



UNIVERSITÉ  
DE LORRAINE

Réalisé Pour la **PDIE**



LORRAINE  
**INP** Ensem

---

# Présentation PTI Covélotage

---

**Membres du jury:**

A. Lahmadi

R. Ranta

S. Le Cam

**BEN RAIES Sabri**

**DUBOIS Christophe**

Soutenu le 15 Février 2024

# Plan

1 Présentation de l'organisme d' accueil

2 Cadre du projet et problématique

3 Partie 1: Recherche et développement

4 Partie 2: Conception et développement du Site

5 Démonstration

6 Conclusions et perspectives



# Présentation de l'organisme d'accueil

# Présentation de l'organisme d'accueil

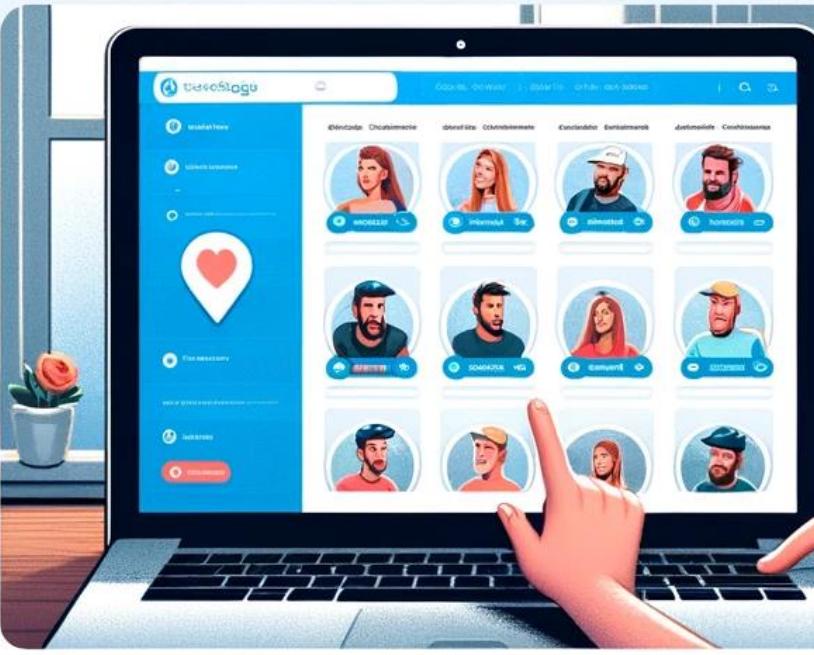
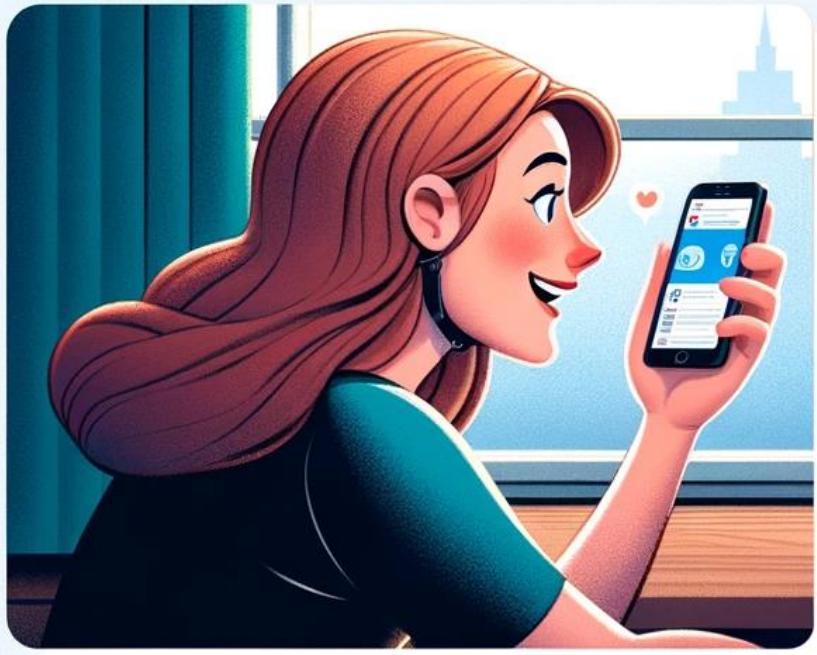


- l'association des entreprises et des établissements du Technopôle de Nancy – Brabois
- Plan de Déplacement Inter Entreprises
- Lancé en septembre 2011

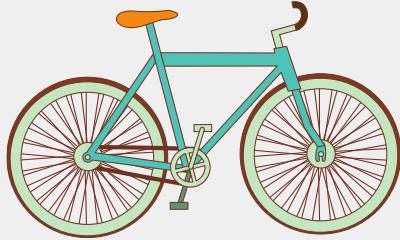




# Cadre du projet et problématique



# Cadre du projet et problématique



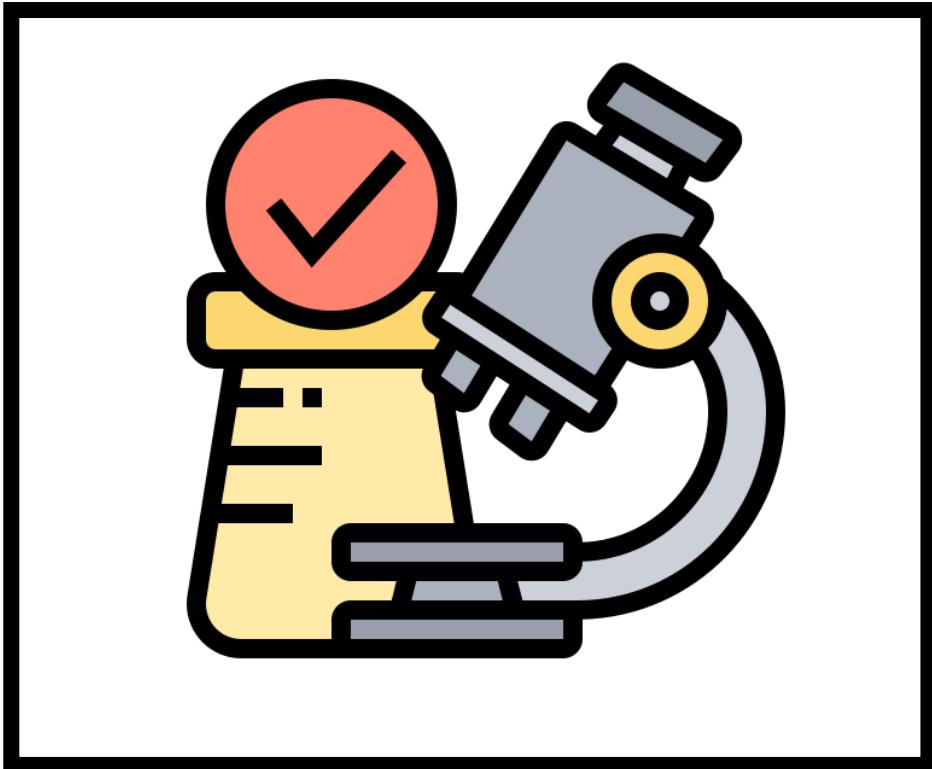
## Covélotage

- Mettre en relation des personnes effectuant les mêmes itinéraires

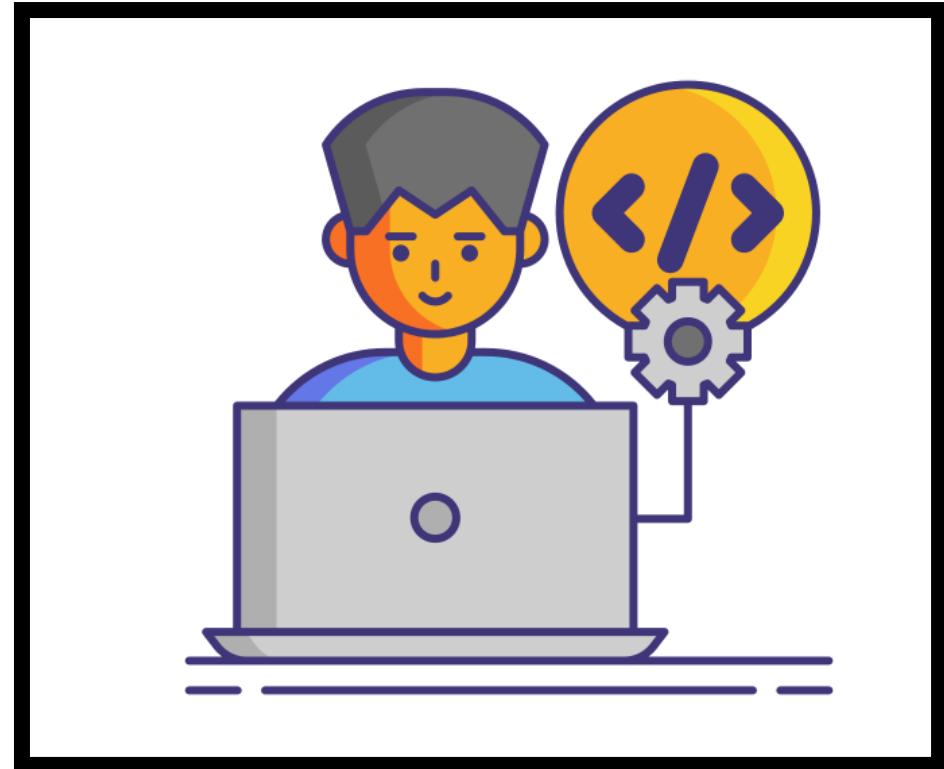


- Proposer des challenges liés à la mobilité
- Services d'information sur le parcours (garages à vélo protégé, outils en libre-service, ...)

# Phases du projet



Partie 1:  
Recherche et Développement:



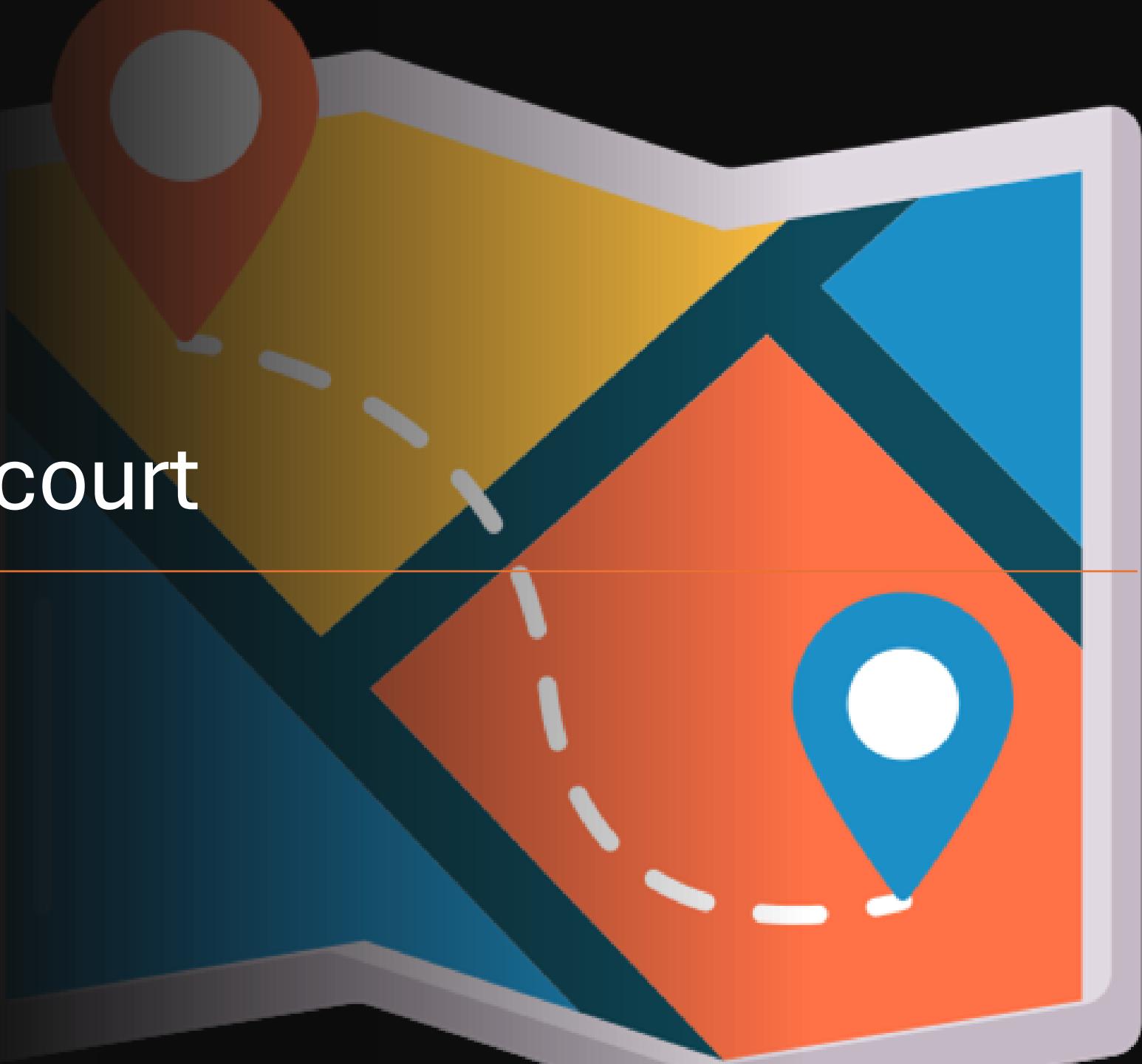
Partie 2:  
Conception et  
Développement du Site



# Recherche et Développement:

# Algorithme de recherche du chemin le plus court

---



## Architecture

Choix ORS

Configuration ORS

Analyse Réponse ORS

FrontEnd

BackEnd



FrontEnd



BackEnd



Architecture

Choix ORS

Configuration ORS

Analyse Réponse ORS

FrontEnd

BackEnd

Interaction BackEnd  
FrontEnd

# openroute service



# openroute service



Architecture

Choix ORS

Configuration ORS

Analyse Réponse ORS

FrontEnd

BackEnd

Interaction BackEnd  
FrontEnd

Architecture  
Choix ORS

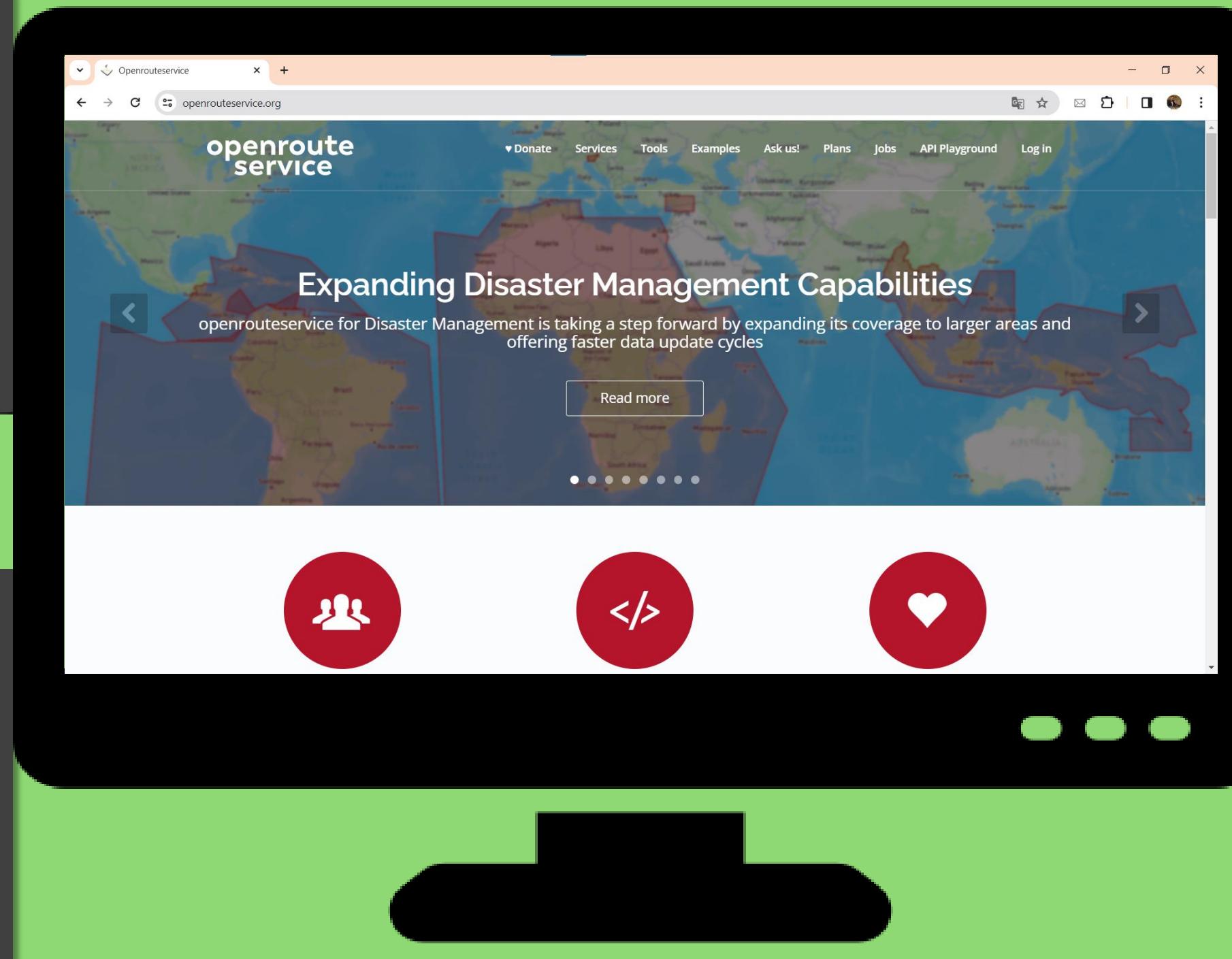
Configuration  
ORS

Analyse Réponse ORS

FrontEnd

BackEnd

Interaction BackEnd  
FrontEnd



Architecture

Choix ORS

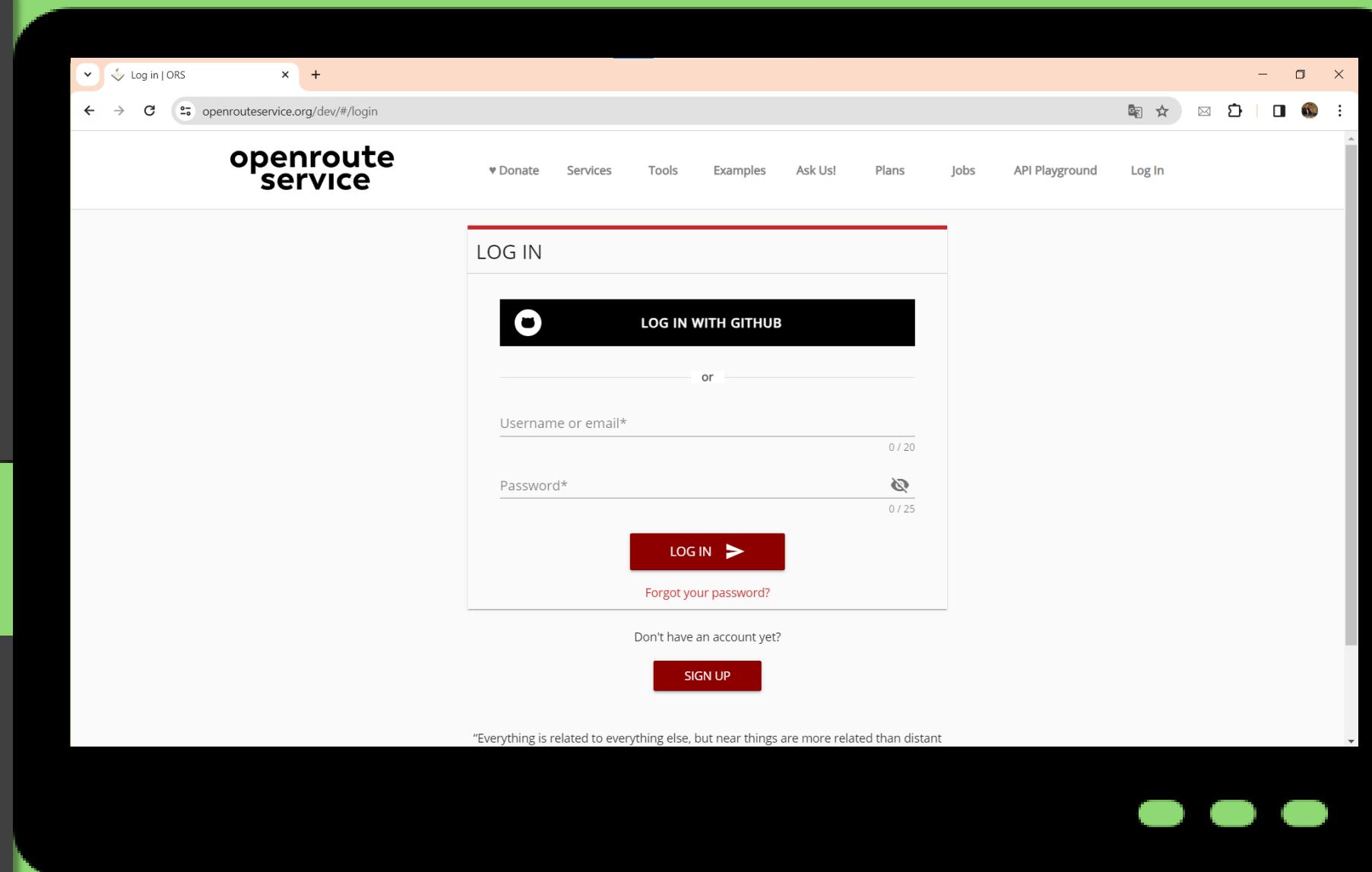
Configuration  
ORS

Analyse Réponse ORS

FrontEnd

BackEnd

Interaction BackEnd  
FrontEnd



# Architecture

## Choix ORS

# Configuration ORS

## Analyse Réponse ORS

### FrontEnd

### BackEnd

## Interaction BackEnd FrontEnd

The screenshot shows the OpenRouteService Dev dashboard. At the top, there is a navigation bar with links: 'Donate', 'Services', 'Tools', 'Examples', 'Ask Us!', 'Plans', 'Jobs', 'API Playground', 'Dashboard' (which is underlined), and a user profile 'Hi Sabri'. Below the navigation bar, the title 'Dev dashboard' is displayed. Underneath, there are two tabs: 'TOKENS' (which is selected) and 'PROFILE'. The 'TOKENS' tab displays a table with columns: 'Name', 'Key', 'Is valid', 'Remaining Quota', and 'Actions'. A row for 'sabri PTI' is highlighted with a red border. The 'Actions' column for this row contains icons for edit, copy, refresh, and delete. Below the tokens section, there is a heading 'Token quota' followed by a table showing usage statistics for various endpoints.

