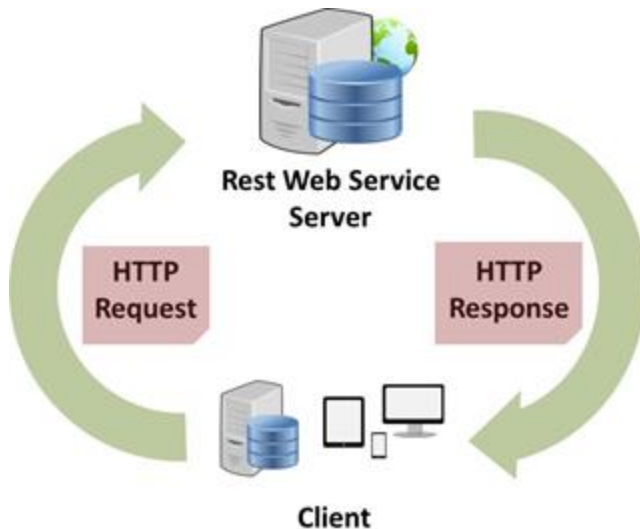
Ayoub KARINE

E-Mail : [ayoub.karine@isen-ouest.yncrea.fr](mailto:ayoub.karine@isen-ouest.yncrea.fr)

Numéro de bureau : A3-78

# Définition

- **REST (REpresentational State Transfer)** est une architecture de développement Web permettant de construire des applications Web « orientées ressource » (i.e. Resource-oriented architecture). Les applications qui respectent cette architecture sont dites « RESTful »



# Principes de l'architecture REST

- Pour standardiser les échanges client/serveur. L'architecture REST est basée sur les principes suivants :
  - **Adressabilité** : chaque ressource est identifiée par son URI

`http://serveur:port/chemin/vers/ressource?c1=v1&c2=v2`

← *protocole + serveur*    ← *chemin vers la ressource*    ← *paramètres de la requête* →

# Principes de l'architecture REST

- Pour standardiser les échanges client/serveur. L'architecture REST est basée sur les principes suivants :
  - **Adressabilité** : chaque ressource est identifiée par son URI
  - **Uniformité des interfaces** : toute requête correspond à un verbe (CRUD)

Verbe	Méthode HTTP
Create (créer)	POST

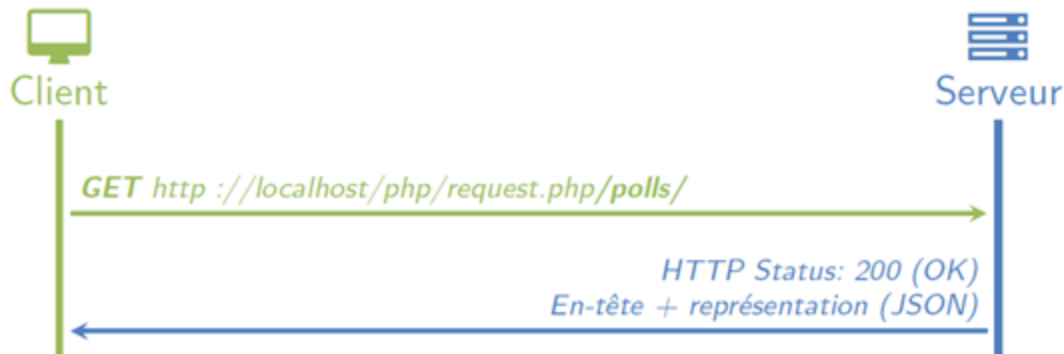


*Création du sondage de titre s1 par tibo.*

# Principes de l'architecture REST

- Pour standardiser les échanges client/serveur. L'architecture REST est basée sur les principes suivants :
  - **Adressabilité** : chaque ressource est identifiée par son URI
  - **Uniformité des interfaces** : toute requête correspond à un verbe (CRUD)

Verbe	Méthode HTTP
Retrieve (Récupérer)	GET



*Récupération de tous les sondages.*

# Principes de l'architecture REST

- Pour standardiser les échanges client/serveur. L'architecture REST est basée sur les principes suivants :
  - **Adressabilité** : chaque ressource est identifiée par son URI
  - **Uniformité des interfaces** : toute requête correspond à un verbe (CRUD)

