

API Rest

API

Définition

- Application Programming Interface
- Protocole d'échange et de communication entre différents services

API

Pour le web

- Permet de fournir les données à un ou des clients depuis un serveur
- SOAP, XML-RPC, « custom », ...

REST

Les avantages

- Réfléchi par **Roy Fielding**
- Representational State Transfer
- Basé sur le protocole HTTP

- Interface uniforme
- Découplage client-serveur
- Sans état

REST Richardson

Le modèle de Richardson

- Plusieurs niveaux de maturités
- 4 niveaux (de 0 à 3)
- HATEOAS

Richardson

Niveau 0

- Aucune maturité
- Utilisation de HTTP
- Pas de cohérence dans les URI
- Pas de cohérence dans le format de réponse

POST http://localhost:80/myWok

```
{  
  "type": "woks"  
}
```

```
<carteRestaurant>  
  <woks>  
    <wok id="porc" prix="8.5" />  
    <wok id="poulet" prix="10.5" />  
  </woks>  
</carteRestaurant>
```

Richardson

Niveau 1

- Une ressource = une URI

POST http://localhost:80/woks

```
<woks>
  <wok id="porc" prix="8.5" />
  <wok id="poulet" prix="10.5" />
</woks>
```

###

POST http://localhost:80/woks/1

```
<wok id="porc" prix="8.5" />
```

Richardson

Niveau 2

- Utilisation des verb HTTP
- Utilisation des codes retours

GET http://localhost:80/woks

```
<woks>
  <wok id="porc" prix="8.5" />
  <wok id="poulet" prix="10.5" />
</woks>
```

###

POST http://localhost:80/woks

```
{
  "id": "crevette"
  "prix": "12"
}
```

```
<wok id="porc" prix="8.5" />
```


Richardson

Niveau 3

- Hateoas

GET http://localhost:80/woks

```
<woks>
  <wok id="porc" prix="8.5" />
  <wok id="poulet" prix="10.5" />
</woks>
<link href='work/porc' >
<link href='work/poulet' >
```

API REST

Exemple

GET /users
Accept: application/json

```
[{  
  "id": 32,  
  "name": "Michel"  
}]
```

###

GET /users/{id}
Accept: application/json

```
{  
  "id": 32  
}
```

```
{  
  "id": 32,  
  "name": "Michel"  
}
```

CRUD

Create - Read - Update - Delete

POST /users
Accept: application/json

```
{  
  "id": 32,  
  "name": "Michel"  
}
```

###

PUT /users
Accept: application/json

```
{  
  "id": 32,  
  "name": "Marc"  
}
```

###

DELETE /users/{id}
Accept: application/json

```
{  
  "id": 32  
}
```

Outils

- Insomnia pour tester