Main Endpoints	Quota left (renews in)	Per Min	Microservice Endpoints	Quota left (renews in)	Per Min
Directions V2	2000/2000	40	ElevationLine	200/200	40
Isochrones V2	500/500	20	ElevationPoint	2000/2000	100

Architecture

Choix ORS

Configuration ORS

Analyse

Réponse ORS

FrontEnd

BackEnd

Interaction BackEnd  
FrontEnd

```
[  
  "type": "FeatureCollection",  
  "metadata": { ...  
  },  
  "bbox": [ ...  
  ],  
  "features": [ ...  
  ]  
]
```

Architecture  
Choix ORS  
Configuration ORS

# Analyse

## Réponse ORS

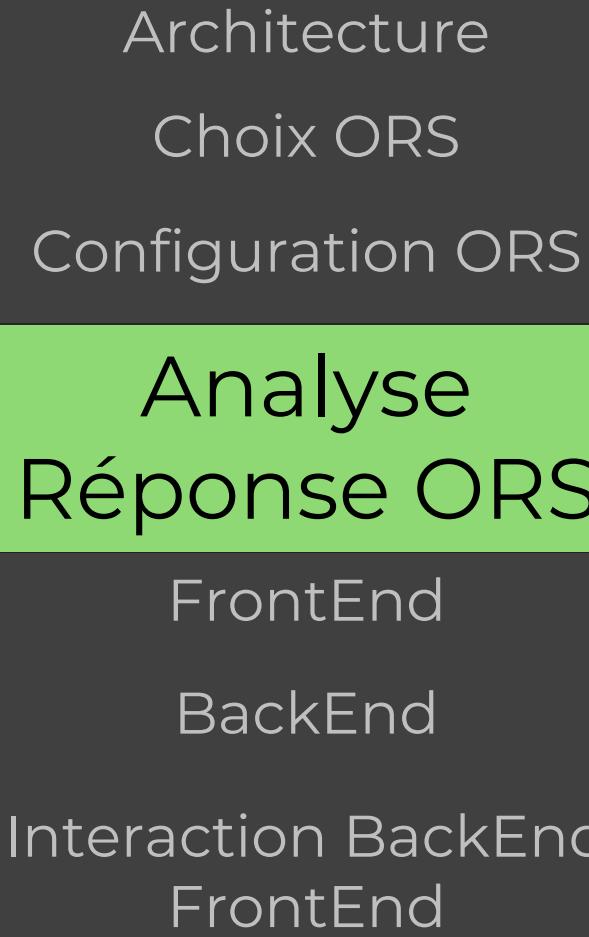
FrontEnd  
BackEnd

Interaction BackEnd  
FrontEnd

```
{  
    "type": "FeatureCollection",  
    "metadata": {  
        "attribution": "openrouteservice.org | OpenStreetMap contributors",  
        "service": "routing",  
        "timestamp": 1702309857447,  
        "query": {  
            "coordinates": [  
                [ 6.183264, 48.693684 ],  
                [ 6.172724, 48.667496 ]  
            ],  
            "profile": "cycling-regular",  
            "format": "geojson"  
        },  
        "engine": {  
            "version": "7.1.0",  
            "build_date": "2023-12-10T05:30:50Z",  
            "graph_date": "2023-12-03T23:16:49Z"  
        }  
    },  
    "bbox": [ ... ],  
    "features": [ ... ]  
}
```

Architecture  
Choix ORS  
Configuration ORS  
**Analyse**  
**Réponse ORS**  
FrontEnd  
BackEnd  
Interaction BackEnd  
FrontEnd

```
{  
  "type": "FeatureCollection",  
  "metadata": { ...  
  },  
  "bbox": [  
    6.165847,  
    48.667494,  
    6.183232,  
    48.693944  
  ],  
  "features": [ ...  
  ]  
}
```



```
< {
    "type": "FeatureCollection",
    "metadata": { ... },
    "bbox": [ ... ],
    "features": [
        {
            "bbox": [ ... ],
            "type": "Feature",
            "properties": {
                "transfers": 0,
                "fare": 0,
                "segments": [
                    {
                        "distance": 4254.1,
                        "duration": 910.6,
                        "steps": [
                            {
                                "distance": 6.0,
                                "duration": 3.6,
                                "type": 11,
                                "instruction": "Head southwest",
                                "name": "-",
                                "way_points": [
                                    0,
                                    1
                                ]
                            },
                            {
                                "distance": 28.9,
                                "duration": 17.4,
                                "type": 1,
                                "instruction": "Turn right",
                                "name": "-",
                                "way_points": [

```

Architecture  
Choix ORS  
Configuration ORS  
  
Analyse  
Réponse ORS  
  
FrontEnd  
BackEnd  
  
Interaction BackEnd  
FrontEnd

```
<pre>
    "type": "FeatureCollection",
    "metadata": { ... },
    "bbox": [ ... ],
    "features": [
        {
            "bbox": [ ... ],
            "type": "Feature",
            "properties": { ... },
            "geometry": {
                "coordinates": [
                    [
                        [
                            6.183232,
                            48.69373
                        ],
                        [
                            6.183158,
                            48.693709
                        ],
                        [
                            6.182992,
                            48.693944
                        ],
                        [
                            6.182876,
                            48.693909
                        ],
                        [
                            6.182399,
                            48.693767
                        ],
                        [
                            6.182409,
                            48.693742
                        ]
                    ]
                ]
            }
        }
    ]
}</pre>
```

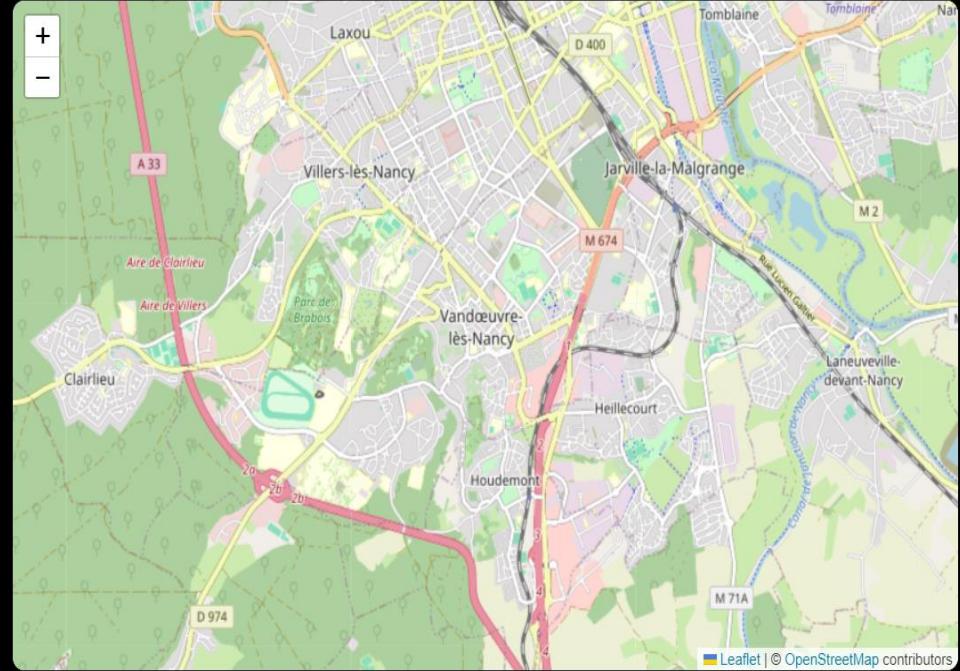
Architecture  
Choix ORS  
Configuration ORS  
Analyse Réponse ORS

## FrontEnd

## BackEnd

Interaction BackEnd  
FrontEnd

Start Latitude
<input type="text"/>
Start Longitude
<input type="text"/>
End Latitude
<input type="text"/>
End Longitude
<input type="text"/>



Architecture  
Choix ORS  
Configuration ORS  
Analyse Réponse ORS

## FrontEnd

BackEnd

Interaction BackEnd  
FrontEnd



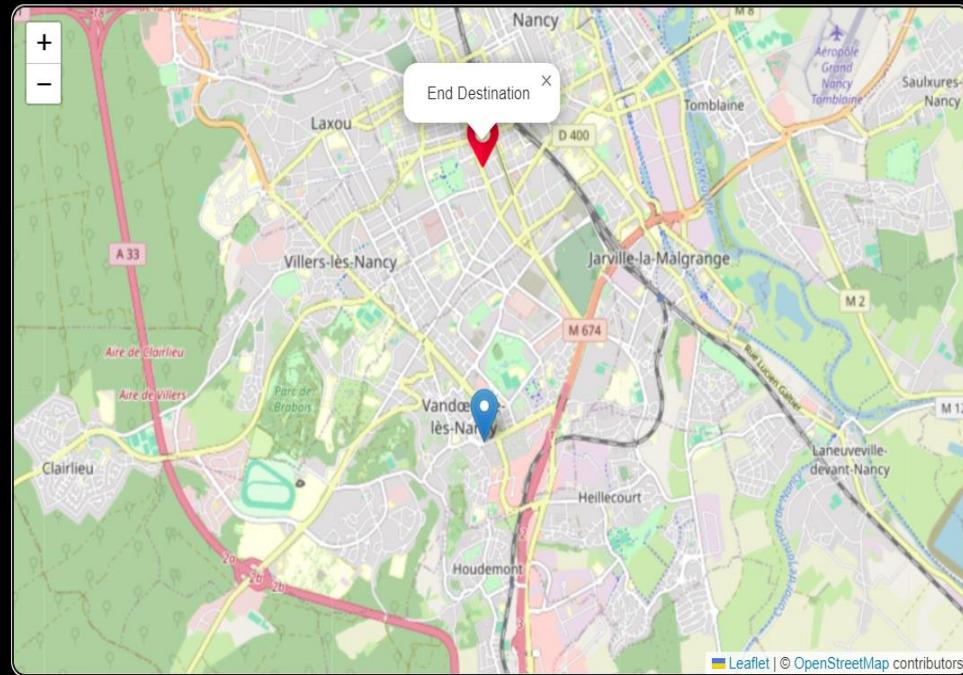
Architecture  
Choix ORS  
Configuration ORS  
Analyse Réponse ORS

## FrontEnd

BackEnd

Interaction BackEnd  
FrontEnd

48.657634129568045  
6.175518035888673  
48.683367570119955  
6.175346374511719



Architecture

Choix ORS

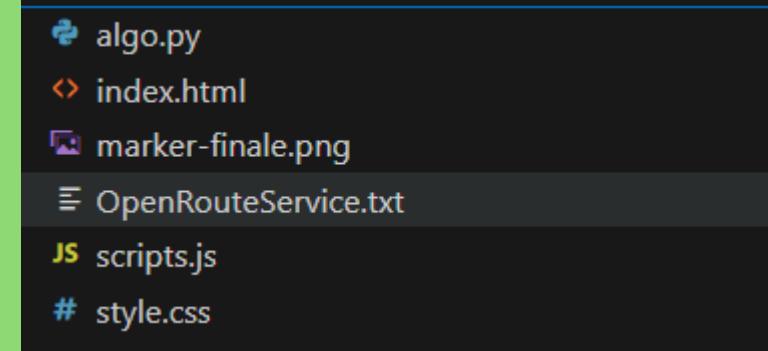
Configuration ORS

Analyse Réponse ORS

## FrontEnd

BackEnd

Interaction BackEnd  
FrontEnd

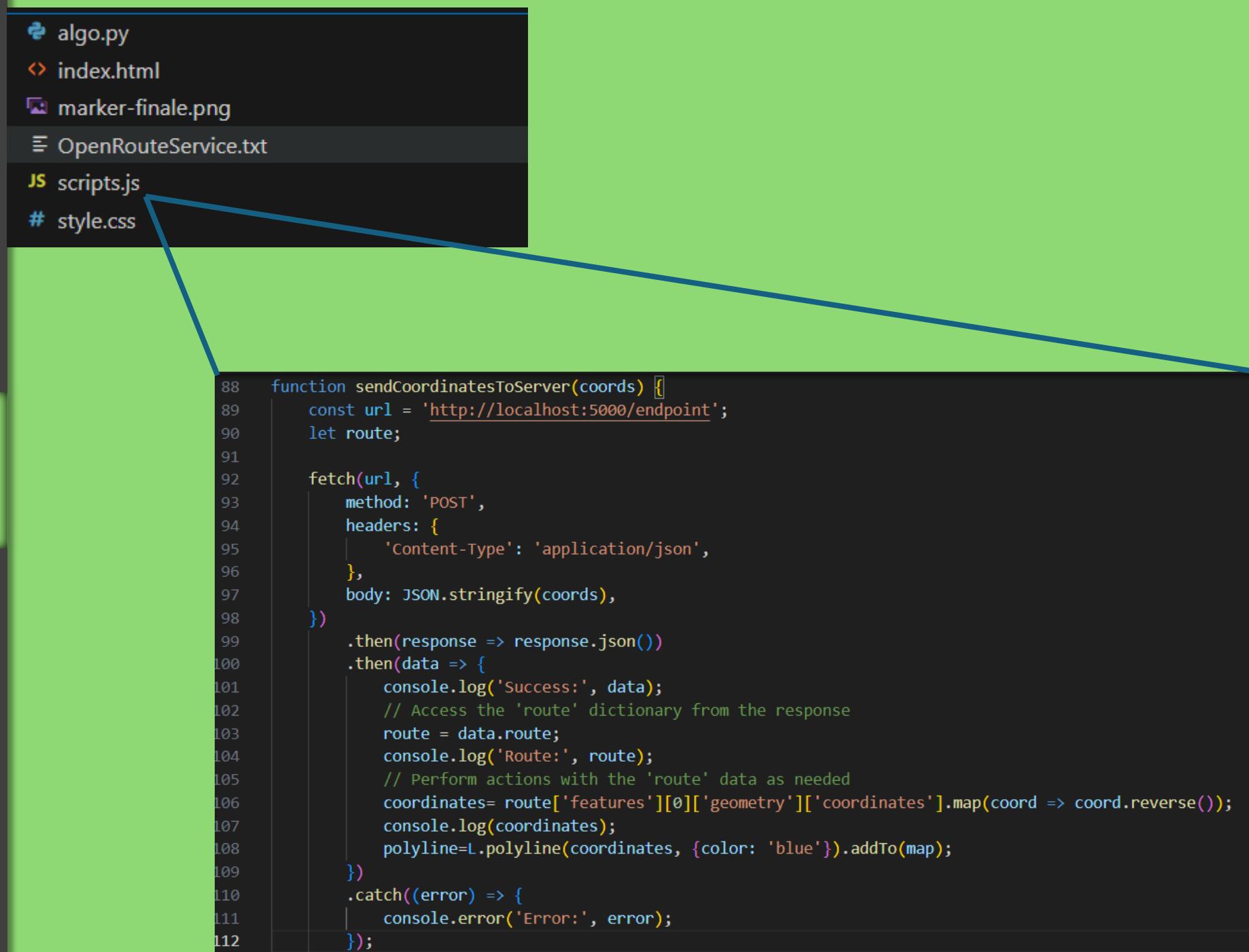


Architecture  
Choix ORS  
Configuration ORS  
Analyse Réponse ORS

## FrontEnd

## BackEnd

Interaction BackEnd  
FrontEnd



```
88 function sendCoordinatesToServer(coords) {
89   const url = 'http://localhost:5000/endpoint';
90   let route;
91
92   fetch(url, {
93     method: 'POST',
94     headers: {
95       'Content-Type': 'application/json',
96     },
97     body: JSON.stringify(coords),
98   })
99     .then(response => response.json())
100    .then(data => {
101      console.log('Success:', data);
102      // Access the 'route' dictionary from the response
103      route = data.route;
104      console.log('Route:', route);
105      // Perform actions with the 'route' data as needed
106      coordinates= route['features'][0]['geometry']['coordinates'].map(coord => coord.reverse());
107      console.log(coordinates);
108      polyline=L.polyline(coordinates, {color: 'blue'}).addTo(map);
109    })
110    .catch(error) => {
111      console.error('Error:', error);
112    });
}
```

Architecture

Choix ORS

Configuration ORS

Analyse Réponse ORS

FrontEnd

## BackEnd

Interaction BackEnd

FrontEnd

```
8     app = Flask(__name__)
9     CORS(app)
10
11    @app.route('/endpoint', methods=['POST'])
12    def receive_coordinates():
13        data = request.get_json()
14
15        # Access the coordinates from the JSON data
16        start_latitude = data.get('start_latitude')
17        start_longitude = data.get('start_longitude')
18        end_latitude = data.get('end_latitude')
19        end_longitude = data.get('end_longitude')
20
21        # Process the coordinates as needed
22        print(f'Start Latitude: {start_latitude}')
23        print(f'End Latitude: {end_latitude}')
24        print(f'Start Longitude: {start_longitude}')
25        print(f'End Longitude: {end_longitude}')
```

Architecture  
Choix ORS  
Configuration ORS  
Analyse Réponse ORS  
FrontEnd

## BackEnd

Interaction BackEnd  
FrontEnd

```
11 @app.route('/endpoint', methods=[ POST ])
12 def receive_coordinates():
13     data = request.get_json()
14
15     # Access the coordinates from the JSON data
16     start_latitude = data.get('start_latitude')
17     start_longitude = data.get('start_longitude')
18     end_latitude = data.get('end_latitude')
19     end_longitude = data.get('end_longitude')
20
21     # Process the coordinates as needed
22     print(f'Start Latitude: {start_latitude}')
23     print(f'End Latitude: {end_latitude}')
24     print(f'Start Longitude: {start_longitude}')
25     print(f'End Longitude: {end_longitude}')
26
27     # Perform operations and generate the 'route' dictionary
28
29
30 #####API=open("OpenRouteService.txt","r")
31 APIKey=API.read()
32 client=ors.Client(key=APIKey)
33 route={}
34 coords = [[start_longitude,start_latitude], [end_longitude,end_latitude]]
35 print(coords)
36 route = client.directions(coordinates=coords,
37 | | | | profile='cycling-regular',
38 | | | | format='geojson')
39
40 print(len(route))
41
42 #####
43
44 #send a response back to the JavaScript code with the 'route' dictionary
45 return jsonify({'message': 'Coordinates received successfully', 'route': route})
```

Architecture  
Choix ORS

Configuration ORS

Analyse Réponse ORS

FrontEnd  
BackEnd

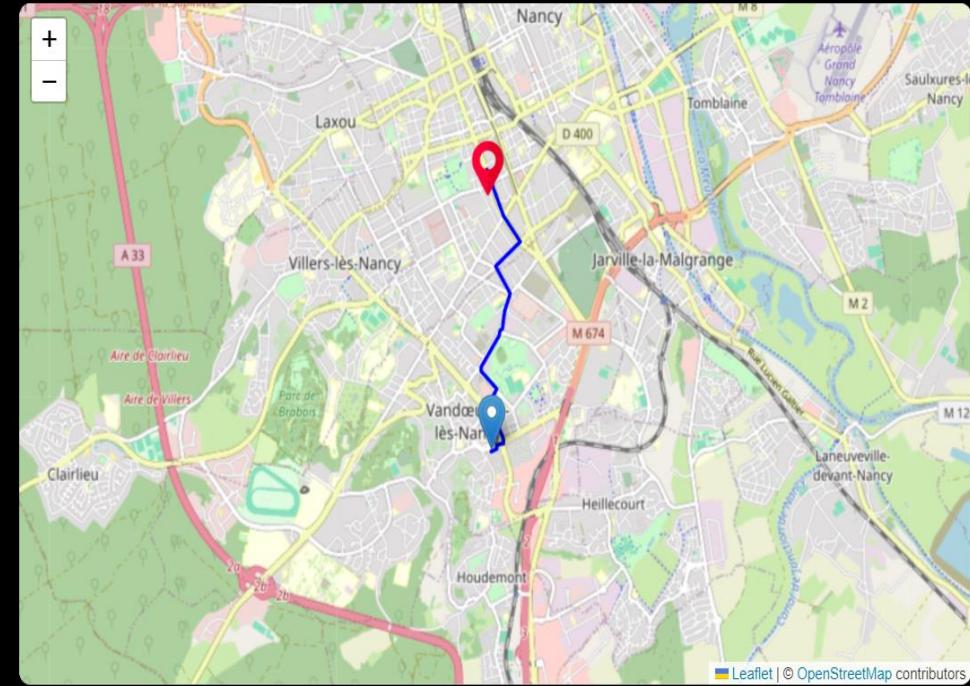
Interaction BackEnd  
FrontEnd

48.65752073731804

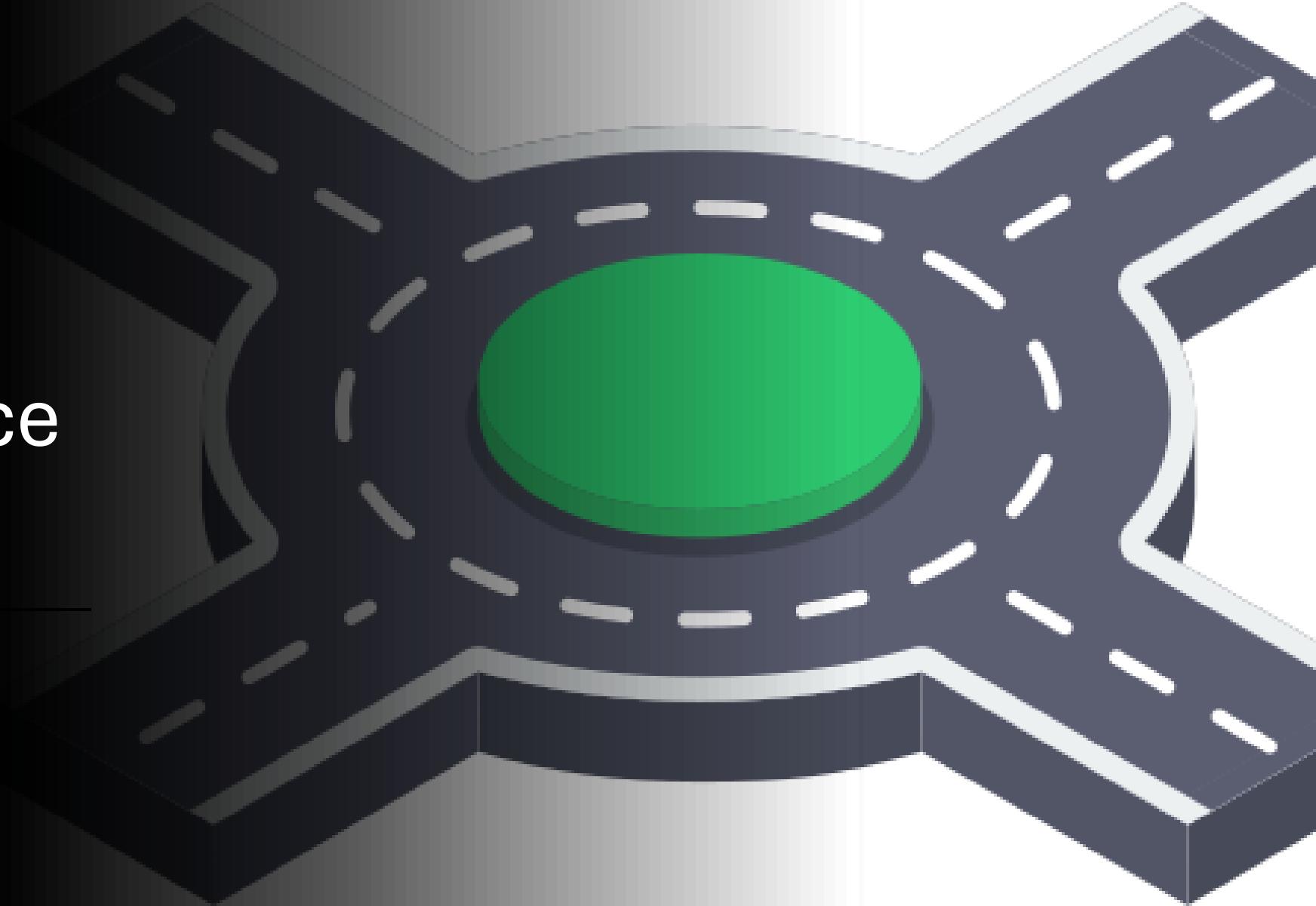
6.176033020019532

48.68098749511624

6.175518035888673



# Algorithme de correspondance de chemin



# Algorithme LCS



Séquence A: BDCABA

Séquence B: ABCBDAB

Séquence A: **BDCABA**

Séquence B: ABCB**DAB**

Séquence A: BDCABA

Séquence B: ABCBDAB

BDCABA

ABCBDA

	B	D	C	A	B	A
A						
B						
C						
B						
D						
A						
B						

	B	D	C	A	B	A
A						
B						
C						
B						
D						
A						
B						



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	



Plus grand  
entre b et c

B D C A B A

	0	0	0	0	0	0	0
A	0						
B	0						
C	0						
B	0						
D	0						
A	0						
B	0						



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c

	B	D	C	A	B	A
A	0	0	0	0	0	0
B	0					
C	0					
B	0					
D	0					
A	0					
B	0					



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c

	B	D	C	A	B	A
A	0	0	0	0	0	0
B	0					
C	0					
B	0					
D	0					
A	0					
B	0					



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	



Plus grand  
entre b et c

	B	D	C	A	B	A
A	0	0	0	0	0	0
B	0					
C	0					
B	0					
D	0					
A	0					
B	0					



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	



Plus grand  
entre b et c

B D C A B A

0	0	0	0	0	0	0	0
A	0	0	0	0	1		
B	0						
C	0						
B	0						
D	0						
A	0						
B	0						



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c

B D C A B A

0	0	0	0	0	0	0	0
A	0	0	0	0	1	1	
B	0						
C	0						
B	0						
D	0						
A	0						
B	0						



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c

	B	D	C	A	B	A
A	0	0	0	0	0	0
B	0					
C	0					
B	0					
D	0					
A	0					
B	0					



1



1



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	



Plus grand  
entre b et c

B D C A B A

0	0	0	0	0	0	0
A	0	0	0	0	1	1
B	0	1				
C	0					
B	0					
D	0					
A	0					
B	0					



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c

B D C A B A

0	0	0	0	0	0	0	0
A	0	0	0	0	1	1	1
B	0	1	1	1	1	2	2
C	0						
B	0						
D	0						
A	0						
B	0						



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c



Règles à suivre

Correspondance

a	
	<b>a+1</b>

Non Correspondance

	c
b	<b>b</b>

Plus grand  
entre b et c

	B	D	C	A	B	A
A	0	0	0	0	0	0
B	0	0	0	0	1	1
C	0	1	1	1	1	2
D	0	1	1	2	2	2
A	0	1	1	2	2	2
B	0	1	1	2	3	3
C	0	1	1	2	2	3
D	0	1	2	2	3	3
A	0	1	2	2	3	3
B	0	1	2	2	3	4



