

API Introduction L'ARCHITECTURE REST

DEMBELE Mouhamoud

Software Ingenieur - DevOps Officer - Consultant IT

Ce contenu est basé sur les
transparent de Fadel Chafai

Qu'est-ce qu'une API?

Application Programming Interface

Une API est un logiciel qui offre des services à d'autres logiciels.

Une API spécifie comment un consommateur peut consommer les services que l'API expose:

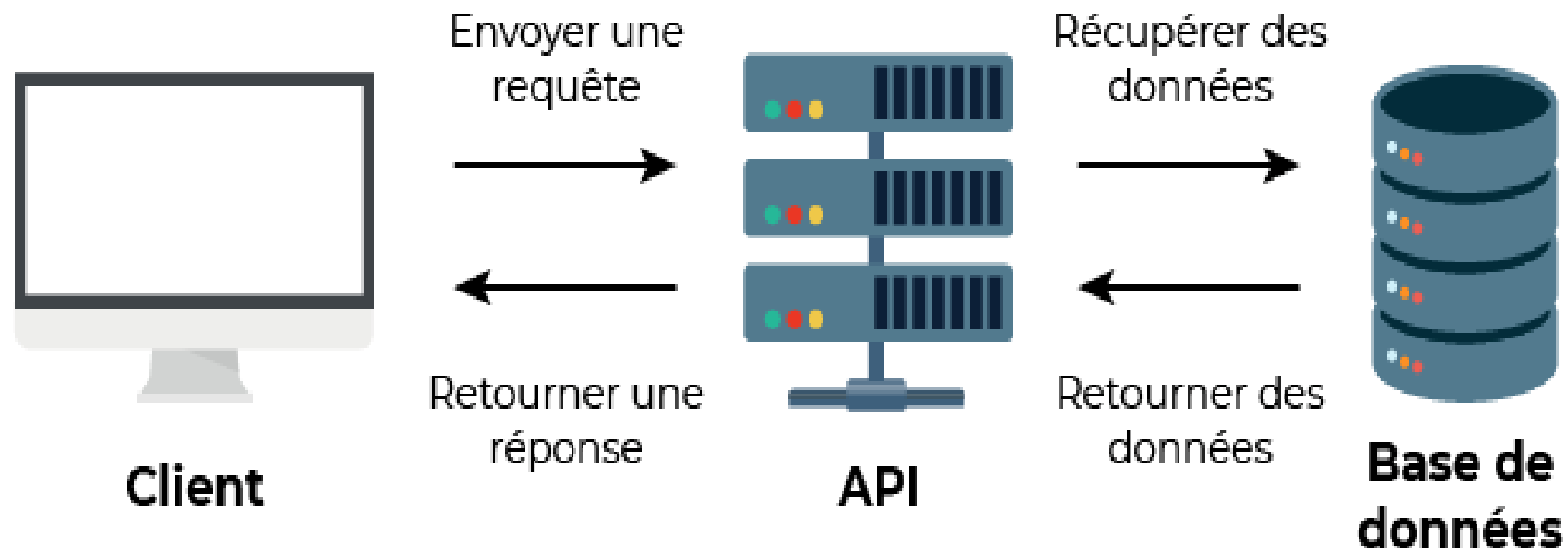
- Quels URI's sont disponibles ?
- Quelles méthodes HTTP peuvent être utilisées avec chaque URI ?
- Quels paramètres de requête elle accepte ?
- Quelles données peuvent être envoyées dans le corps de requête ?
- Ce que le consommateur peut Attendre comme une réponse ?

Pourquoi une API ?

Transformation digitale

ATAWAD : Any time, anywhere, any device





Une conversation entre le client, l'API et la base de données

Types d'API

Les API's Web peuvent être divisées en deux catégories :

- Remote Procedure Call (RPC)
- REpresentational State Transfer (REST)

Remote Procedure Call (RPC)

RPC est généralement caractérisé comme un seul URI sur lequel de nombreuses opérations peuvent être appelées, généralement uniquement via POST. Exemples incluent **XML-RPC** et **SOAP**. Habituellement, vous passerez une requête structurée qui inclut le nom de l'opération à invoquer et tous les arguments que vous souhaitez transmettre à l'opération, La réponse sera dans un format structuré.