Verbe	Méthode HTTP
Update (Mettre à jour)	PUT



*Mise à jour du sondage n°5 avec le choix 1.*

# Principes de l'architecture REST

- Pour standardiser les échanges client/serveur. L'architecture REST est basée sur les principes suivants :
  - **Adressabilité** : chaque ressource est identifiée par son URI
  - **Uniformité des interfaces** : toute requête correspond à un verbe (CRUD)

Verbe	Méthode HTTP
Delete (Supprimer)	DELETE



*Suppression du sondage n°1.*

# Principes de l'architecture REST

- Pour standardiser les échanges client/serveur. L'architecture REST est basée sur les principes suivants :
  - **Adressabilité** : chaque ressource est identifiée par son URI
  - **Uniformité des interfaces** : toute requête correspond à un verbe (CRUD)

Verbe	Méthode HTTP
<b>Create</b> (créer)	POST
<b>Retrieve</b> (Récupérer)	GET
<b>Update</b> (Mettre à jour)	PUT
<b>Delete</b> (Supprimer)	DELETE



# Principes de l'architecture REST

- Pour standardiser les échanges client/serveur. L'architecture REST est basée sur les principes suivants :
  - **Adressabilité** : chaque ressource est identifiée par son URI
  - **Uniformité des interfaces** : toute requête correspond à un verbe
  - **Communication stateless** :
    - o aucune gestion des états côté serveur (pas de session, pas de cookies, ...).
    - o chaque requête envoyée vers le serveur doit contenir toutes les informations nécessaires à son traitement. L'état est alors conservé dans les ressources  
⇒ minimisation des ressources systèmes
    - o les requêtes sont dites autonomes


# TP2

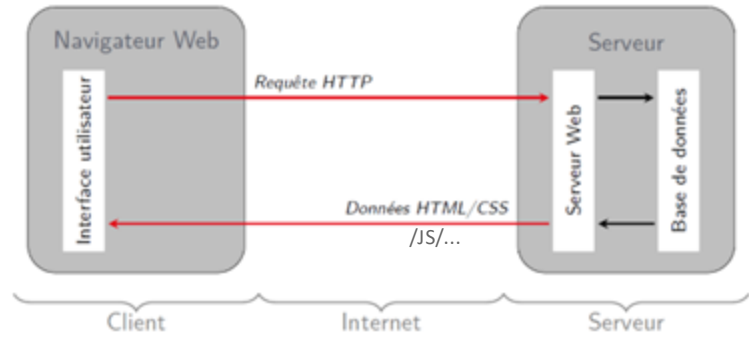
## Première communication :

### TP n°2 REST

#### Liste des tweets

Tweet >

 Envoyer



## Deuxième communication :

### TP n°2 REST

#### Liste des tweets

dt1 : Un tweet des CR1 !!

dt2 : Un tweet des CR2 !!

dt3 : Un tweet des CR3 !!

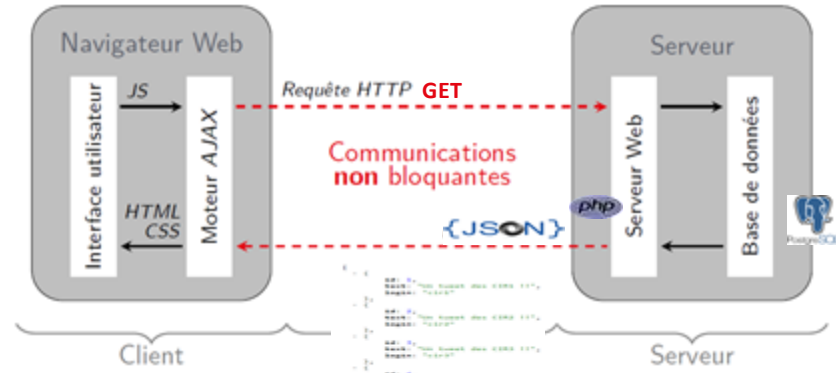
mt1 : Un tweet des M1 !!

mt2 : Un tweet des M2 !!

Tweet >

 Envoyer

### Le modèle de communication avec AJAX :



# TP2

Autres communication (ajout + mise à jour + suppression) :

Le modèle de communication avec **AJAX** :