Règles à suivre

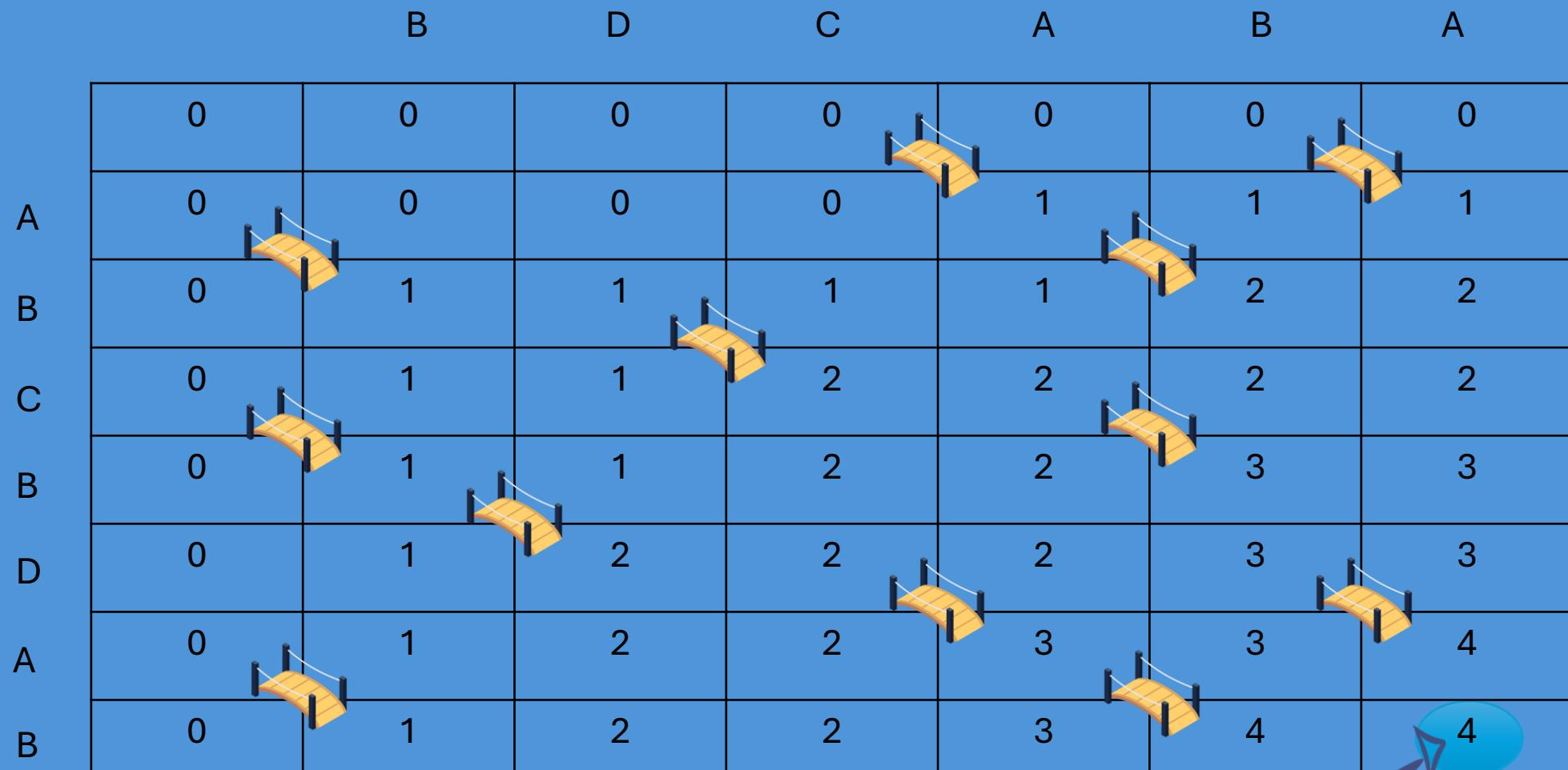
Correspondance

a	
	<b>a+1</b>

Non Correspondance

	c
b	<b>b</b>

Plus grand  
entre b et c



Longueur de la plus grande sous  
séquence commune



Règles à suivre

Correspondance

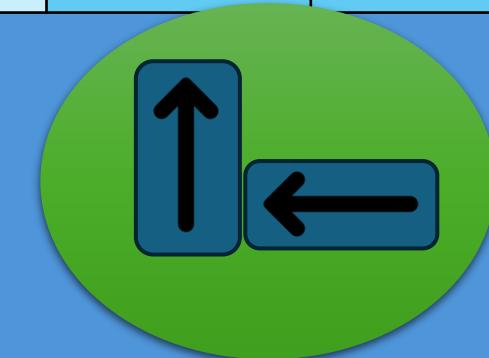
a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c

	B	D	C	A	B	A	
A	0	0	0	0	0	0	0
B	0	0	0	0	1	1	1
C	0	1	1	1	1	2	2
D	0	1	1	2	2	2	2
B	0	1	1	2	2	3	3
D	0	1	2	2	2	3	3
A	0	1	2	2	3	3	4
B	0	1	2	2	3	4	4





Règles à suivre

Correspondance

a	
	$a+1$

Non Correspondance

	c
b	

Plus grand  
entre b et c

	B	D	C	A	B	A	
A	0	0	0	0	0	0	
B	0	1	1	1	1	2	2
C	0	1	1	2	2	2	2
B	0	1	1	2	2	3	3
D	0	1	2	2	2	3	3
A	0	1	2	2	3	3	4
B	0	1	2	2	3	4	4



Règles à suivre

Correspondance

a	
	$a+1$

Non Correspondance

	c
b	

Plus grand  
entre b et c

	B	D	C	A	B	A
A	0	0	0	0	0	0
B	0	0	0	0	1	1
C	0	1	1	1	1	2
D	0	1	1	2	2	2
A	0	1	1	2	2	3
B	0	1	1	2	2	3
C	0	1	2	2	3	3
D	0	1	2	2	3	3
A	0	1	2	2	3	4
B	0	1	2	2	3	4



Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c

	B	D	C	A	B	A
A	0	0	0	0	0	0
B	0	0	0	0	1	1
C	0	1	1	1	1	2
D	0	1	1	2	2	2
B	0	1	1	2	2	3
D	0	1	2	2	2	3
A	0	1	2	2	3	3
B	0	1	2	2	3	4

B



Règles à suivre

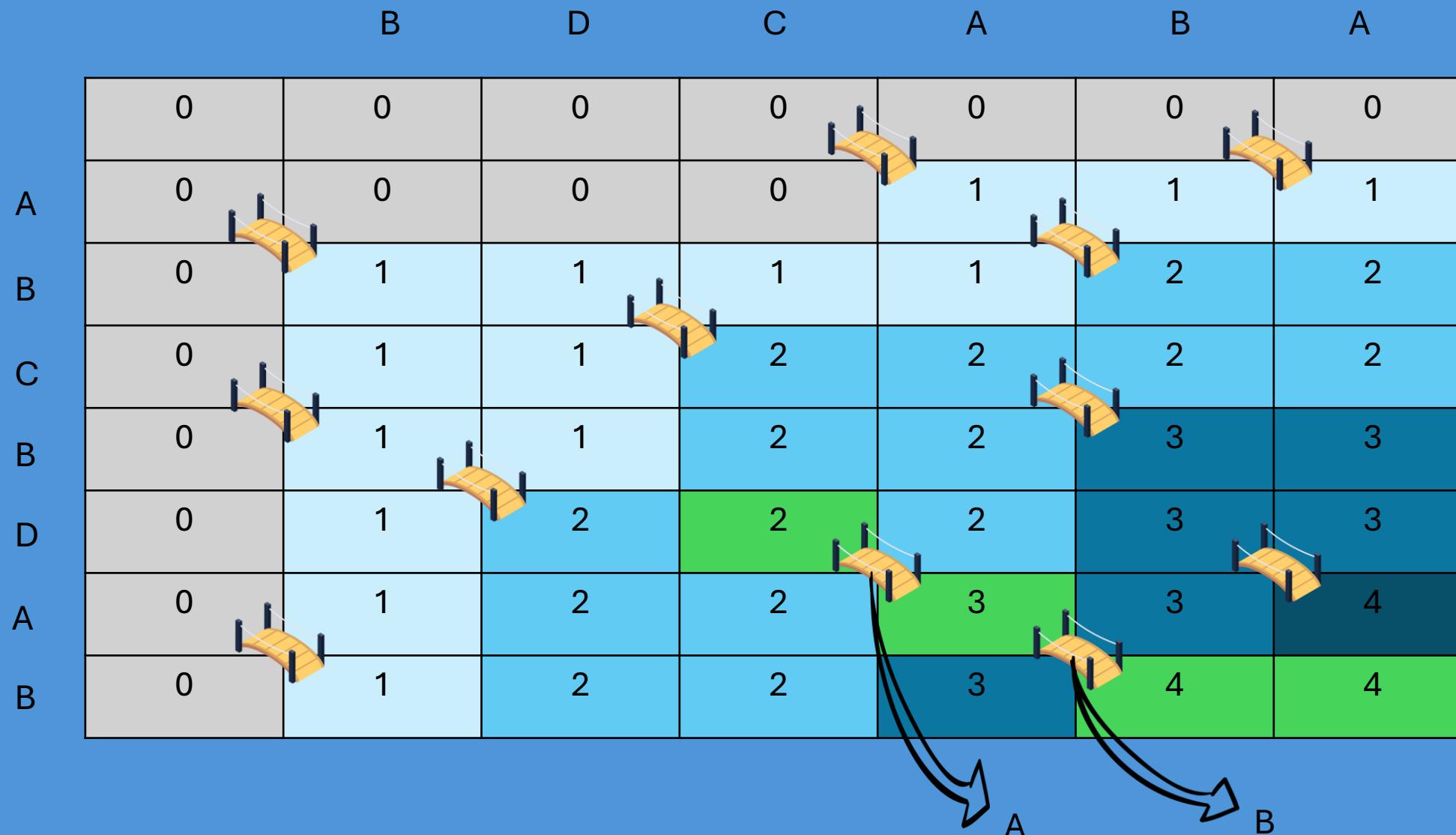
Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c





Règles à suivre

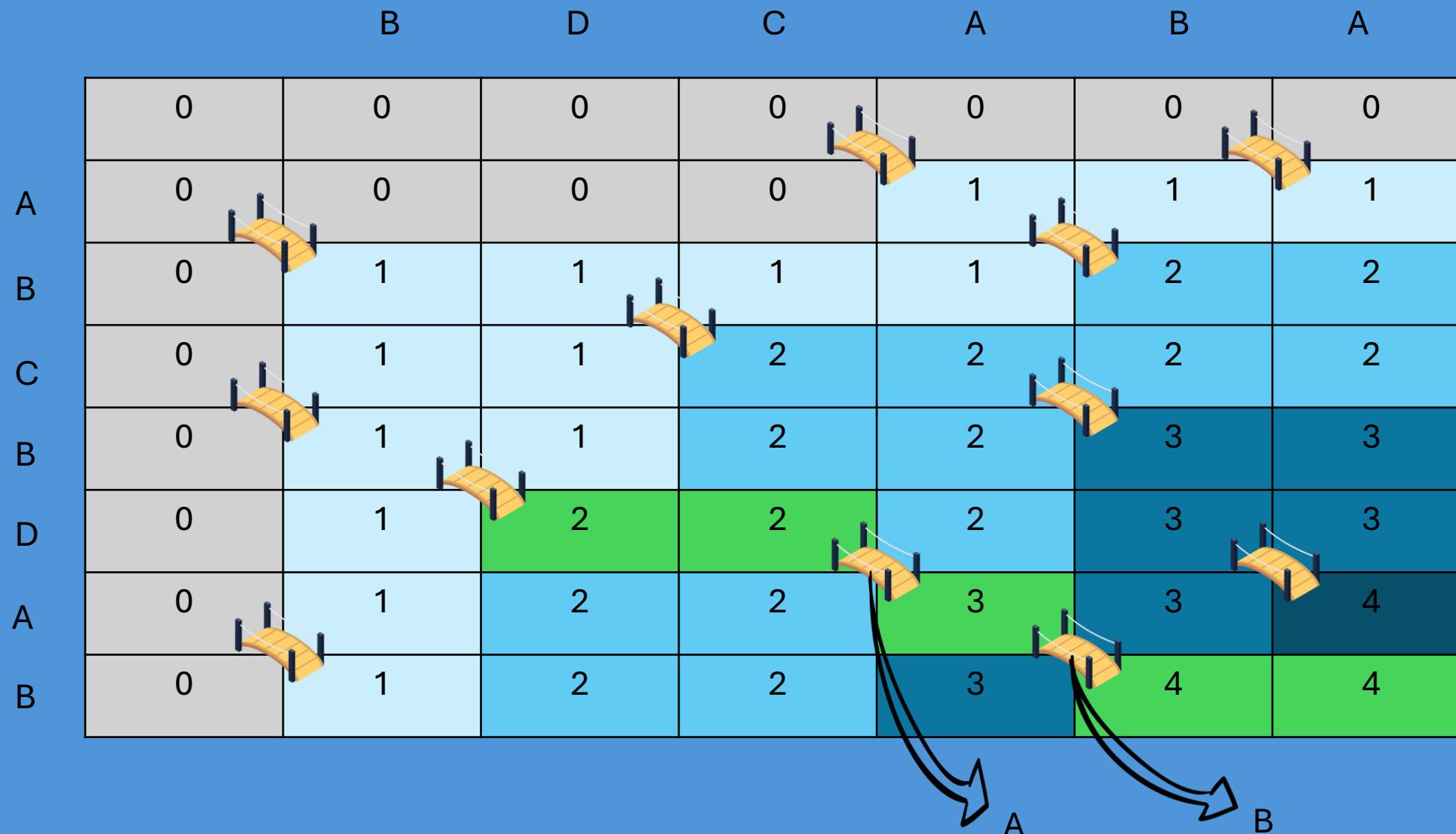
Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c





Règles à suivre

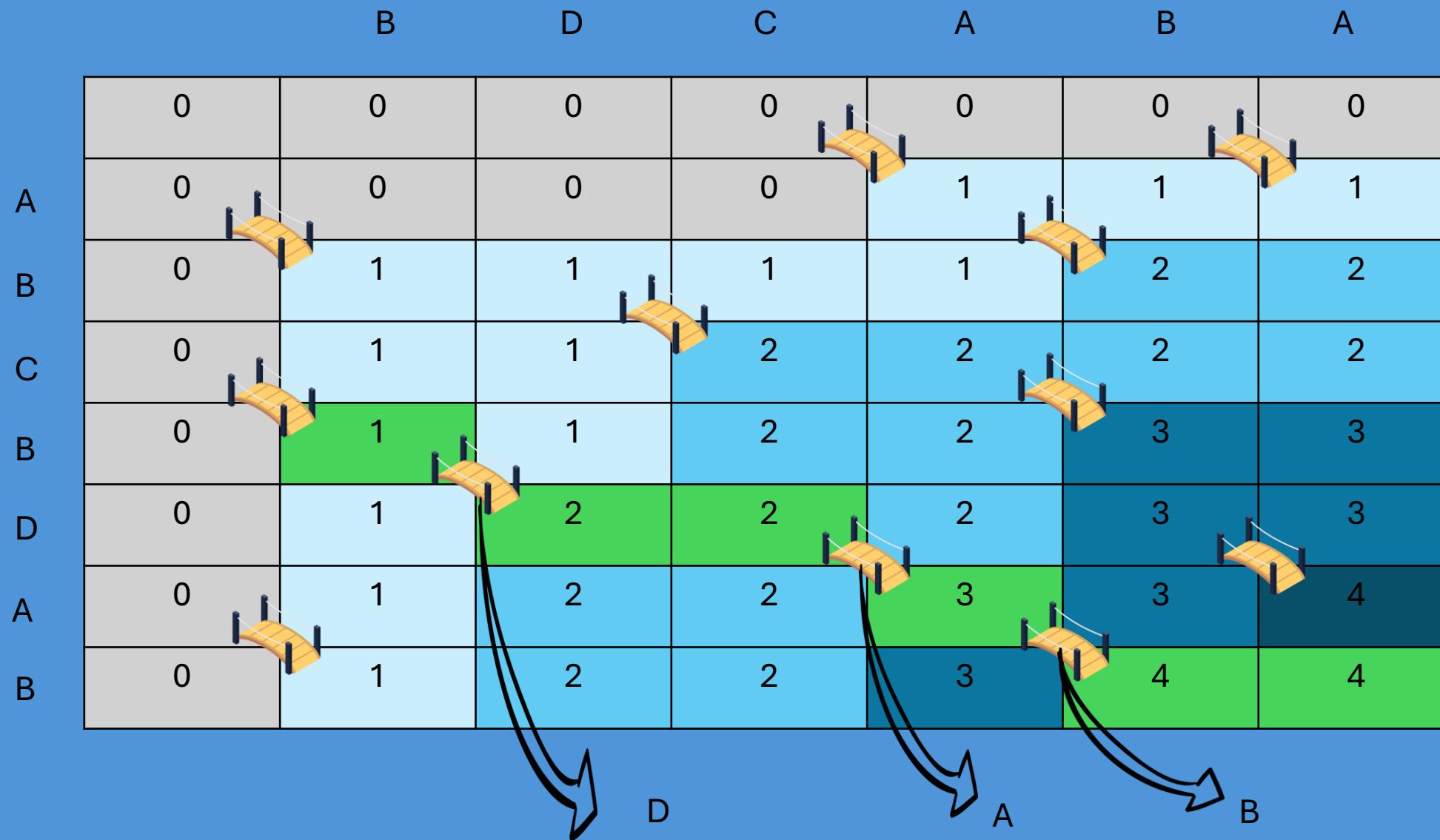
Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c





Règles à suivre

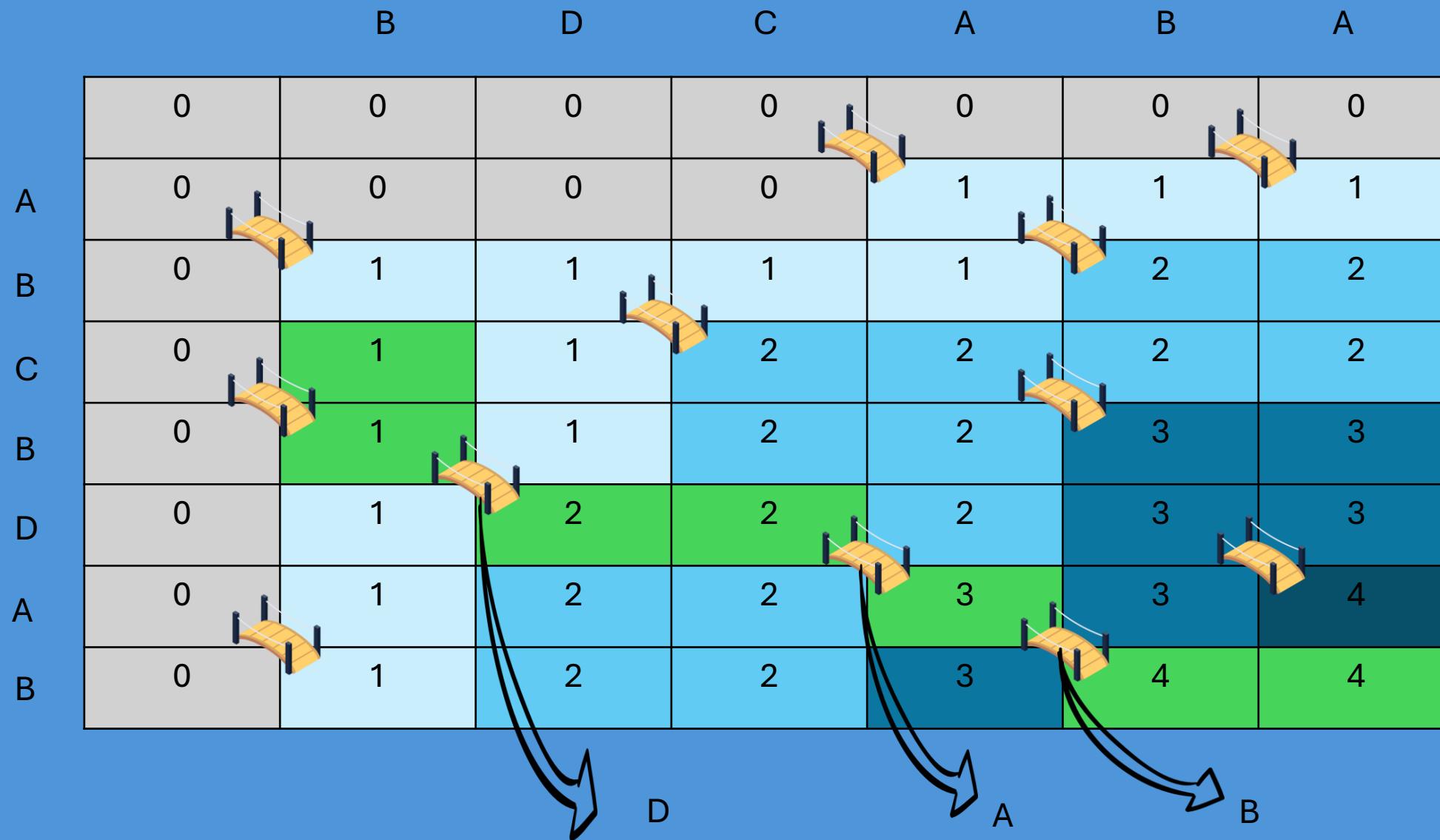
Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c





Règles à suivre

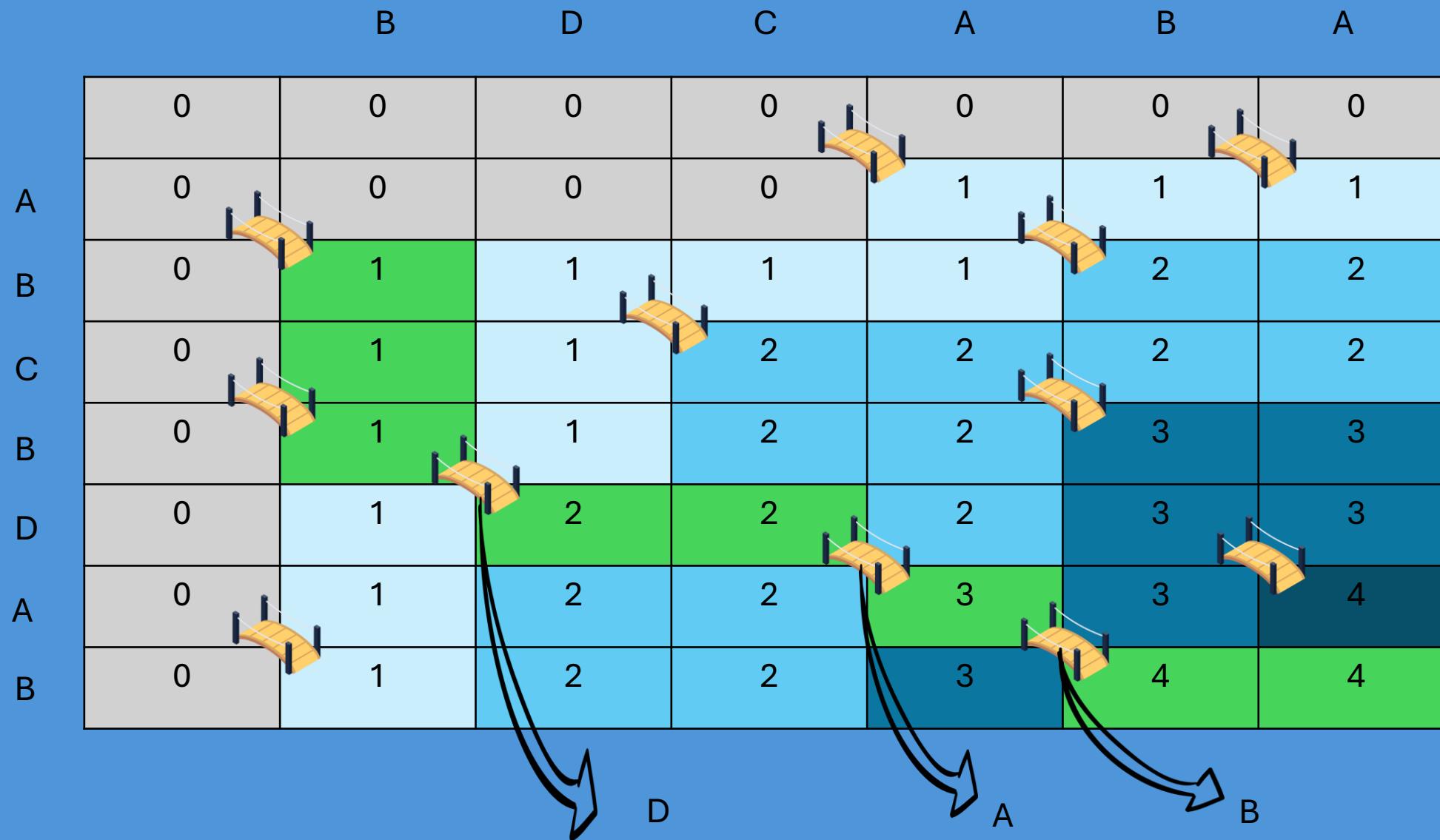
Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c