POST /xml-rpc

POST /xml-rpc HTTP/1.1

Content-Type: text/xml

```
<?xml version="1.0" encoding="utf-8"?>
<methodCall>
  <methodName>status.create</methodName>
  <params>
    <param>
      <value><string>First post!</string></value>
    </param>
    <param>
      <value><string>user1</string></value>
    </param>
    <param>
      <value><dateTime.iso8601>20170313T14:22:21</dateTime.iso8601></value>
    </param>
  </params>
</methodCall>
```

Les points à retenir sur RPC

- Une seule Endpoint offre de nombreuses opérations.
- Généralement 2 méthodes HTTP (GET/POST).
- Formats de demande/réponse structuré et prévisible.
- Format de rapport d'erreur structuré et prévisible.
- Documentation structurée des opérations disponibles.

Le WEB n'aime pas RPC

- Vous ne pouvez pas déterminer via l'URI combien de ressources sont disponibles.
- Manque de cache HTTP, incapacité d'utiliser des verbes HTTP natifs pour les opérations courantes, L'absence de codes de réponse HTTP pour les rapports d'erreurs nécessite une analyse des résultats pour déterminer si une erreur s'est produite.

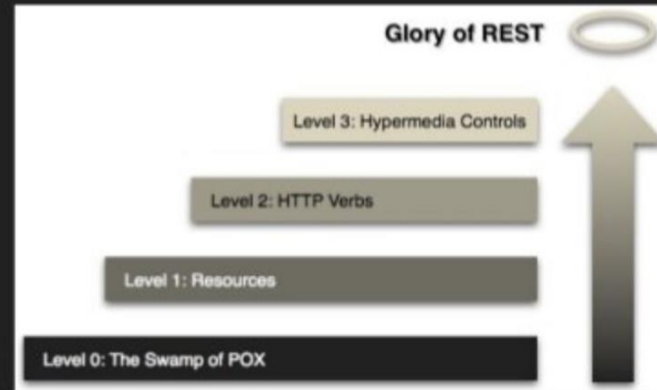
API REST

REpresentational State Transfer

Quand on parle du REST on utilise souvent Richardson Maturity Model pour décrire les règles à respecter lors de la mise en place d'une API REST.

RMM se compose de quatre niveaux :

- Level 0 : HTTP
- Level 1 : Resources
- Level 2 : Verbes et en-têtes HTTP
- Level 3 : Hypermedia Controls



<https://martinfowler.com/articles/richardsonMaturityModel.html>

Richardson Maturity Model

Level 0

Il faut HTTP comme mécanisme de transport.

Permet l'utilisation des URIs pour désigner des ressources uniques, des Verbes HTTP, possibilité de spécifier et de retourner plusieurs types de média, ou de lier des ressources.

Richardson Maturity Model

Level 1

Utilisation des URI (R = Resource) pour désigner les ressources individuelles comme des services.

`/users, /users/123,`

`/customers , /customers/456, /customers/456/orders`

Richardson Maturity Model

Level 2

Utilisation des verbes et en-têtes HTTP pour les interactions avec les ressources

- GET, POST, PUT, PATCH, DELETE, OPTIONS
- Accept, Content-Types, Authorization, If-Match , Access-Control-Allow-Methods ...

*GET /users HTTP/1.1
Content-Type: application/json*

Authorization Basic ZmFkZWw6e1NIQX0rQTZSZldqSXRxVmNtNFZiY0UyVHhYzY3RU09

Méthode HTTP PATCH

```
PATCH /posts/3c10c391-f56c-4d04-a889-bd1bd8f746f0 HTTP/1.1
```

```
Accept: application/json
```

```
Content-Type: application/json
```

```
{"message": "[Updated] First Post!"}
```

```
HTTP/1.1 403 Forbidden
```

```
Content-Type: application/problem+json
```

```
{  
  "detail": "Forbidden",  
  "status": 403,  
  "title": "Forbidden",  
  "type": "http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html"  
}
```

Richardson Maturity Model

Level 3

Hypermédia comme moteur d'application (HATEOAS : Hypertext As The Engine Of Application State), la ressource indique comment accéder aux autres ressources et Quelles sont les actions possibles sur la ressource.

