

Programmation Web – Avancé

JavaScript & Node.js

Consommation de RESTful APIs avec REST
Client



Attribution –
Partage dans les
Mêmes Conditions
4.0 International
(CC BY-SA 4.0)

*Presentation template
by [SlidesCarnival](#)*



Tests de RESTful APIs

Comment pourrions nous appeler nos RESTful APIs sans développer de frontend ?



Consommer des web APIs rapidement

- Installation du REST Client [\[77.\]](#) dans VS Code
- Création de fichiers **.http**
- Ajout des requêtes vers vos web APIs

```
### POST /user/login for a user that does not exist  
Send Request  
POST http://localhost:3000/api/users/login
```

```
1 HTTP/1.1 401 Unauthorized  
2 X-Powered-By: Express  
3 Content-Type: text/html; charset=utf-8  
4 Content-Length: 18  
5 ETag: W/"12-1McGD5NQMPAKQrUcKcYA2woAoEw"  
6 Date: Thu, 30 Jul 2020 10:05:49 GMT  
7 Connection: close  
8  
9 bad email/password
```



Consommer des web APIs rapidement

```
### POST users : john@here.com / john
```

Send Request

```
POST http://localhost:3000/api/users/
```

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

```
{
  "email": "john@here.com",
  "password": "John"
}
```

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 28
5 ETag: W/"1c-46ddGT2wnHx0IFplr3gPHnL2CZc"
6 Set-Cookie: connect.sid=s%3A18YhgPtEL4R74B7ITUDEXMPK159
  KM-QA.mK%2BlZqKwvEFOIMPObMxiWc7a6n1d3eHpUP%2B%2FrnlVdh
  E; Path=/; HttpOnly
7 Date: Thu, 30 Jul 2020 10:28:17 GMT
8 Connection: close
9
10 {
11   "username": "john@here.com"
12 }
```



Consommer des web APIs rapidement

```
### POST/users/login for an existant user
Send Request
POST http://localhost:3000/api/users/login
Content-Type: application/x-www-form-urlencoded

email=john@here.com
&password=John

1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 28
5 ETag: W/"1c-46ddGT2wnHx0IFplr3gPHnL2CZc"
6 Date: Thu, 30 Jul 2020 10:35:18 GMT
7 Connection: close
8
9 {
10   "username": "john@here.com"
11 }
```



Consommer des web APIs rapidement

- REST Client :
 - Séparation des requêtes par ### (ou plus)
 - Exécution d'une requête :



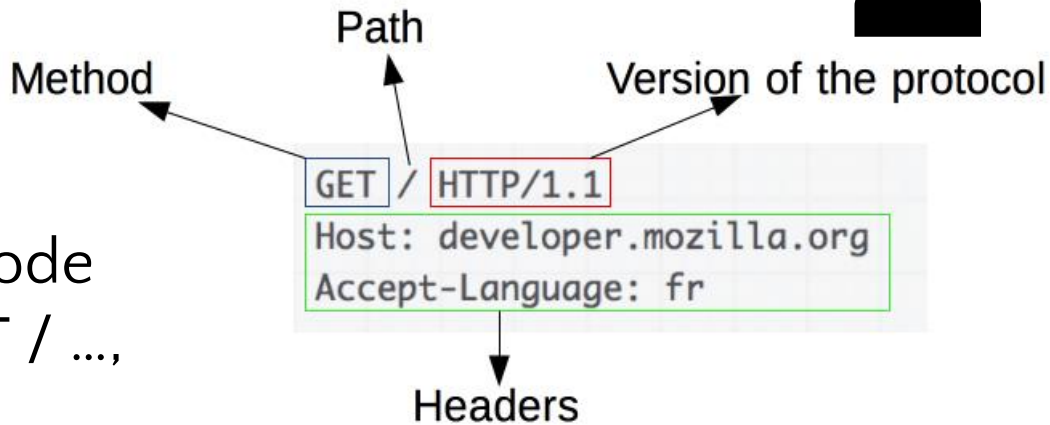
```
Send Request  
POST http://localhost:3000/api/users/login
```

- Existence de beaucoup d'autres options pour tester vos APIs, notamment Postman [\[78.\]](#)



Requêtes HTTP

- Start line (Méthode HTTP GET / PUT / ..., URL)
- Headers (host, cookies...)
- Body (HTML form parameters, JSON...)



Basics of HTTP [53.]



Réponses HTTP

- Status line (status code & text...)
- Header (content type, Location, cookies...)
- Body (ressource demandée, message d'erreur, JSON...)

Status code
Version of the protocol
Status message



HTTP/1.1

200

OK

```
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html
```

Headers

Basics of HTTP [53.]



GET : lecture de ressource(s)

```
### Read all pizzas
GET http://localhost:3000/pizzas

### Read pizza identified by 2
GET http://localhost:3000/pizzas/2
```

- Paramètres dans l'URL
- Mise en cache possible



POST : création d'une ressource

```
### Create a pizza
POST http://localhost:3000/pizzas
Content-Type: application/json

{
  "title": "Magic Green",
  "content": "Epinards, Brocolis, Olives vertes, Basilic"
}
```

- Paramètres dans le body de la requête
- Généralement non mis en cache



PUT : modification de ressource(s)

```
### Update the pizza identified by 2
PUT http://localhost:3000/pizzas/2
Content-Type: application/json

{
  "title": "Magic Green 2"
}
```

- Param. à mettre à jour = body de la requête
- ID de la ressource à mettre à jour dans l'URL



DELETE : suppression de ressource(s)

```
### Delete pizza identified by 2  
DELETE http://localhost:3000/pizzas/2
```

- 🕒 ID de la ressource à supprimer dans l'URL



Consommation de RESTful API avec REST Client

- DEMO : Création d'une RESTfull API pour une pizzeria : Part 1 – CRUD pizzas : test des opérations sur les pizzas