Règles à suivre

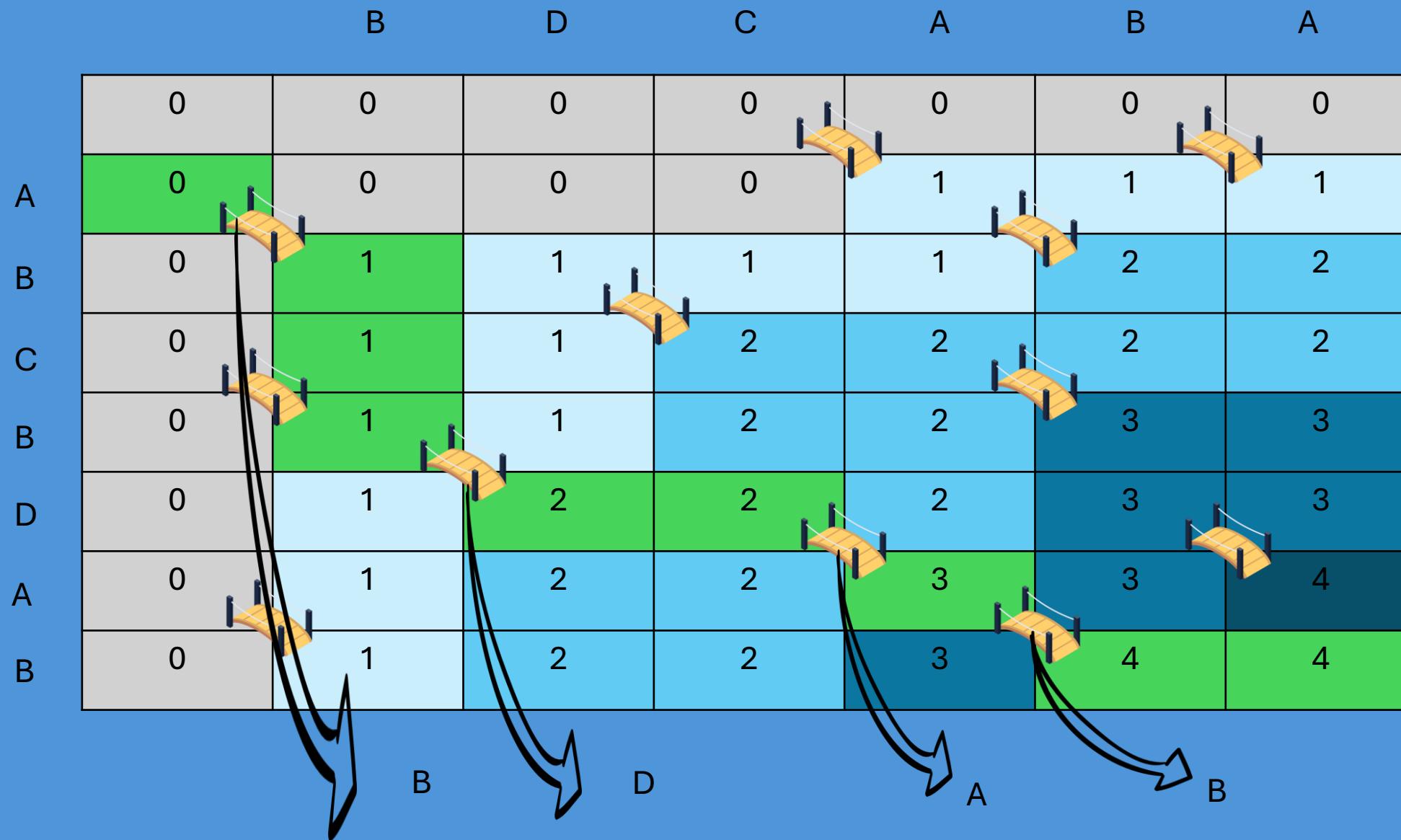
Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c





Règles à suivre

Correspondance

a	
	a+1

Non Correspondance

	c
b	

Plus grand  
entre b et c



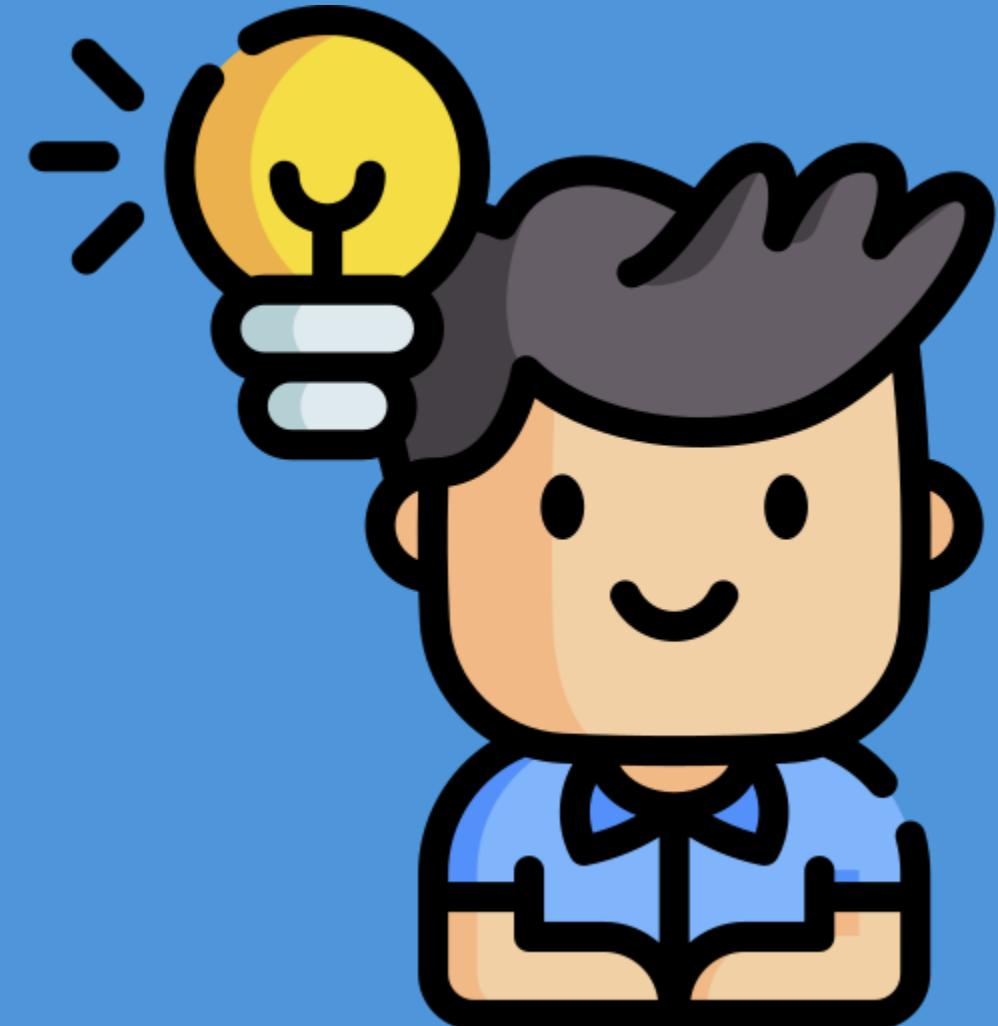
La plus longue sous séquence: BDAB



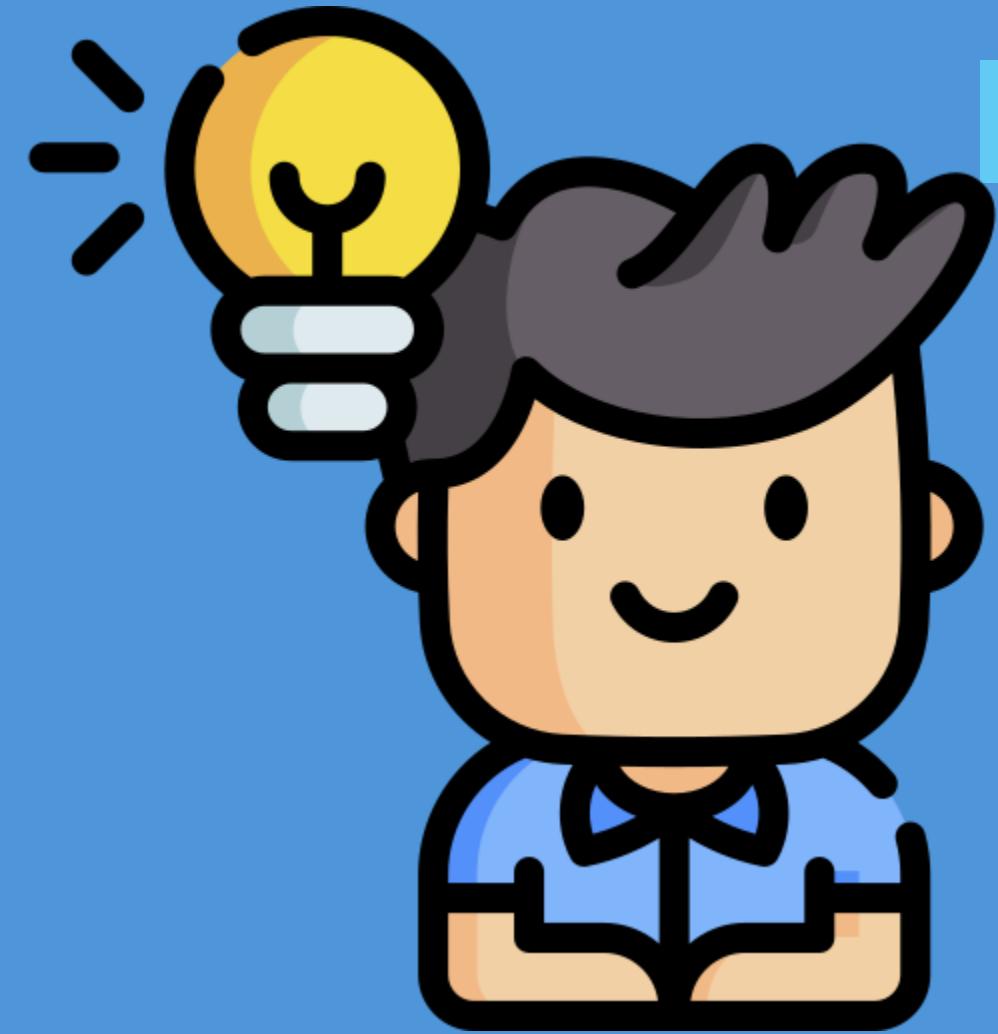
---

La plus longue sous  
séquence: BDAB

Comment calculer le  
coefficient de similarité?



Coefficient de similarité = 
$$\frac{\text{longeur de la sous séquence commune}}{\min(\text{longeur séquence } A, \text{longeur séquence } B)}$$



$$\text{Coefficient de similarité} = \frac{\text{longeur de la sous séquence commune}}{\min(\text{longeur séquence A}, \text{longeur séquence B})}$$

Dans notre exemple:

Séquence A: BDCABA

Séquence B: ABCBDAB

La plus longue sous séquence commune: BDAB

Longueur(séquence A) = 6

Longueur (Séquence B) = 7

Longueur (sous séquence commune) = 4

$$\text{Coefficient de similarité} = \frac{4}{6} = 66,67\%$$

## Configuration de l'API OpenRouteService :

- Chargement de la clé API : La clé API est lue depuis un fichier « OpenRouteService.txt ».
- Création d'un client ORS : Un client OpenRouteService est initialisé avec la clé API.

```
# Charger la clé API depuis un fichier
API = open("OpenRouteService.txt", "r").read()
client = ors.Client(key=API)
# si hosted en local
# client = ors.Client(base_url='localhost:8080/ors')
```

## Définition des coordonnées des utilisateurs :

- Coordonnées des utilisateurs : Des paires de coordonnées sont définies pour simuler les points de départ et d'arrivée de différents utilisateurs.

```
# Coordonnées des utilisateurs
coords_user1 = [[6.183264, 48.693684], [6.172724, 48.667496]]
coords_user2 = [[6.183264, 48.693684], [6.172724, 48.667496]]
coords_user3 = [[6.175689, 48.697079], [6.142902, 48.679287]]
coords_user4 = [[6.171398, 48.712372], [6.204700, 48.710333]]
coords_user5 = [[6.156463, 48.686314], [6.178264, 48.672939]]
coords_user6 = [[6.165390, 48.686767], [6.169338, 48.666137]]
```

## Obtention des itinéraires :

- Demande d'itinéraires : Pour chaque paire de coordonnées, une requête est envoyée à l'API OpenRouteService pour obtenir un itinéraire cyclable. Ces itinéraires sont stockés dans des variables distinctes.

```
# Obtenir les itinéraires pour chaque utilisateur
route_user1 = client.directions(coordinates=coords_user1, profile='cycling-regular', format='geojson')
route_user2 = client.directions(coordinates=coords_user2, profile='cycling-regular', format='geojson')
route_user3 = client.directions(coordinates=coords_user3, profile='cycling-regular', format='geojson')
route_user4 = client.directions(coordinates=coords_user4, profile='cycling-regular', format='geojson')
route_user5 = client.directions(coordinates=coords_user5, profile='cycling-regular', format='geojson')
route_user6 = client.directions(coordinates=coords_user6, profile='cycling-regular', format='geojson')
```

## Comparaison des itinéraires :

- Longest Common Subsequence (LCS) : La fonction « `longest_common_subsequence` » (LCS) conçue pour identifier les segments d'itinéraire partagés entre deux ensembles de coordonnées.
- Calcul de similarité : La fonction « `compare_routes` », exploitant la méthode Longest Common Subsequence (LCS) pour quantifier la similarité de manière objective. Cette fonction est essentielle pour l'application, permettant de déterminer les utilisateurs ayant des segments de trajet en commun significatifs.

```
# Fonction pour trouver la plus longue sous-séquence commune entre deux listes
def longest_common_subsequence(list1, list2):
    #algorithme LCS
```

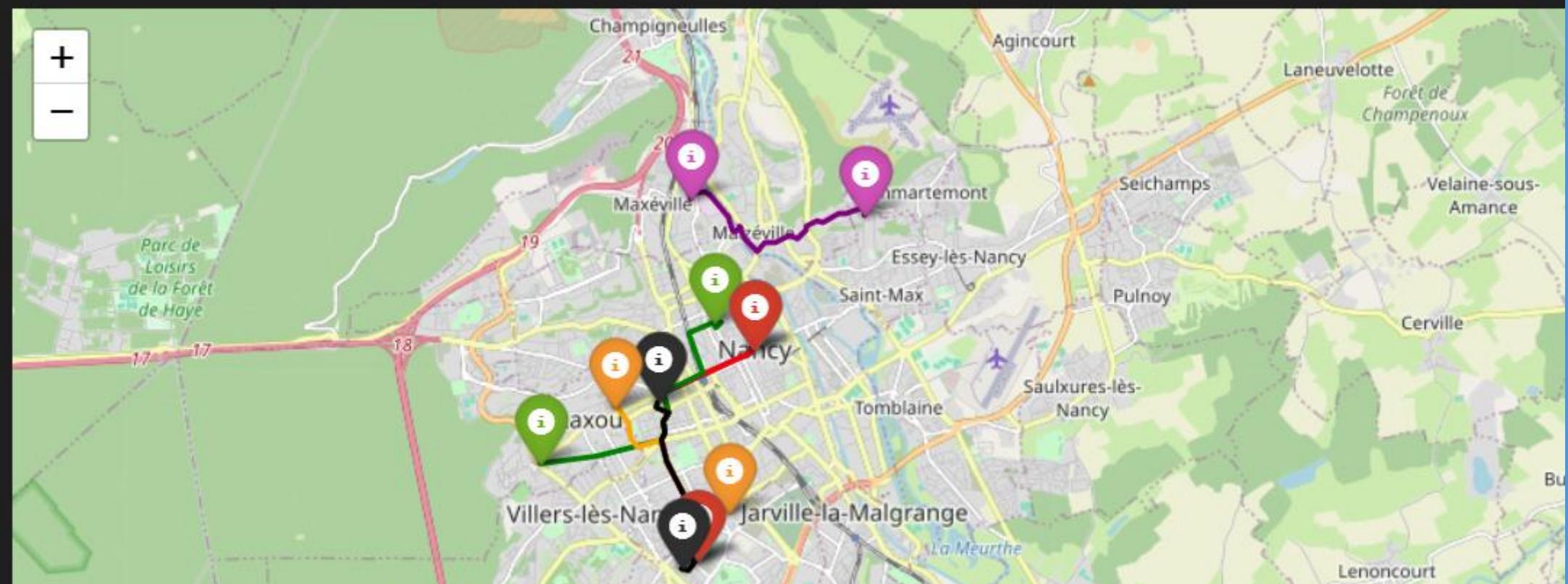
```
# Fonction pour comparer deux tracés de chemin et déterminer la similarité
def compare_routes(route1, route2):
    coords1 = route1['features'][0]['geometry']['coordinates']
    coords2 = route2['features'][0]['geometry']['coordinates']

    lcs = longest_common_subsequence(coords1, coords2)
    similarity = len(lcs) / min(len(coords1), len(coords2))

    return lcs, similarity
```

## Utilisateurs avec des parties de chemin communes :

User1 et User2 - Similarité : 1.0, Début : [6.183232, 48.69373], Fin : [6.172721, 48.667494]  
User1 et User3 - Similarité : 0.34782608695652173, Début : [6.173629, 48.690584], Fin : [6.165969, 48.682613]  
User1 et User5 - Similarité : 0.475, Début : [6.165851, 48.682259], Fin : [6.174537, 48.671782]  
User1 et User6 - Similarité : 0.74, Début : [6.16689, 48.68601], Fin : [6.17187, 48.667993]  
User2 et User3 - Similarité : 0.34782608695652173, Début : [6.173629, 48.690584], Fin : [6.165969, 48.682613]  
User2 et User5 - Similarité : 0.475, Début : [6.165851, 48.682259], Fin : [6.174537, 48.671782]  
User2 et User6 - Similarité : 0.74, Début : [6.16689, 48.68601], Fin : [6.17187, 48.667993]  
User3 et User5 - Similarité : 0.0125, Début : [6.161173, 48.681519], Fin : [6.161173, 48.681519]  
User3 et User6 - Similarité : 0.16, Début : [6.16689, 48.68601], Fin : [6.165969, 48.682613]  
User5 et User6 - Similarité : 0.475, Début : [6.165851, 48.682259], Fin : [6.174537, 48.671782]



# Hébergement local de l'API OpenRouteService



Activities

Feb 14 17:06



sabri@sabri-VirtualBox: ~



```
sabri@sabri-VirtualBox:~$ git clone --branch v7.1.1 https://github.com/GIScience/openrouteservice.git
```



Home

Activities Terminal

Feb 14 17:07



sabri@sabri-VirtualBox: ~/Documents

sabri@sabri-VirtualBox:~/Documents\$ cd openrouteservice

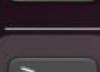


Home

Activities

Terminal

Feb 14 17:08



sabri@sabri-VirtualBox: ~/Documents/openrouteservice

```
sabri@sabri-VirtualBox:~/Documents$ cd openrouteservice
sabri@sabri-VirtualBox:~/Documents$ cd docker/
```



Home

Activities

Terminal

Feb 14 17:08



```
sabri@sabri-VirtualBox: ~/Documents/openrouteservice/doc...  
sabri@sabri-VirtualBox:~/Documents/openrouteservice$ cd openrouteservice  
sabri@sabri-VirtualBox:~/Documents/openrouteservice$ cd docker/  
sabri@sabri-VirtualBox:~/Documents/openrouteservice/docker$ mkdir -p conf elevation_cache graphs logs\ors logs\tomcat
```



Home

Activities

Terminal

Feb 14 17:05



sabri@sabri-VirtualBox: ~



sabri@sabri-VirtualBox:~\$ systemctl start docker



Home

Activities

Terminal

Feb 14 17:10



sabri@sabri-VirtualBox: ~/Documents/openrouteservice/doc...

sabri@sabri-VirtualBox:~/Documents/openrouteservice/docker\$ sudo docker compose up



Home

Activities

Terminal

Feb 14 17:11



```
sabri@sabri-VirtualBox: ~/Documents/openrouteservice/doc... ] =====> Recycling gar  
bage...  
ors-app | 2024-02-14 16:11:05,535 INFO ORS-Init [ o.h.o.r.RoutingProfileManager ] Before: Total - 10  
24 MB, Free - 567.46 MB, Max: 2 GB, Used - 456.54 MB  
ors-app | 2024-02-14 16:11:05,653 INFO ORS-Init [ o.h.o.r.RoutingProfileManager ] After: Total - 102  
4 MB, Free - 900.80 MB, Max: 2 GB, Used - 123.20 MB  
ors-app | 2024-02-14 16:11:05,657 INFO ORS-Init [ o.h.o.r.RoutingProfileManager ] =====  
=====  
ors-app | 2024-02-14 16:11:05,659 INFO ORS-Init [ o.h.o.r.RoutingProfileManager ] =====> Memory usage  
by profiles:  
ors-app | 2024-02-14 16:11:05,679 INFO ORS-Init [ o.h.o.r.RoutingProfileManager ] [1] 44.04 MB (35.5%  
)  
ors-app | 2024-02-14 16:11:05,686 INFO ORS-Init [ o.h.o.r.RoutingProfileManager ] Total: 44.04 MB (35.  
.5%)  
ors-app | 2024-02-14 16:11:05,688 INFO ORS-Init [ o.h.o.r.RoutingProfileManager ] =====
```



Home

localhost:8080/ors/v2/directions/driving-car?start=8.681495,49.41461&end=8.687872,49.420318

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

```
type: "FeatureCollection"
metadata:
  attribution: "openrouteservice.org, OpenStreetMap contributors"
  service: "routing"
  timestamp: 1662679009548
query:
  coordinates:
    0:
      0: 8.681495
      1: 49.41461
    1:
      0: 8.687872
      1: 49.420318
  profile: "driving-car"
  format: "json"
engine:
  version: "6.7.0"
  build_date: "2022-09-08T23:10:10Z"
  graph_date: "2022-09-08T23:15:03Z"
features:
  0:
    bbox:
      0: 8.681445
      1: 49.41461
      2: 8.690123
      3: 49.420514
    type: "Feature"
    properties:
      segments:
        0:
          distance: 1365.3
          duration: 315.2
          steps:
            0:
              distance: 312.2
              duration: 74.9
              type: 11
              instruction: "Head north on Wielandstraße"
              name: "Wielandstraße"
              way_points: [...]
            1:
              distance: 253.2
              duration: 60.8
              type: 1
```

# Index of /extracts/europe/france/lorraine

	<a href="#"><u>Name</u></a>	<a href="#"><u>Last modified</u></a>	<a href="#"><u>Size</u></a>	<a href="#"><u>Description</u></a>
	<a href="#">Parent Directory</a>		-	
	<a href="#">meurthe_et_moselle-latest.osm.pbf</a>	2024-02-13 00:31	50M	
	<a href="#">meurthe_et_moselle.osm.pbf</a>	2024-02-13 00:31	50M	
	<a href="#">meurthe_et_moselle.state.txt</a>	2024-02-13 00:31	56	
	<a href="#">meuse-latest.osm.pbf</a>	2024-02-13 00:31	21M	
	<a href="#">meuse.osm.pbf</a>	2024-02-13 00:31	21M	
	<a href="#">meuse.state.txt</a>	2024-02-13 00:31	56	
	<a href="#">moselle-latest.osm.pbf</a>	2024-02-13 00:31	77M	
	<a href="#">moselle.osm.pbf</a>	2024-02-13 00:31	77M	
	<a href="#">moselle.state.txt</a>	2024-02-13 00:31	56	
	<a href="#">vosges-latest.osm.pbf</a>	2024-02-13 00:31	40M	
	<a href="#">vosges.osm.pbf</a>	2024-02-13 00:31	40M	
	<a href="#">vosges.state.txt</a>	2024-02-13 00:31	56	

Activities

Files

Feb 14 17:15



sabri@sabri-VirtualBox: ~/Documents/openrouteservice/doc... [ ORS-Init [ o.h.o.r.RoutingProfileManager ] =====&gt; Recycling gar

bage...

ors-app | 2024-02-14 16:1

ORS-Init [ o.h.o.r.Ro

24 MB, Free - 567.46 MB, M

ors-app | 2024-02-14 16:1

ORS-Init [ o.h.o.r.Ro

4 MB, Free - 900.80 MB, Ma

ors-app | 2024-02-14 16:1

ORS-Init [ o.h.o.r.Ro

=====

ors-app | 2024-02-14 16:1

ORS-Init [ o.h.o.r.Ro

by profiles:

ors-app | 2024-02-14 16:1

ORS-Init [ o.h.o.r.Ro

)

ors-app | 2024-02-14 16:1

ORS-Init [ o.h.o.r.Ro

.5%)

ors-app | 2024-02-14 16:1

ORS-Init [ o.h.o.r.Ro

=====

[]



Home / Documents / openrouteservice / docker



Recent

Starred

Home

Documents

Downloads

Music

Pictures

Videos

Trash

+ Other Locations

conf

data

elevation\_

cache

graphs

logs

logso

logstomcat

docker-

compose.

yml

meurthe\_

et\_moselle.

osm.pbf



Home

Activities

Text Editor

Feb 14 17:15



```
docker-compose.yml
~/Documents/openrouteservice/docker

1 version: '2.4'
2 services:
3   ors-app:
4     container_name: ors-app
5     ports:
6       - "8080:8080"
7       - "9001:9001"
8     image: openrouteservice/openrouteservice:latest
9     user: "${UID:-0}:${GID:-0}"
10    build:
11      context: ../
12      args:
13        ORS_CONFIG: ./openrouteservice/src/main/resources/ors-config-sample.json
14        OSM_FILE: ./openrouteservice/src/main/files/heidelberg.osm.gz
15    volumes:
16      - ./graphs:/home/ors/ors-core/data/graphs
17      - ./elevation_cache:/home/ors/ors-core/data/elevation_cache
18      - ./logs/ors:/home/ors/ors-core/logs/ors
19      - ./logs/tomcat:/home/ors/tomcat/logs
20      - ./conf:/home/ors/ors-conf
21      - ./meurthe_et_moselle.osm.pbf:/home/ors/ors-core/data/osm_file.pbf
22    environment:
23      - BUILD_GRAPHS=False # Forces the container to rebuild the graphs, e.g. when PBF is changed
24      - "JAVA_OPTS=-Djava.awt.headless=true -server -XX:TargetSurvivorRatio=75 -XX:SurvivorRatio=64 -XX:MaxTenuringThreshold=3 -XX:+UseG1GC -XX:+ScavengeBeforeFullGC -XX:ParallelGCThreads=4 -Xms1g -Xmx2g"
25      - "CATALINA_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9001 -Dcom.sun.management.jmxremote.rmi.port=9001 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=localhost"
```

Save





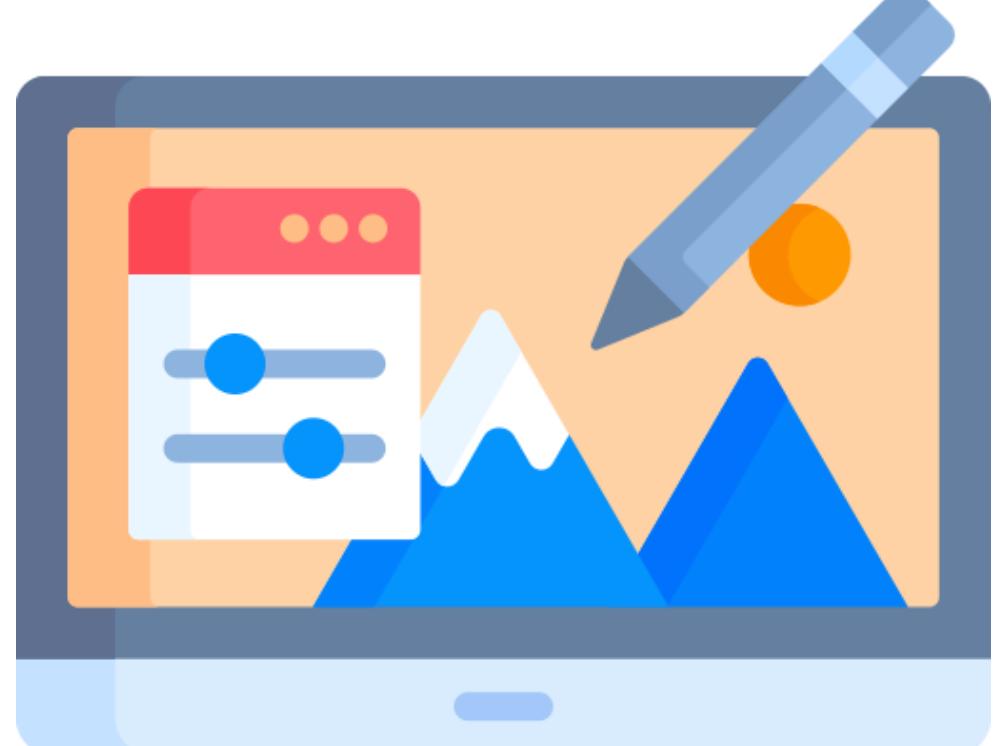
localhost:8080/ors/v2/directions/driving-car?&start=6.1679676,48.6744374&end=6.1482113,48.652941

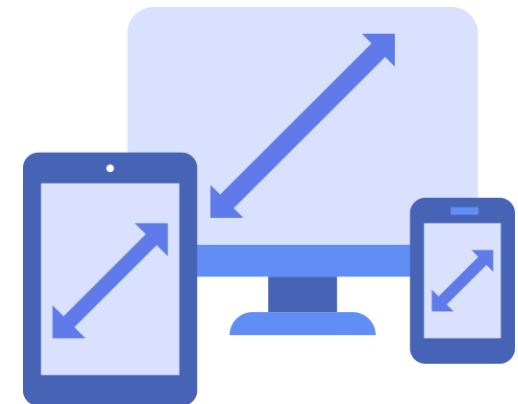
JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

```
type: "FeatureCollection"
metadata:
  attribution: "openrouteservice.org, OpenStreetMap contributors"
  service: "routing"
  timestamp: 1707927636517
query:
  coordinates:
    0:
      0: 6.1679676
      1: 48.6744374
    1:
      0: 6.1482113
      1: 48.652941
    profile: "driving-car"
    format: "json"
  engine:
    version: "7.2.0"
    build_date: "2023-11-16T15:05:04Z"
    graph_date: "2024-02-05T12:16:31Z"
bbox:
  0:
    0: 6.148413
    1: 48.651539
    2: 6.169268
    3: 48.674389
features:
  0:
    bbox:
      0:
        0: 6.148413
        1: 48.651539
        2: 6.169268
        3: 48.674389
        ...
```

# Bootstrap





# Page d'accueil

ACCEUIL

# Page d'accueil

Covélotage Accueil Parcours Communauté Profil

## Covélotage : Votre Communauté Cycliste

Connectez-vous avec des cyclistes expérimentés, trouvez des itinéraires sûrs et profitez d'une expérience de vélo confiante dans la métropole du Grand Nancy.