Liée les ressources entre elles à travers des lien HAL

```
GET /status HTTP/1.1
Accept: application/json
-----
HTTP/1.1 200 OK
Content-Type: application/hal+json

{
  "_embedded": {
    "status": []
  },
  "_links": {
    "self": {
      "href": "http://localhost/posts"
    }
  },
  "page_count": 0,
  "page_size": 10,
  "total_items": 0
}
```

Autres concepts

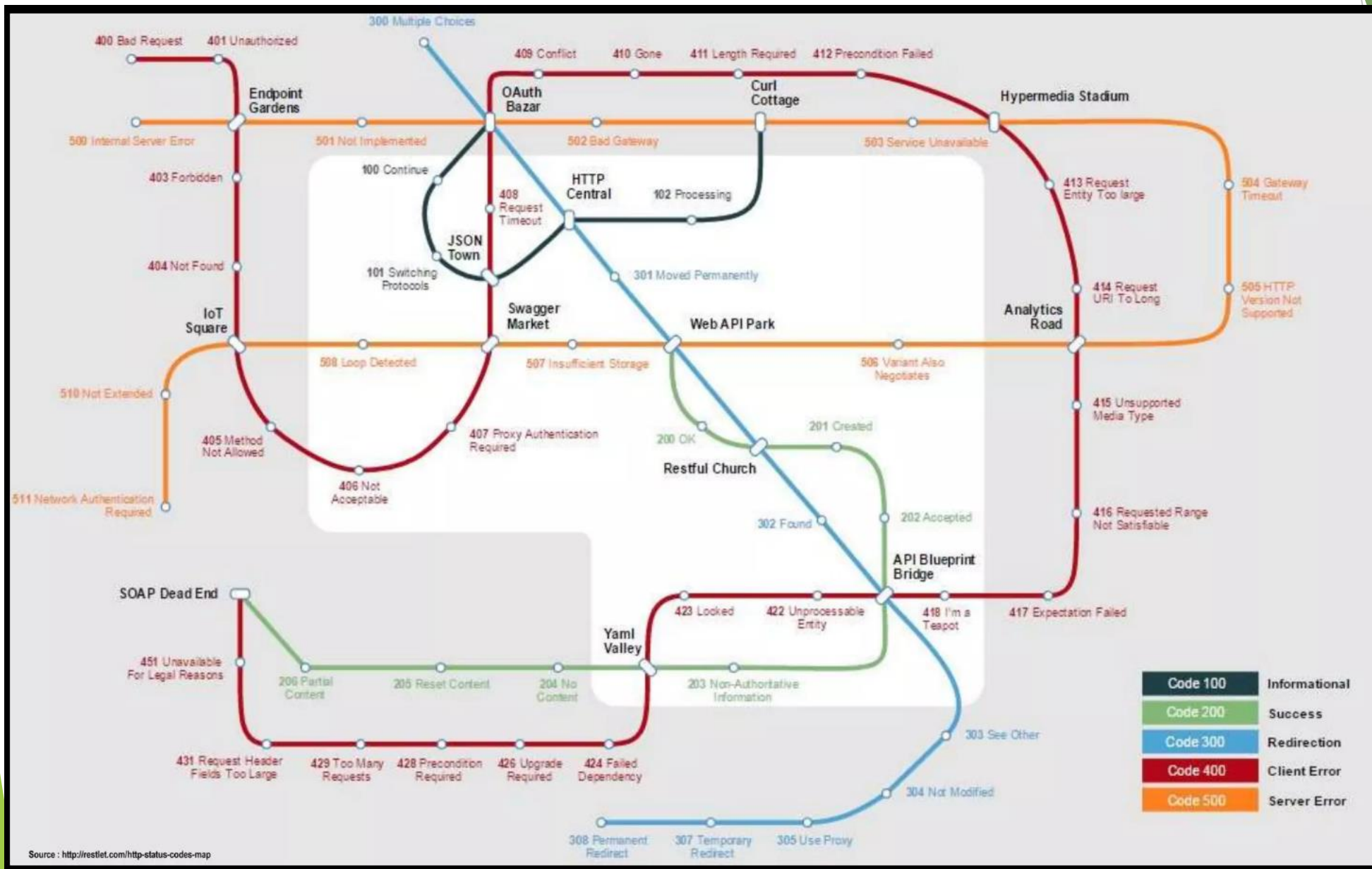
- HTTP Status Codes
- Négociation de contenu
- Cross-domain