Commencer

### Trouver un Compagnon

Connectez-vous avec des cyclistes expérimentés partageant des itinéraires similaires. Améliorez la sécurité et renforcez la confiance en tant que cycliste novice.

### Découvrir des Parcours

Explorez des itinéraires de vélo sûrs et populaires dans le Grand Nancy. Obtenez des informations sur les lieux de stationnement, les points d'eau et plus encore.

### Soutien de la Communauté

Rejoignez une communauté cycliste dynamique. Interagissez avec d'autres utilisateurs, partagez des expériences et contribuez à un environnement de vélo urbain plus sûr.

# Page de login

Hello Again!

Explore More by connecting with us.

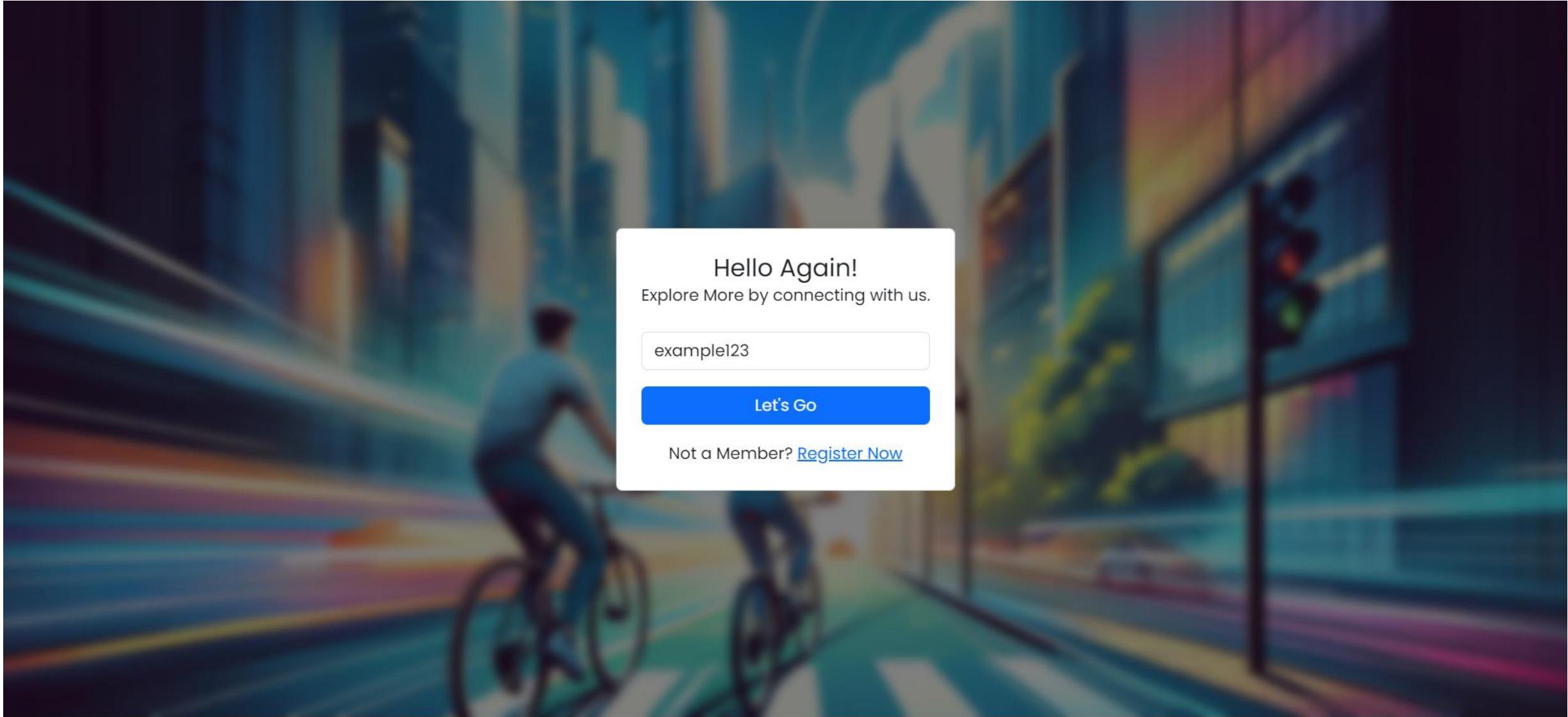
example123

Let's Go

Not a Member [Register Now](#)



# Page de login



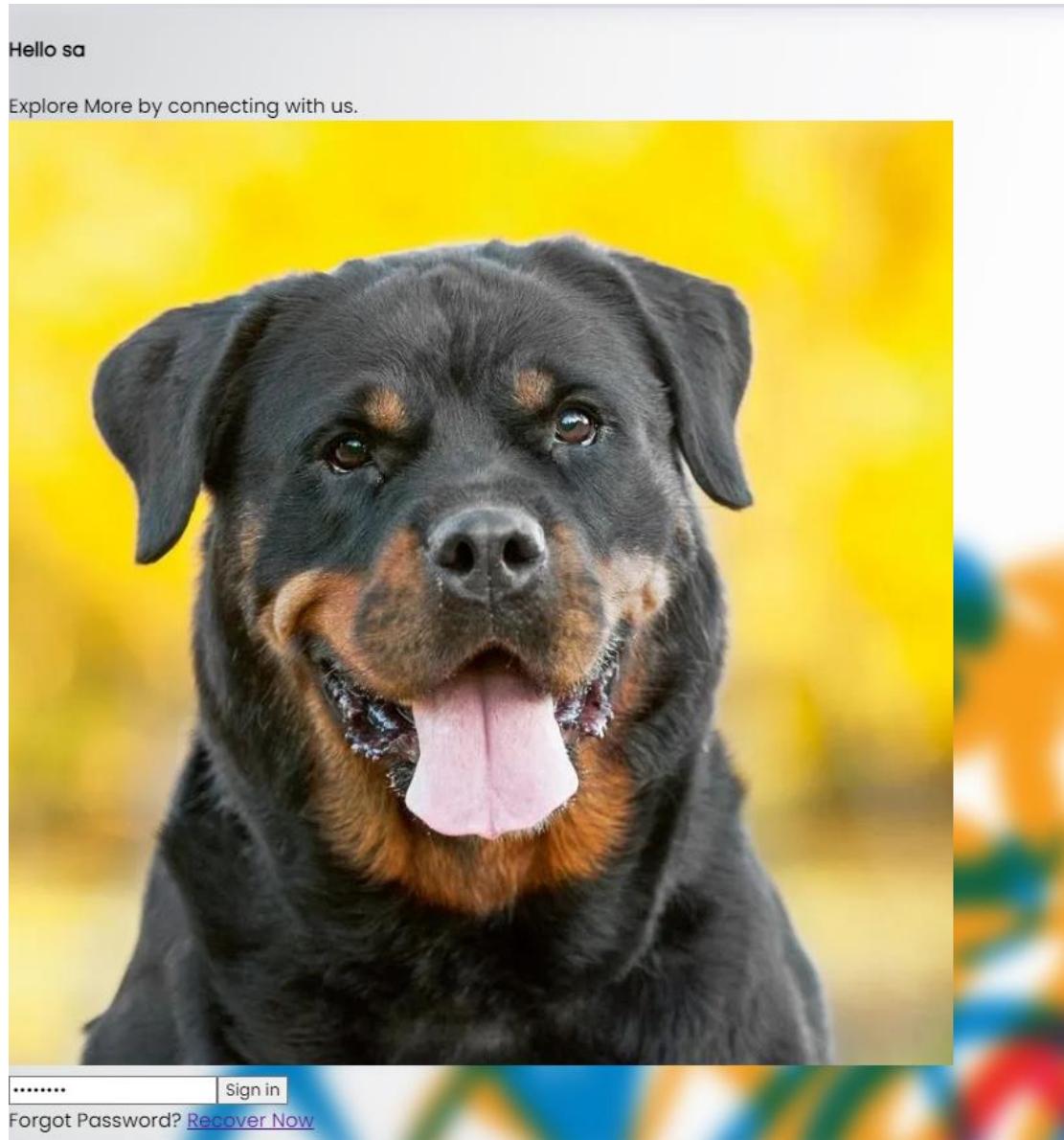
Hello Again!

Explore More by connecting with us.

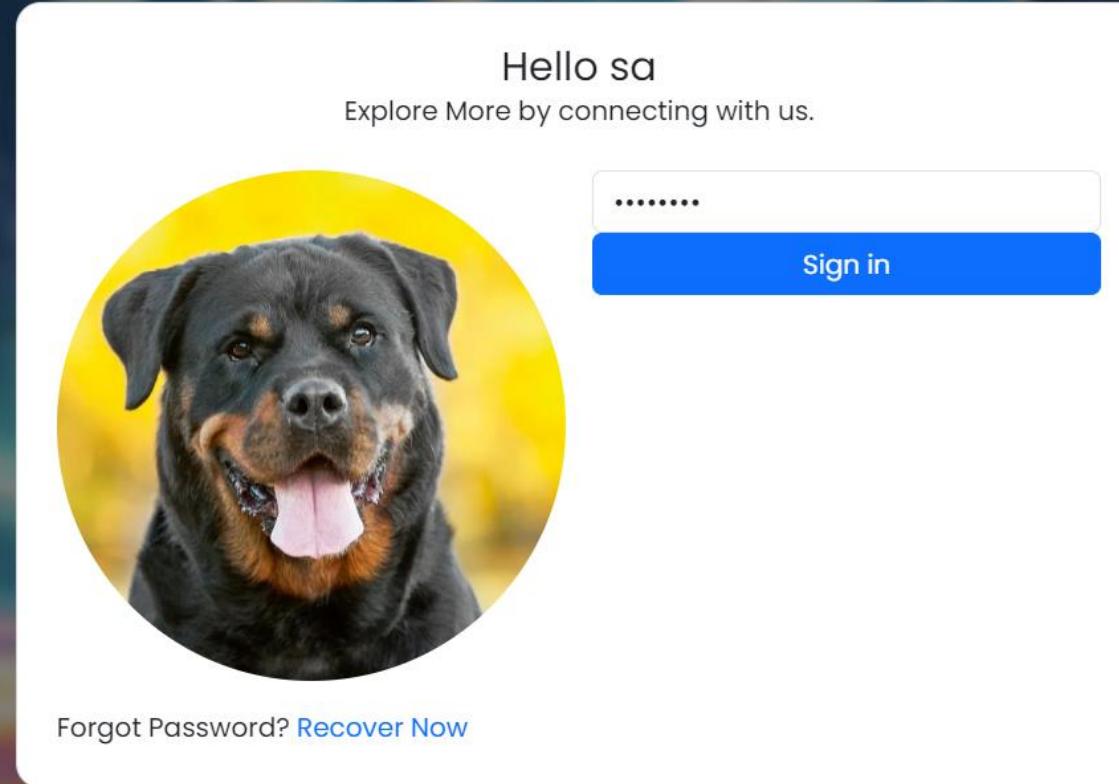
Let's Go

Not a Member? [Register Now](#)

# Page du password



# Page du password



# Page de profil

Profile

You can update the details.



Choisir un fichier Aucun fichier choisi

sa	be
9999	c666@gmail.com
42	Update

come back later? [Logout](#)



# Page de Profil

Profile  
You can update the details.



sa

be

9999

c666@gmail.com

42

**Update**

Come back later? [Logout](#)

# Page d'inscription « register »

Register

Happy to join you!

Choisir un fichier Aucun fichier choisi

c666@gmail.com example123 admin@12 Register

Already register? [Login now](#)

# Page d'inscription « register »

Register  
Happy to join you!



Aucun ...r choisi

Already registered? [Login now](#)

# Page de Recover

## Recovery

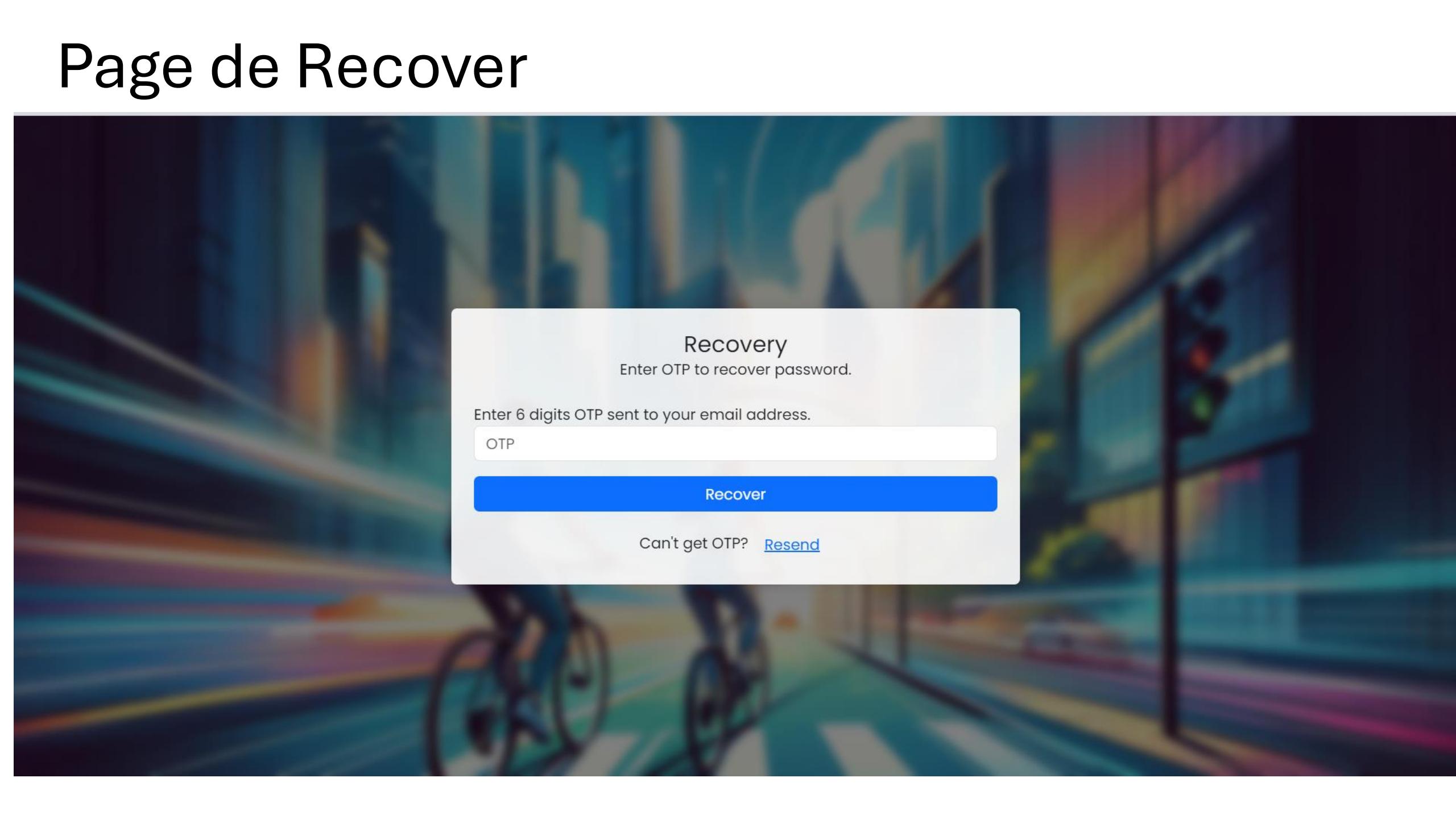
Enter OTP to recover password.

Enter 6 digits OTP sent to your emial address.

[Recover](#)

[Can't get OPT?](#) [Resent](#)

# Page de Recover



Recovery  
Enter OTP to recover password.

Enter 6 digits OTP sent to your email address.

Recover

Can't get OTP? [Resend](#)

# Page de Reset du mot de passe

**Reset**

Enter new password.

# Page de Reset du mot de passe



A blurred background image of a city skyline at night, featuring colorful lights from buildings and traffic.

Reset

Enter new password.

New Password

Repeat Password

Reset



# Conception et développement du Site

# PLAN

I /



Architecture et technologies



II /

Gestion des comptes utilisateurs

III /



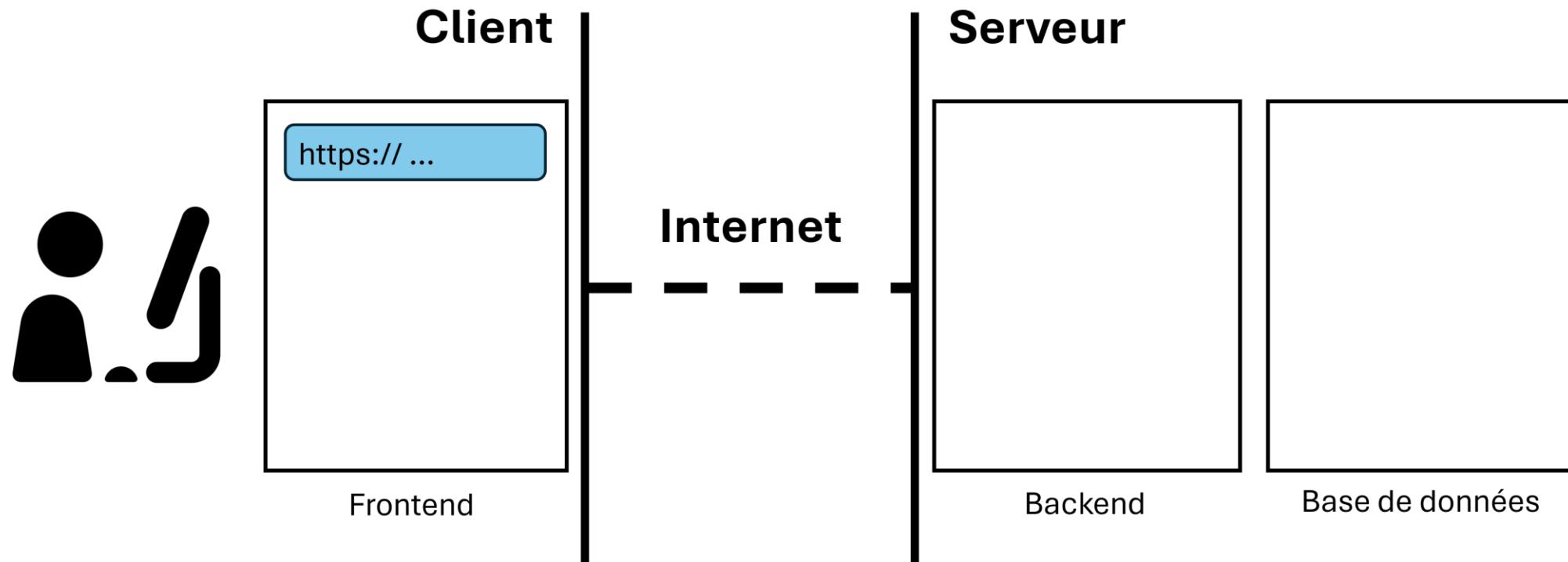
Gestion des trajets



# Architecture et technologies



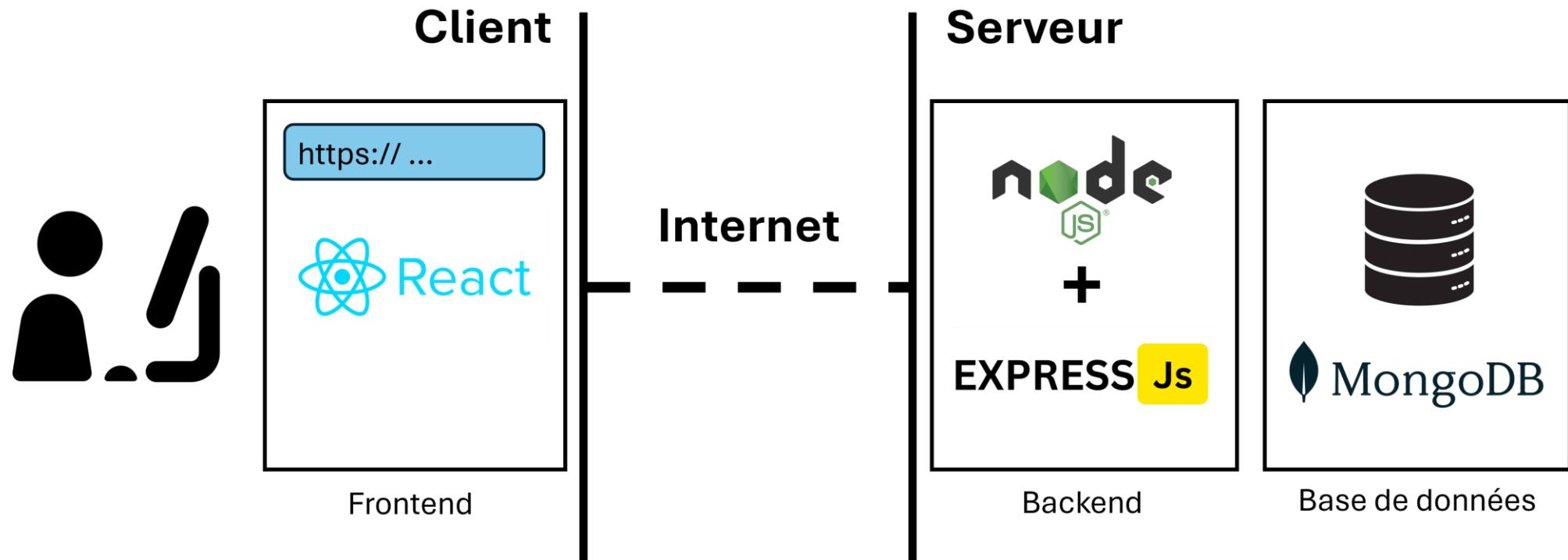
## › Fonctionnement d'une application web