HTTP Status Codes

- 200 – OK
- 201 Created
- 301 Moved Permanently
- 400 Bad Request
- 403 Forbidden
- 404 Not Found
- 500 Internal Server Error
- 503 Service Unavailable

<http://www.ietf.org/assignments/http-status-codes/http-status-codes.xml>

<http://www.restapitutorial.com/httpstatuscodes.html>



Négociation de contenu

Il est conseillé de gérer à minima 2 formats: JSON et XML, l'API pourra distribuer les ressources au format JSON, mais dans les cas où la requête spécifie en premier lieu « Accept: application/xml », les ressources seront fournis au format XML.

Client : *Accept: application/json*

API : *Content-type : application/json*

```
GET /status HTTP/1.1
Accept: application/json
-----
HTTP/1.1 200 OK
Content-Type: application/hal+json

{
  "_embedded": {
    "status": []
  },
  "_links": {
    "self": {
      "href": "http://localhost/posts"
    }
  },
  "page_count": 0,
  "page_size": 10,
  "total_items": 0
}
```

Cross-domain

Lorsqu'une application (JavaScript SPA) interroge une API sur un domaine différent, par exemple :

`https://myapp.com`

`https://api.myapp.com`

Une bonne pratique consiste à utiliser le protocole CORS qui est le standard HTTP.

La mise en oeuvre de CORS coté serveur consiste à ajouter quelques directives sur les serveurs HTTP (Nginx/Apache/Nodejs...).

Coté client le navigateur effectuera avant chaque requête GET/POST/PUT/PATCH/DELETE une requêtes HTTP avec le verbe OPTIONS.

```
OPTIONS /resources/post-here/ HTTP/1.1
Host: bar.other
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1b3pre) Gecko/2001
Accept: application/json,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Origin: http://foo.example
Access-Control-Request-Method: POST
Access-Control-Request-Headers: X-PINGOTHER, Content-Type
```

```
HTTP/1.1 200 OK
Date: Mon, 01 Dec 2008 01:15:39 GMT
Server: Apache/2.0.61 (Unix)
Access-Control-Allow-Origin: http://foo.example
Access-Control-Allow-Methods: POST, GET, OPTIONS
Access-Control-Allow-Headers: X-PINGOTHER, Content-Type
Access-Control-Max-Age: 86400
Vary: Accept-Encoding, Origin
Content-Encoding: gzip
Content-Length: 0
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive
Content-Type: application/json
```

Sécuriser une API REST

- SSL
- OAuth2
- JWT



REST & RESTFuI

- REST est un style architecture
- RESTFUL c'est une application "full of REST"

Les Géants du Web ?

API	REST	SOAP	Comments
Amazon S3	X	X	« SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs. »
Amazon EC2	X	-	
Facebook	X	-	
Google cloud, maps, apps, plus, youtube, etc.	X	-	
Twitter	X	-	
Paypal	X	X	
Instagram	X	-	
Pinterest	X	-	
LinkedIn	X	-	
TripAdvisor	X	-	
Expedia Affiliate Network	X	X	« EAN has discontinued support for SOAP. See our SOAP to REST migration guide for details on changes required for affected integrations. »



Bibliographie:

- ▶ <https://www.twilio.com/docs/voice/tutorials/automated-survey-java-spring>
- ▶ <https://openclassrooms.com/fr/courses/6573181-adoptez-les-api-rest-pour-vos-projets-web/6820251-entraenez-vous-avec-l-api-github>
- ▶ <https://fr.slideshare.net/fadex/api-rest-73086803>