## › Pile MERN

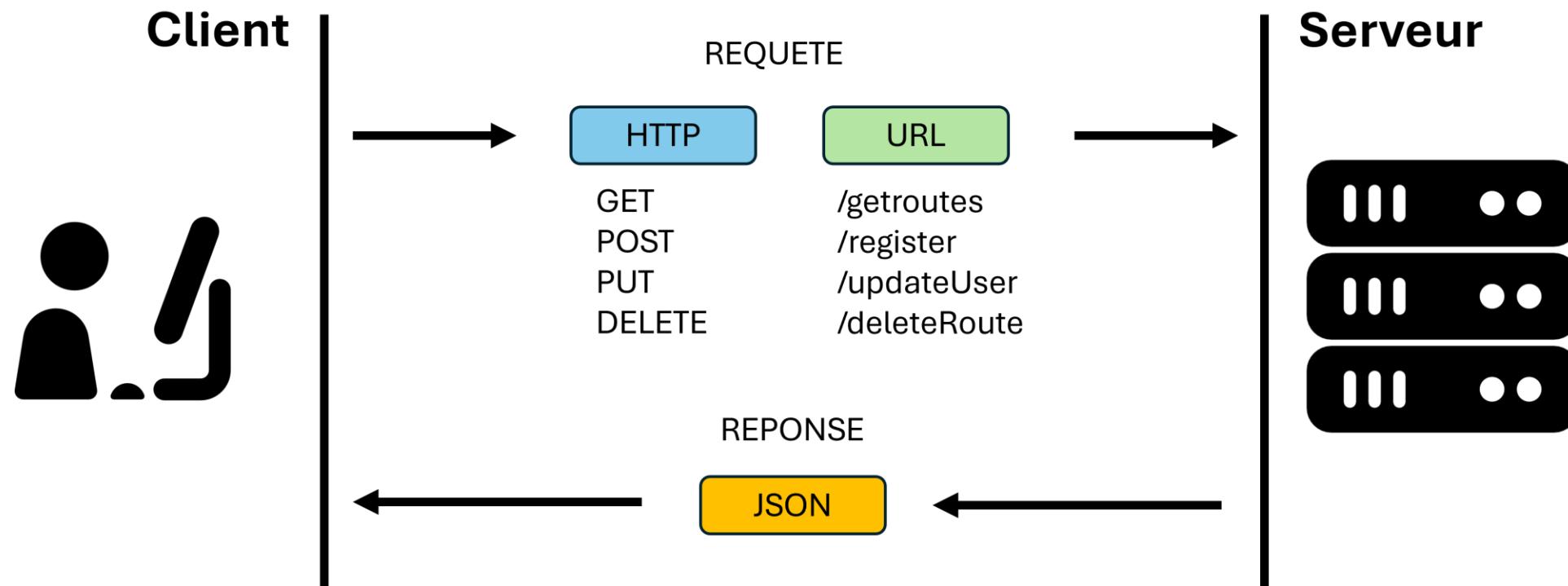
MongoDB . Express . React . Node.js





## > Communication

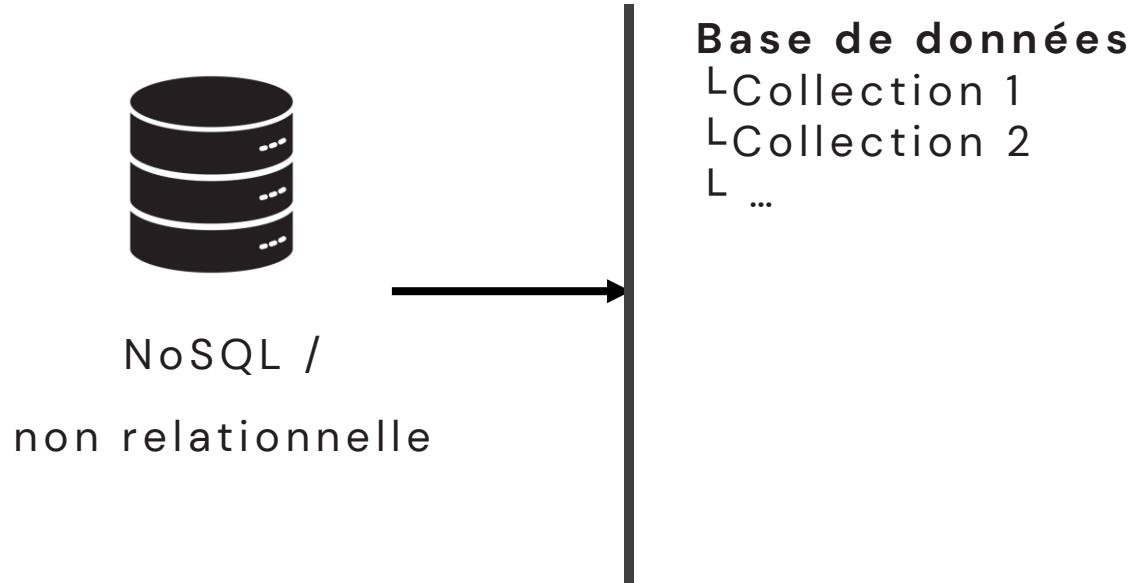
### RESTful API





## › Base de données

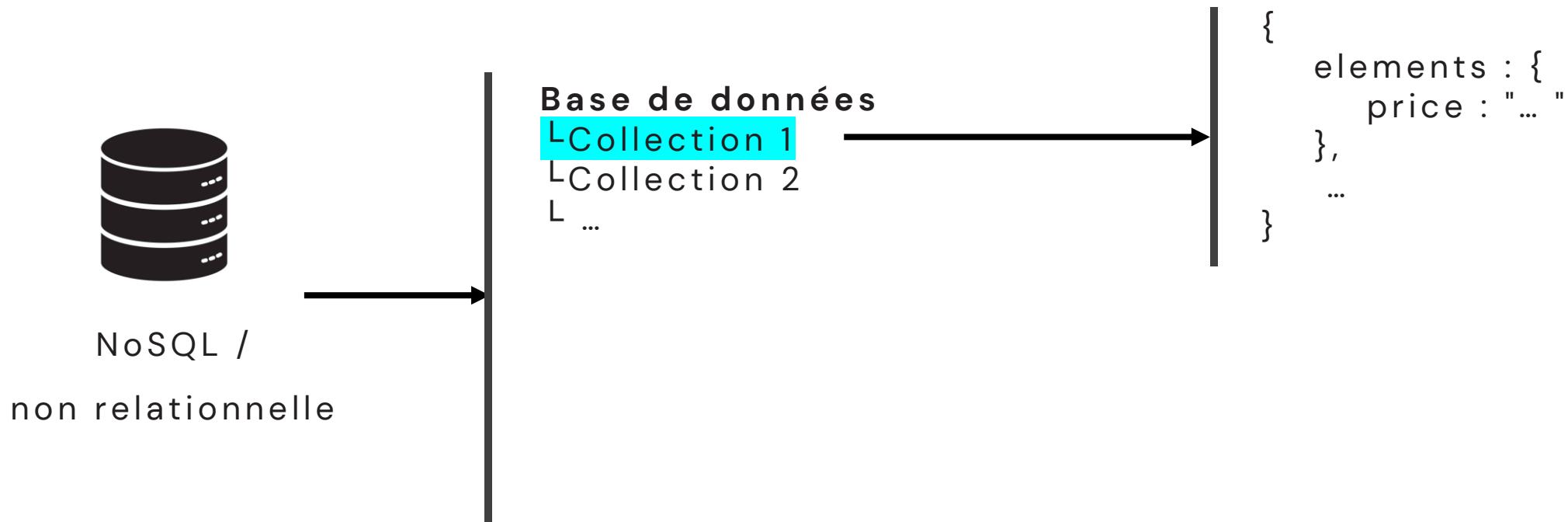
MongoDB





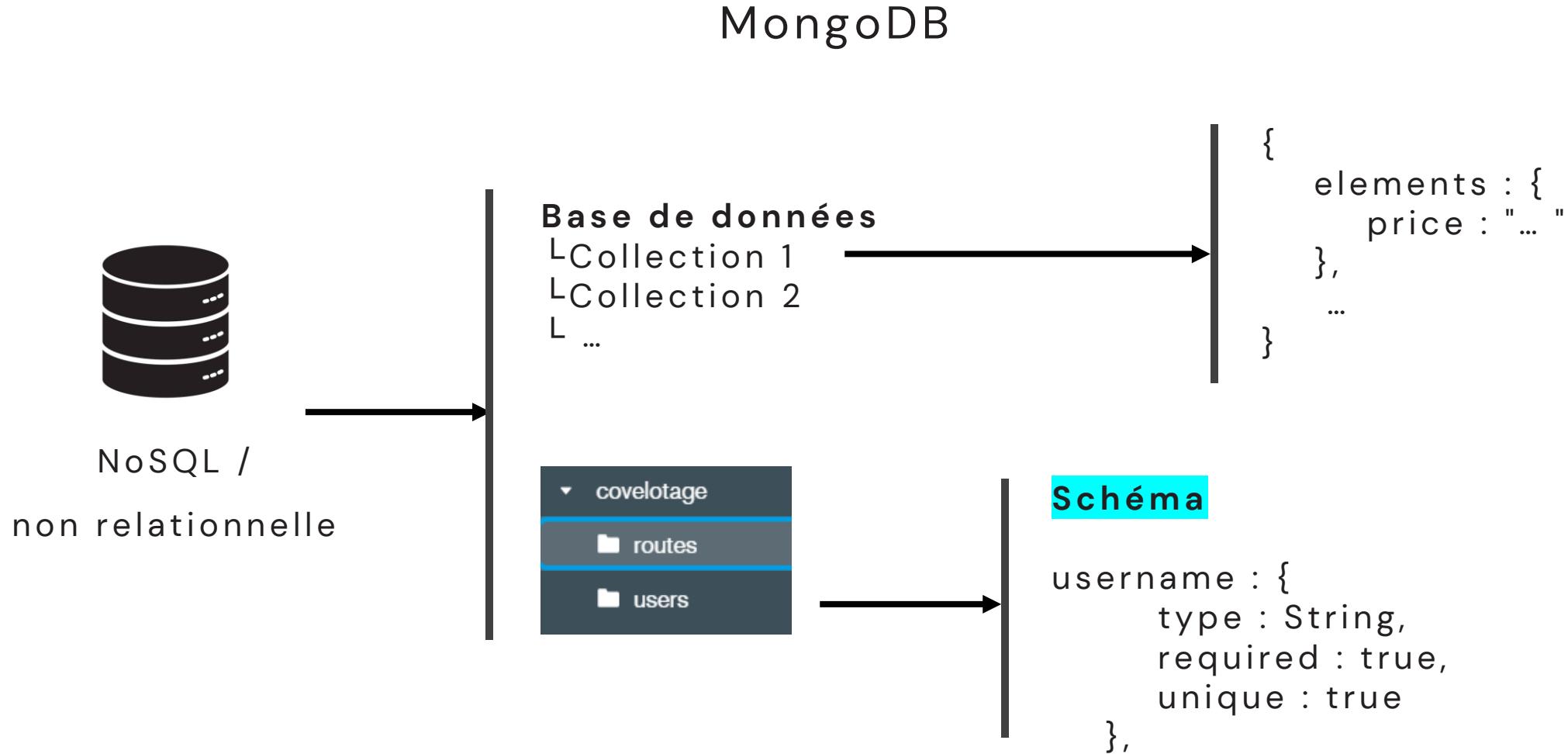
## › Base de données

MongoDB





## › Base de données

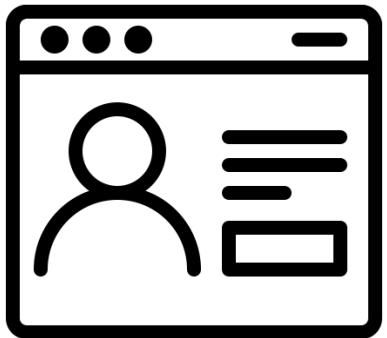




# Gestion des comptes utilisateurs



## > Fonctionnalités à intégrer



- Crédit d'un compte
- Connexion à un compte existant
- Modification du profil utilisateur
- Récupération de mot de passe





## > Crédation d'un compte

L>Frontend

1

The screenshot shows a registration page at the URL 192.168.1.12:3000/register. The page has a light gray background with a large blue circular profile picture placeholder on the left. At the top, it says "Register" and "Happy to join you!". Below that are three input fields: "Email\*", "Username\*", and "Password\*". A large blue "Register" button is centered below the fields. At the bottom, there are two buttons: "Choisir un fichier" and "Aucun fichier choisi", followed by the text "Already registered? [Login now](#)".

### Contraintes

#### Email

- non vide
- Respect du format xxx@xx.xx

#### Nom d'utilisateur

- non vide
- pas d'espaces

#### Mot de passe

- non vide
- pas d'espaces
- plus de 4 caractères dont un spécial



## > Crédation d'un compte

L>Frontend 2

The screenshot shows a registration page at the URL 192.168.1.12:3000/register. The page title is "Register" and the subtext is "Happy to join you!". It features a large circular profile picture placeholder containing an image of Master Chief from Halo. Below the placeholder is a file input field labeled "Choisir un fichier" with "halo.jpg" selected. The registration form consists of three input fields: an email field containing "masterchief@gmail.com", a first name field containing "john177", and a password field containing ".....". A blue "Register" button is positioned below these fields. At the bottom of the form, there is a link "Already registered? [Login now](#)".

## Opérations

### Image (optionnel)

- Redimensionnement
- Base64

### Autres champs

- Vérification des contraintes



## > Création d'un compte

L>Frontend

3

192.168.1.12:3000/register

Register  
Happy to join you!

masterchief@gmail.com

john177

.....

Register

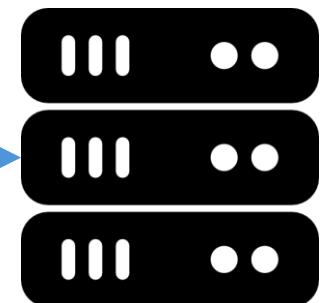
Choisir un fichier halo.jpg

Already registered? [Login now](#)

Serveur

REQUETE

POST /register



Body : {

**email** : masterchief@gmail.com,

**username** : john17,

**password** : admin@12,

**image** : ...4AAQSKZ...

}



## > Création d'un compte

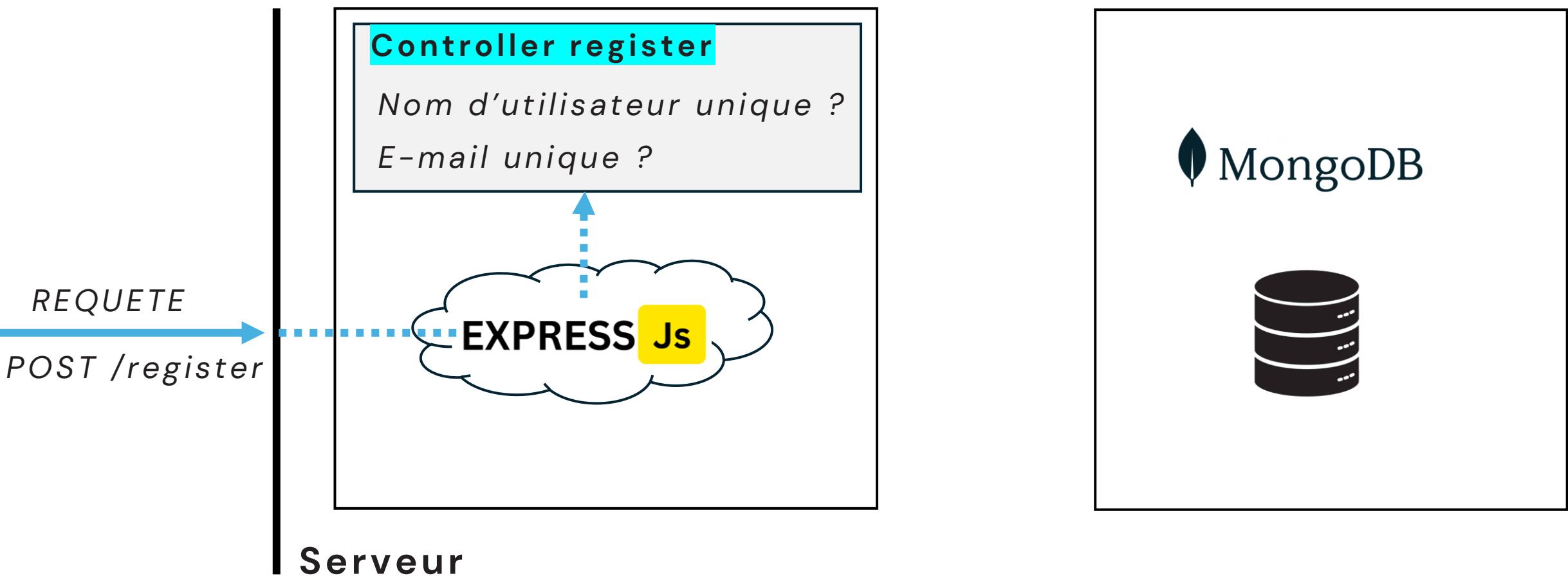
L>Backend 4





## > Création d'un compte

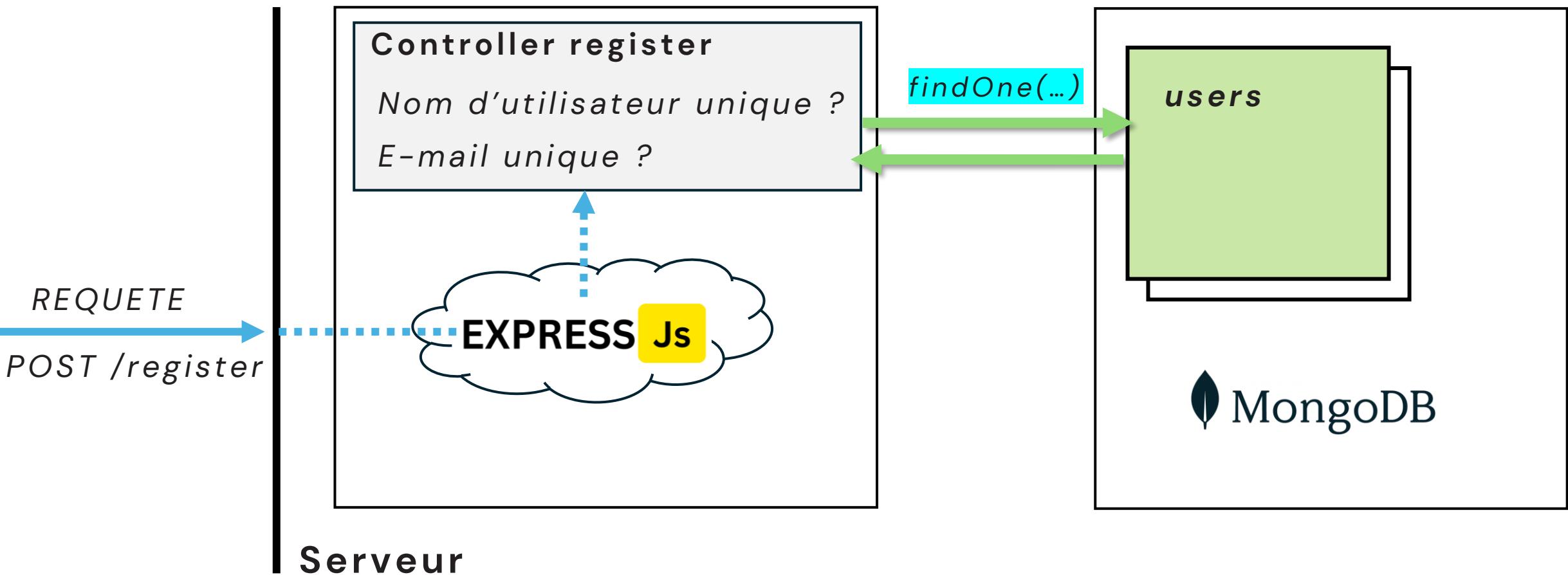
L>Backend 5





## > Création d'un compte

L>Backend 6

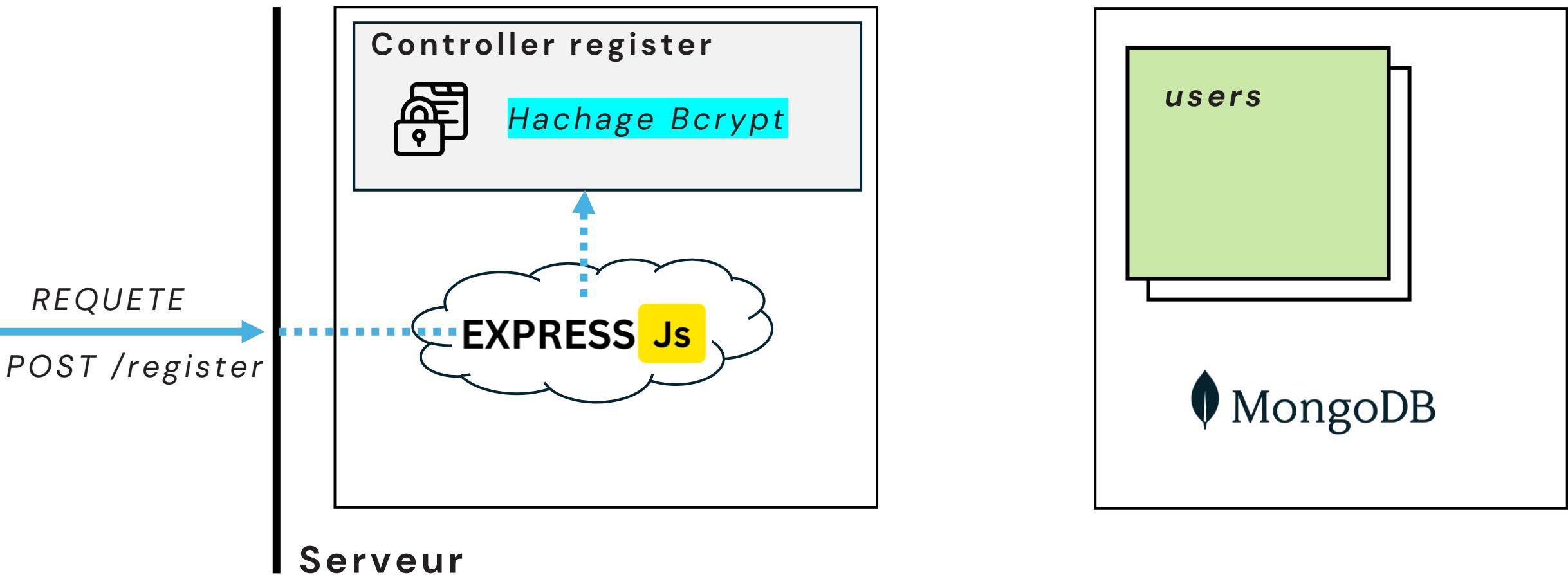




## > Création d'un compte

L>Backend

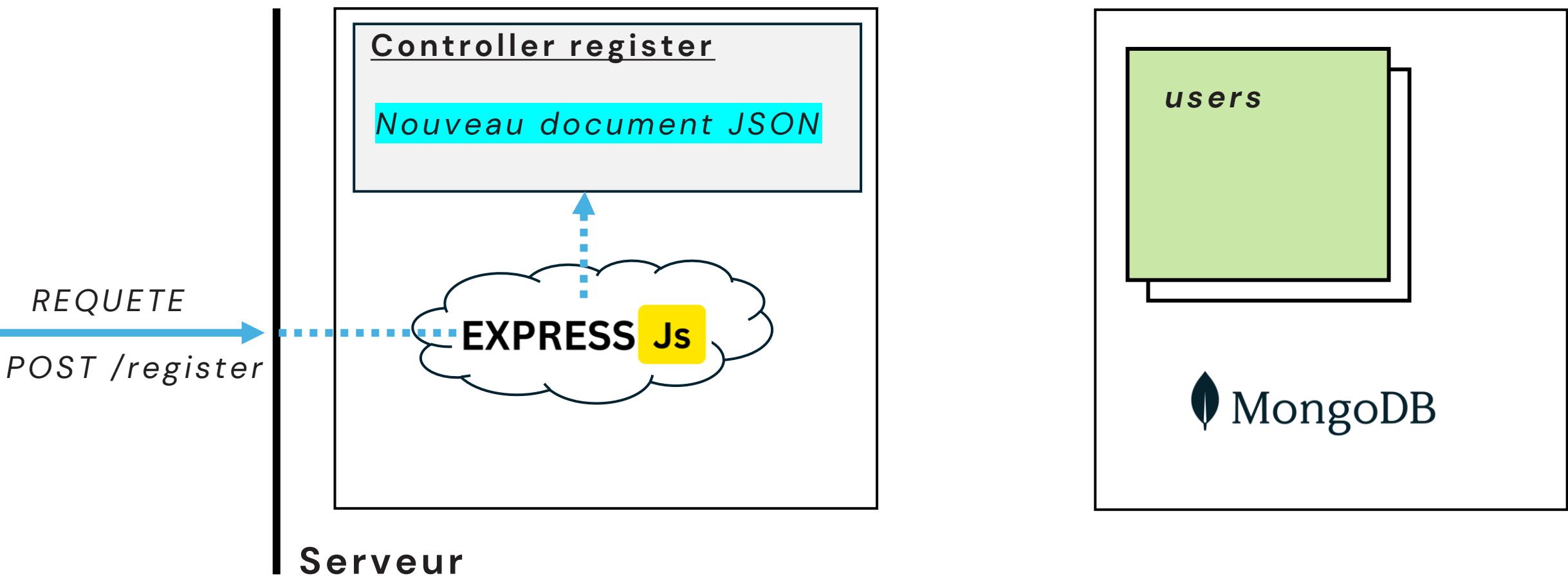
7





## > Création d'un compte

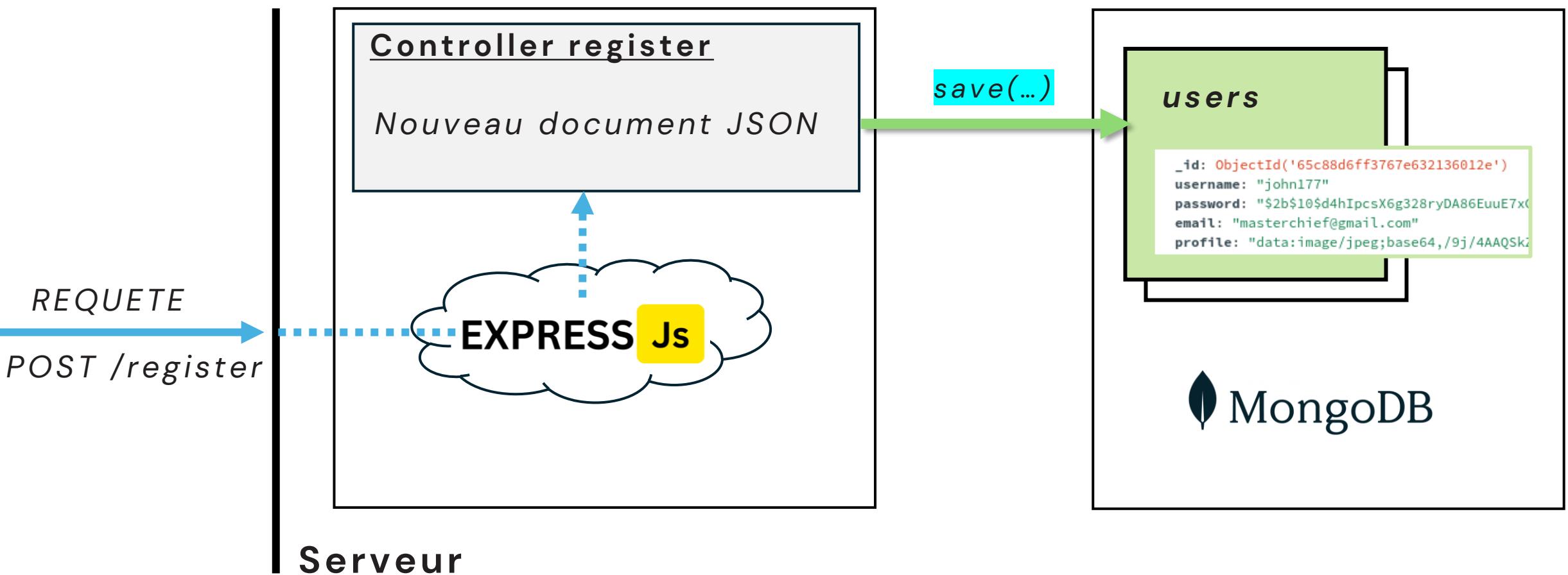
L>Backend 8





## > Création d'un compte

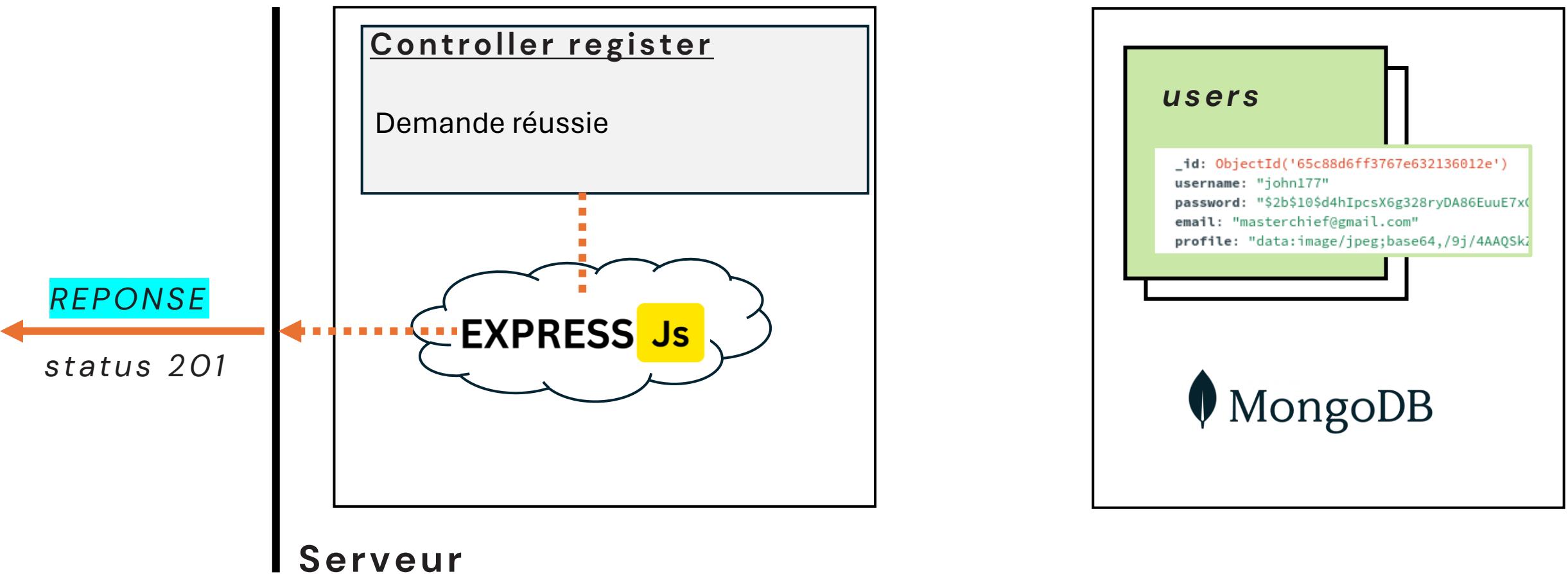
L>Backend 9





## > Création d'un compte

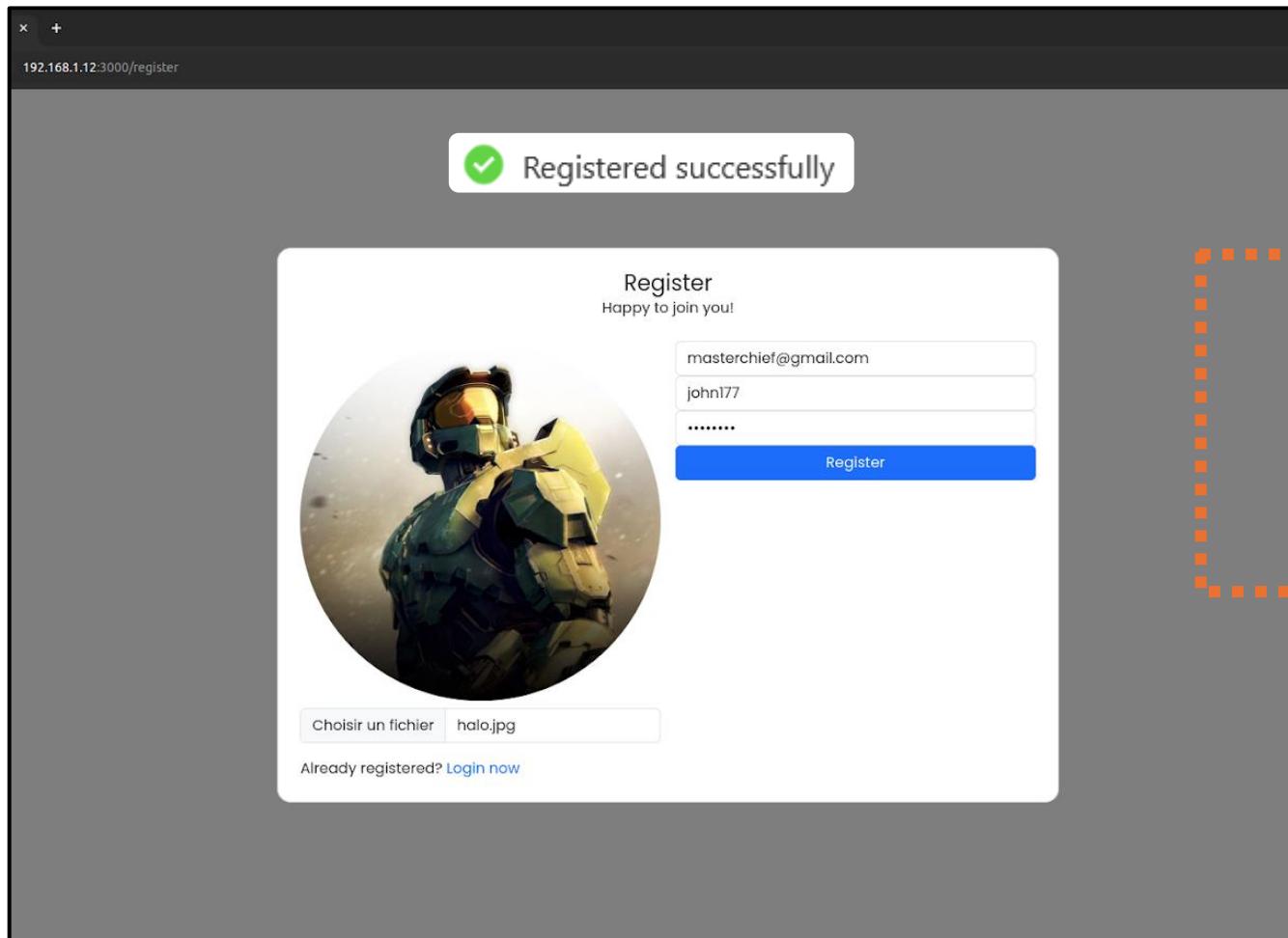
L>Backend 10





## > Crédit d'un compte

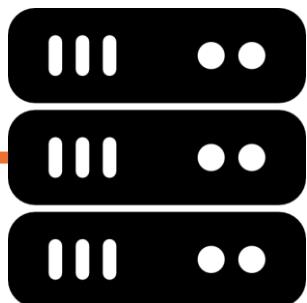
L>Frontend 11



Serveur

REONSE

status 201



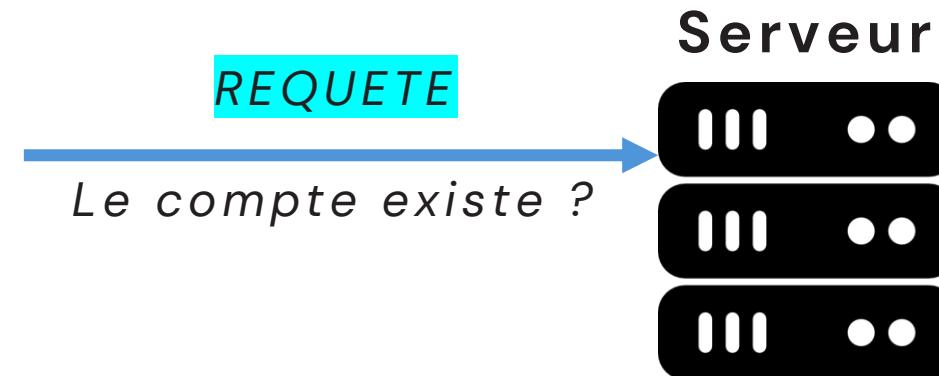
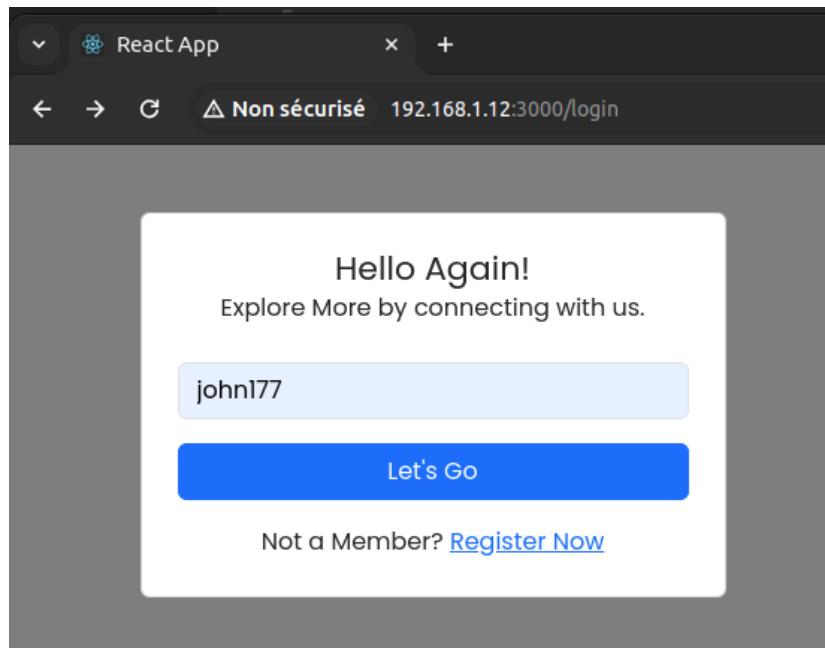
=> PAGE DE CONNEXION



## > Connexion à un compte existant

L>Frontend

Page de connexion 1/2

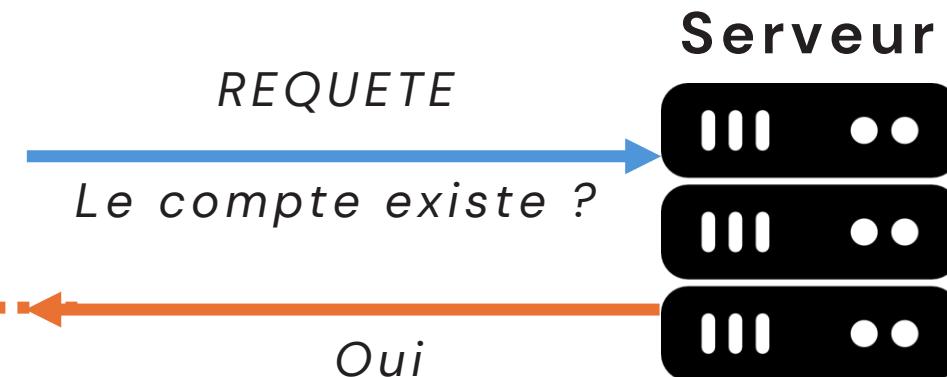
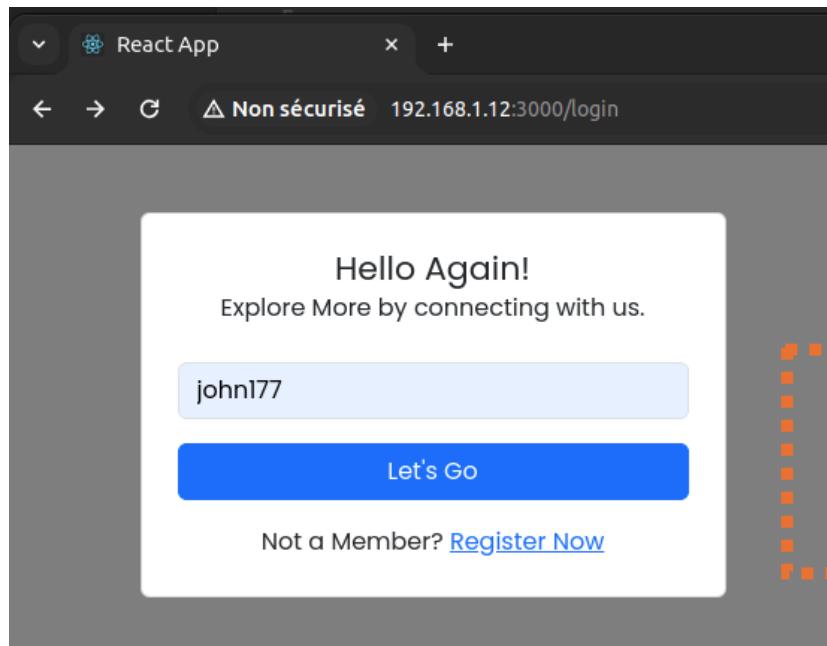




## > Connexion à un compte existant

L>Frontend

### Page de connexion 1/2



=> PAGE DE CONNEXION 2/2

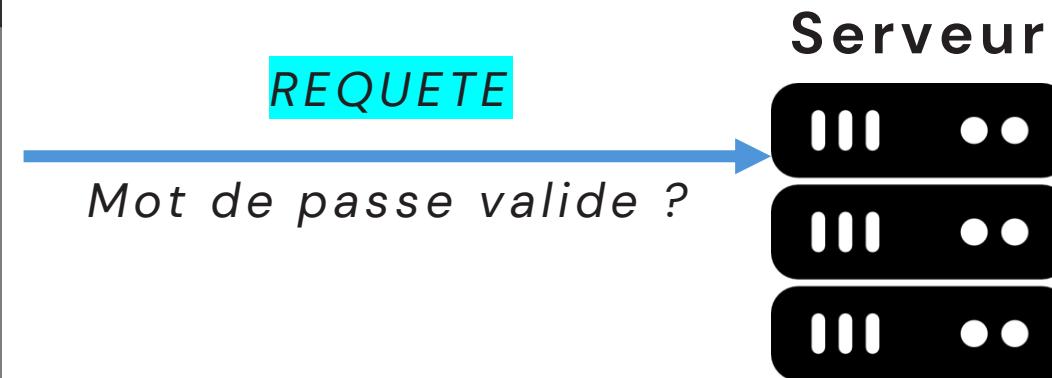
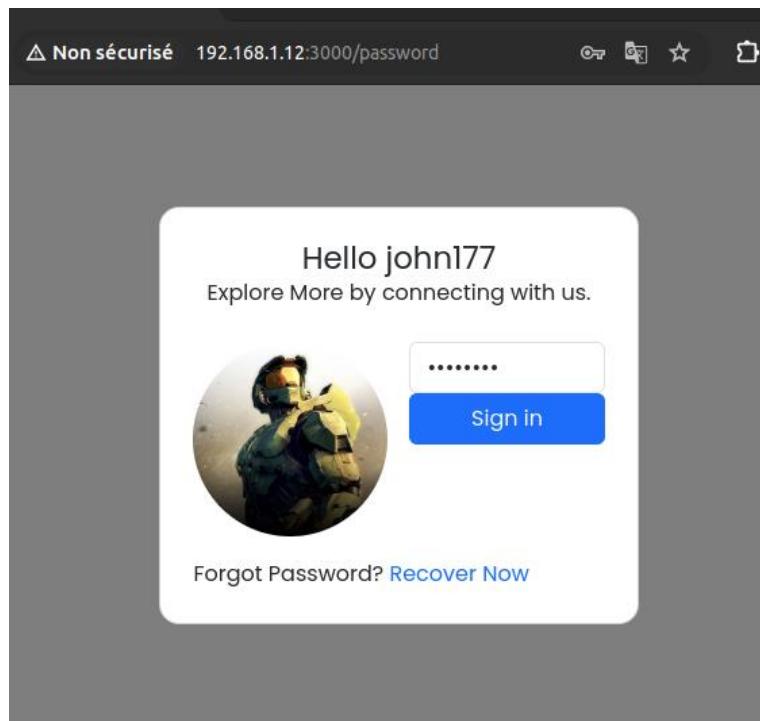
Mémorisation du nom d'utilisateur



## > Connexion à un compte existant

L>Frontend

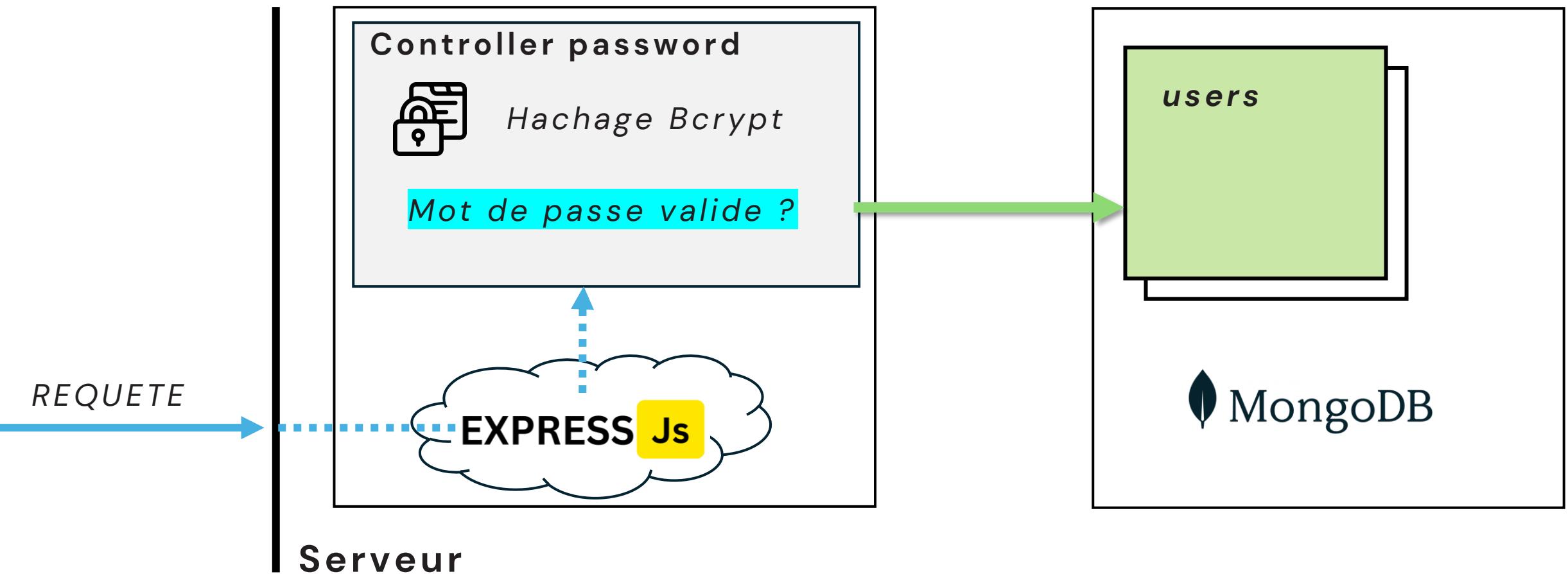
Page de connexion 2/2





## > Connexion à un compte existant

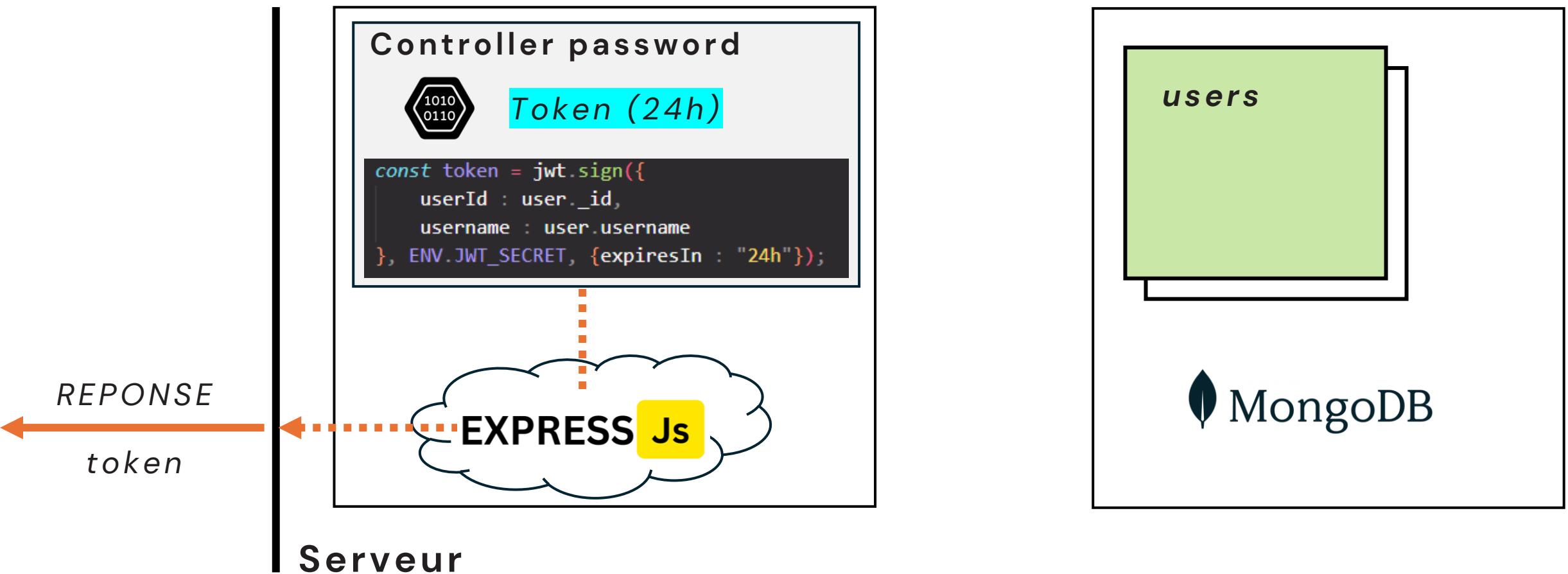
L>Backend





## > Connexion à un compte existant

L>Backend

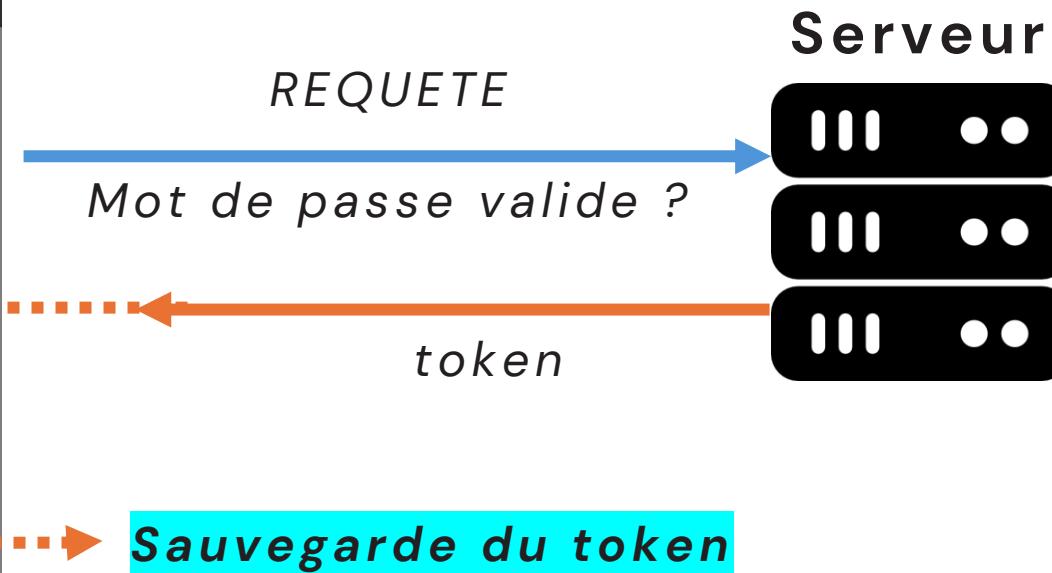
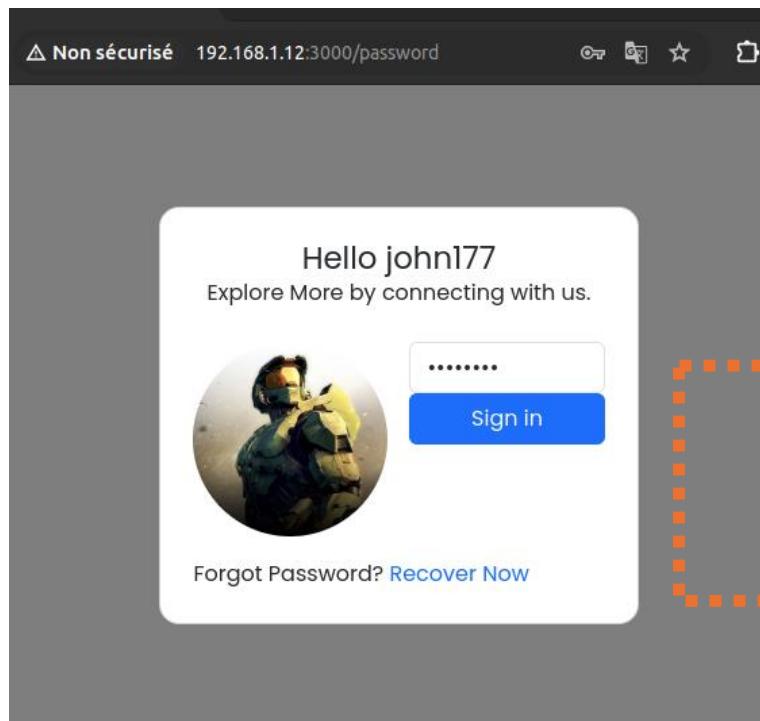




## > Connexion à un compte existant

L>Frontend

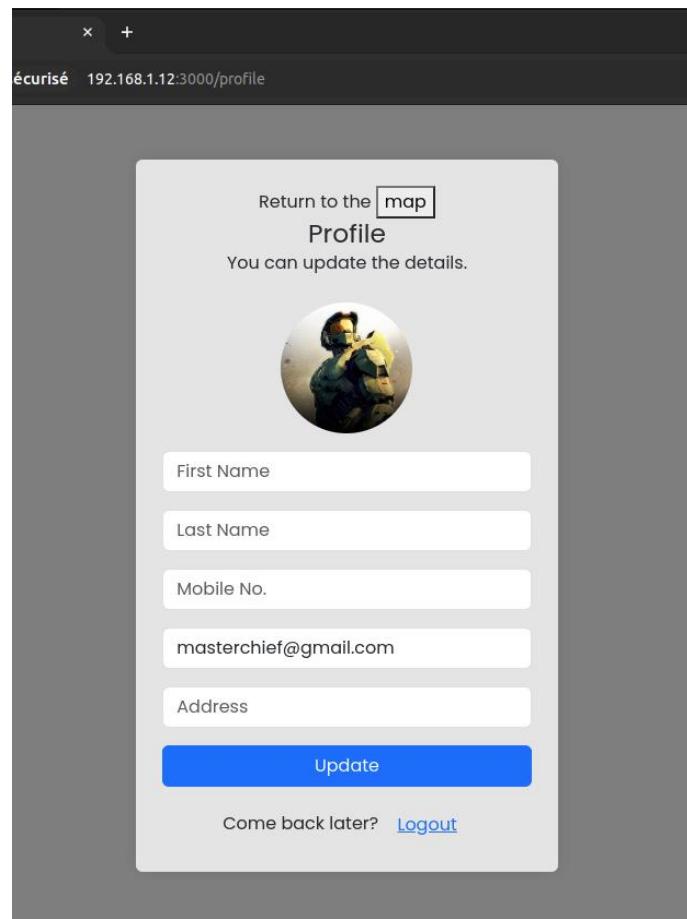
Page de connexion 2/2





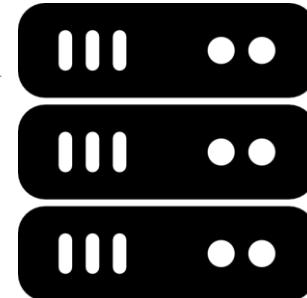
## > Modification du profile utilisateur

L>Frontend



REQUETE  
PUT

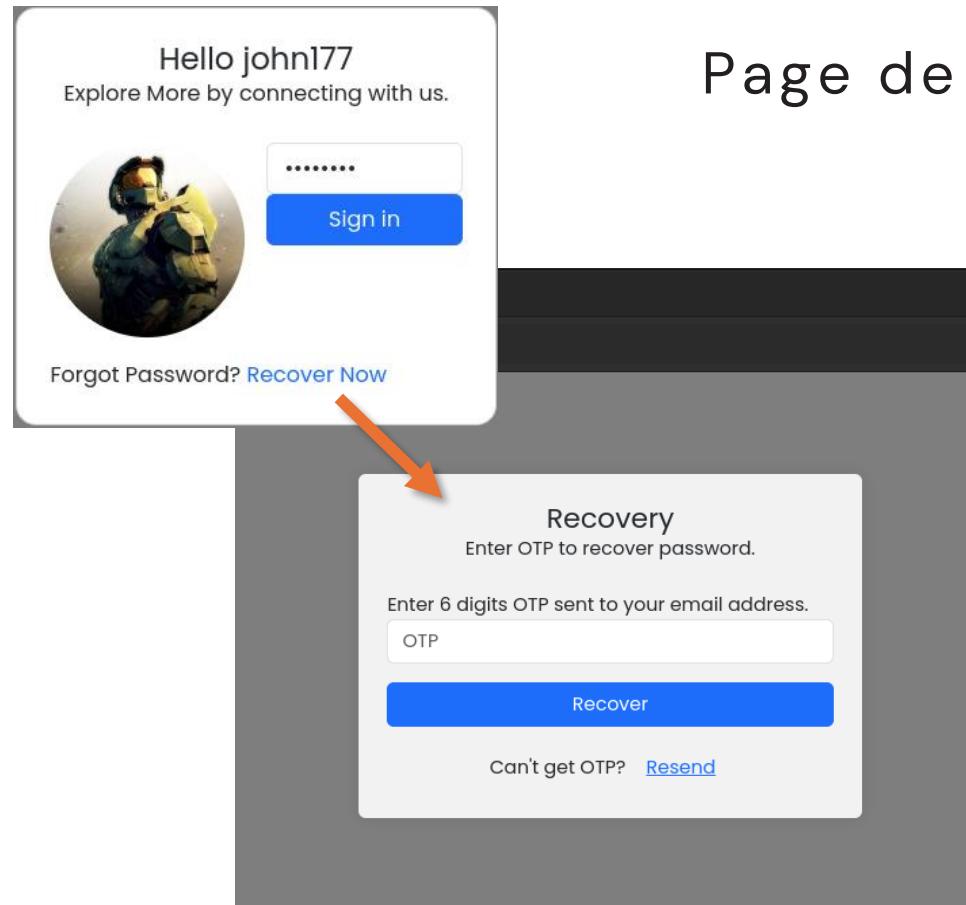
Serveur





## > Récupération de mot de passe

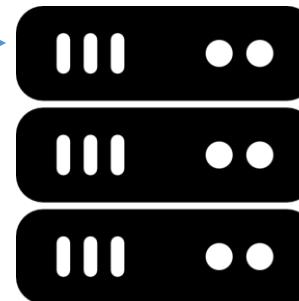
L>Frontend



Page de réinitialisation 1/2

REQUETE

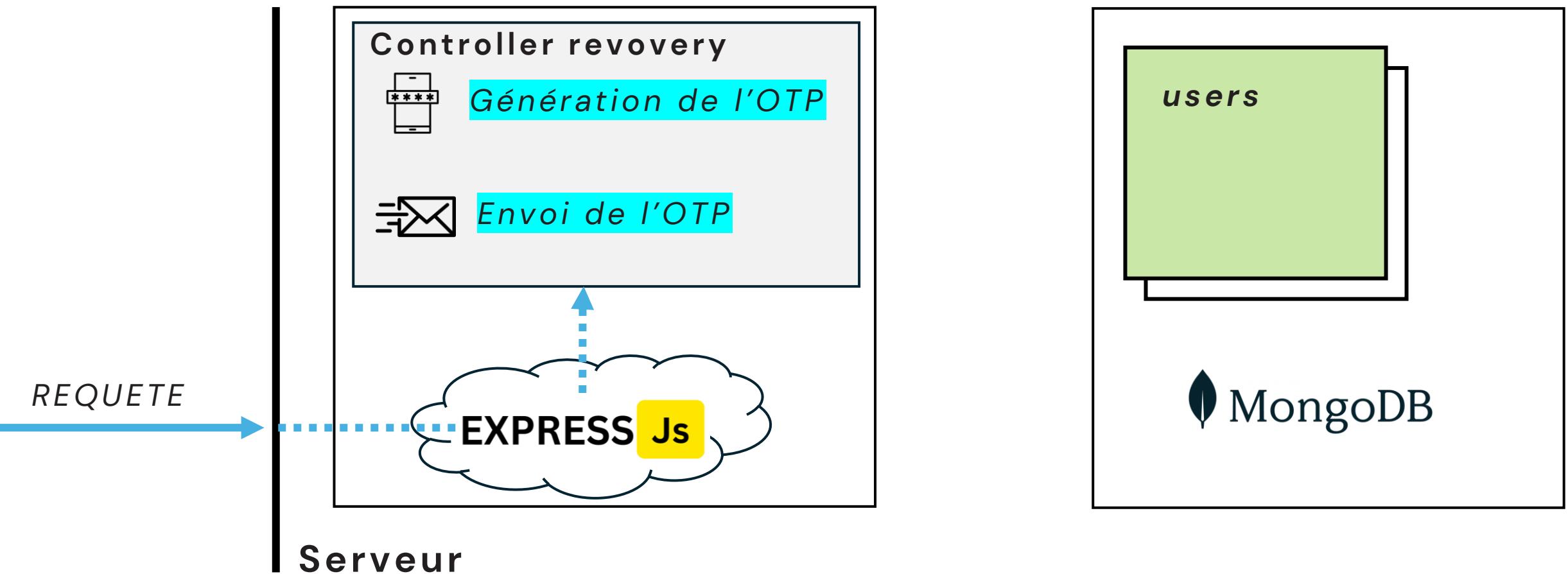
Serveur





## › Récupération de mot de passe

↳ Backend





## > Récupération de mot de passe

L>Boite mail

Ethereal Home FAQ Help Messages [Logout](#)

Headers Envelope Source

[🔗 Public URL of this message](#)

**Subject:** Password Recovery OTP

**From:** <jacques.miller31@ethereal.email>

**To:** <masterchief@gmail.com>

**Time:** Today at 10:12

**Message-ID:** <6278214b-68aa-3814-31ef-d317c43e5f91@ethereal.email>

HTML  Plaintext

Mailgen

Hi john177,

Your Password Recovery OTP is 479433. Verify and recover your password.

Need help, or have questions? Placeholder

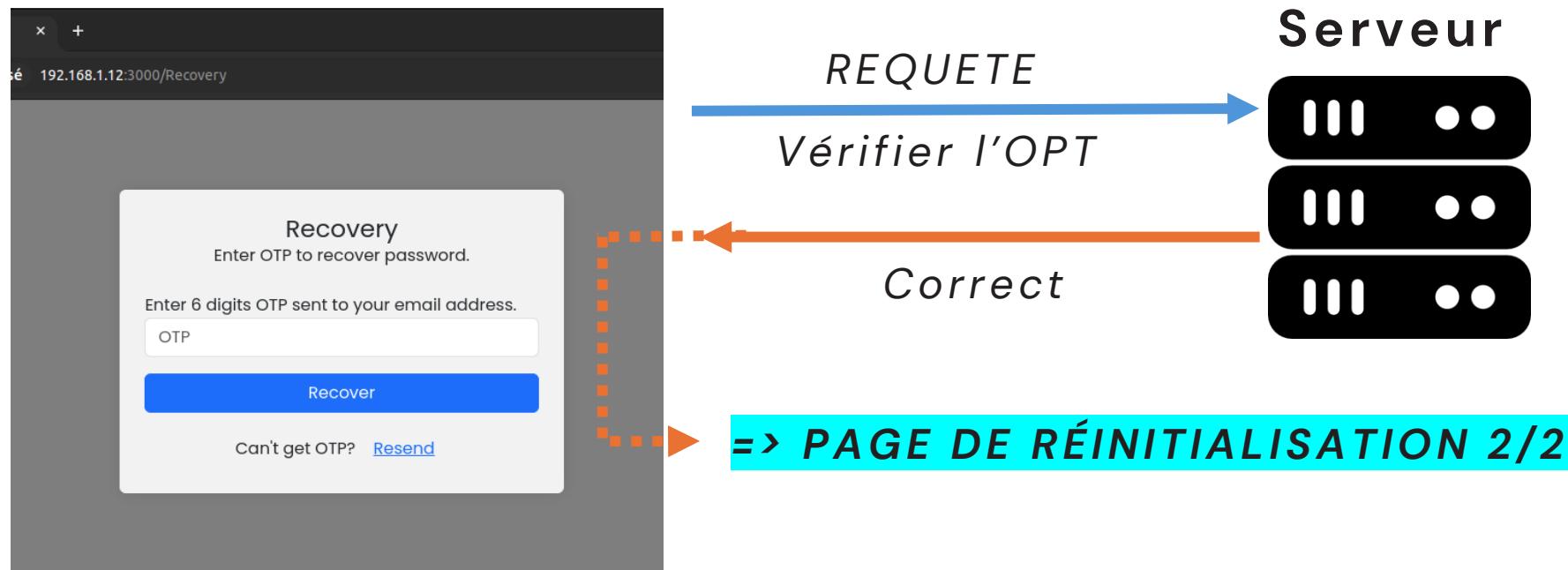
Yours truly,  
Mailgen



## > Récupération de mot de passe

L>Frontend

Page de réinitialisation 1/2

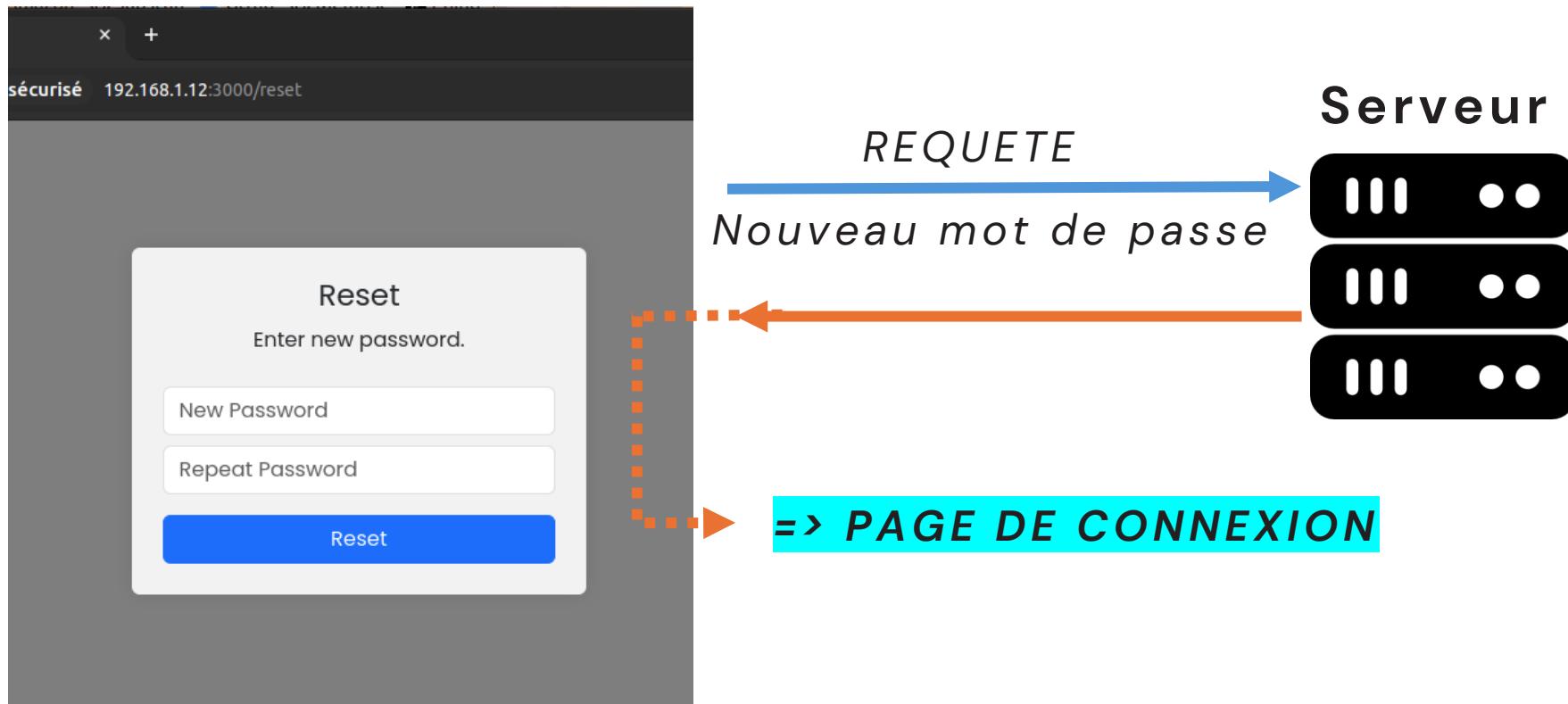


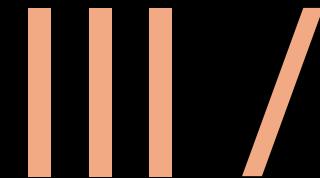


## > Récupération de mot de passe

L>Frontend

Page de réinitialisation 2/2

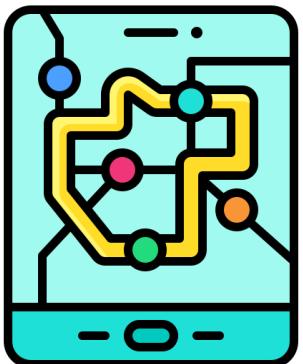




## Gestion des trajets



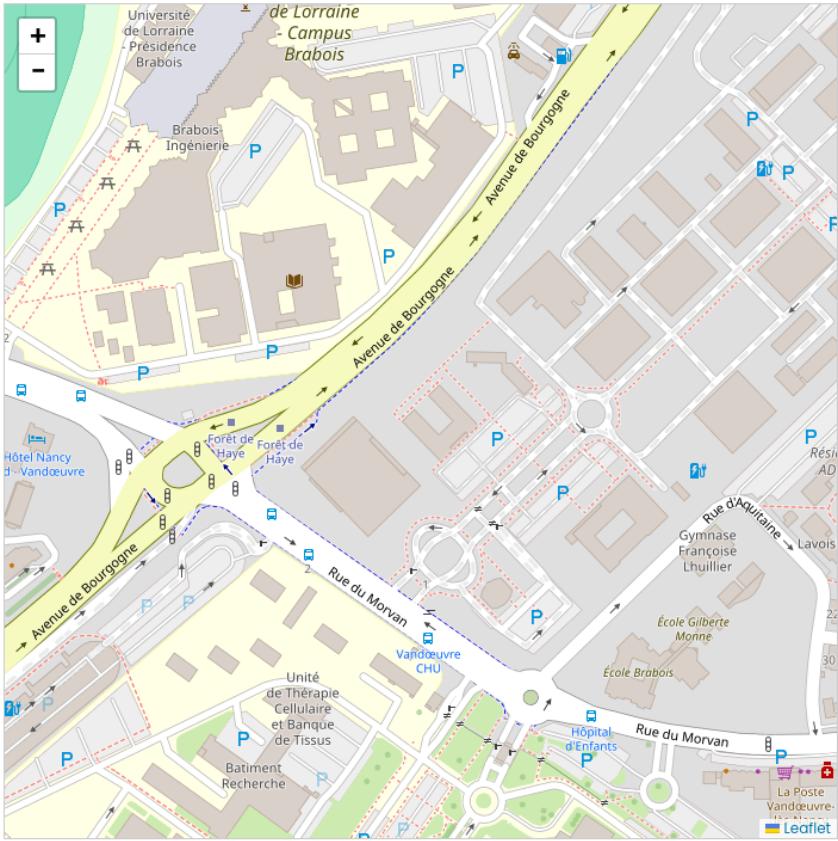
## › Fonctionnalités à intégrer



- Implémentation d'une carte
- Création de trajets
- Sauvegarde des trajets
- Chargement des trajets
- Ajout de la correspondance des trajets



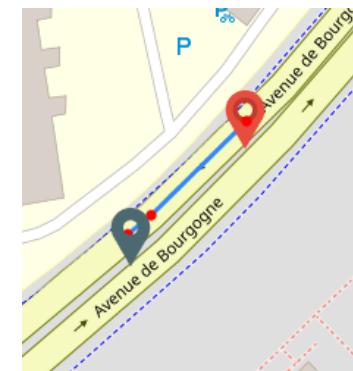
## > Implémentation d'une carte



**OpenStreetMap**



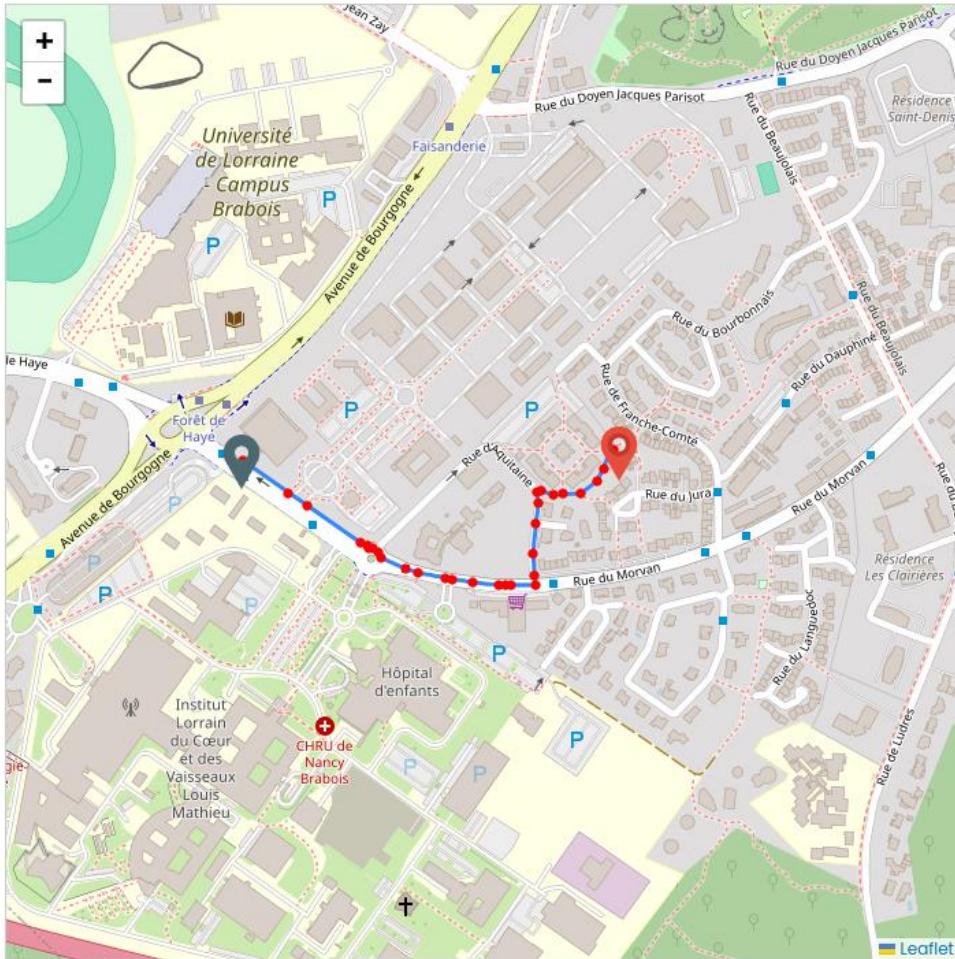
**React Leaflet**





## > Crédit de trajets

- Point de départ
- Point d'arrivée



## Fonctionnalités

- Glissement des points
- Mise à jour automatique
- Trajet personnalisable manuellement



## > Sauvegarde des trajets

### Trajet courant

Nom du chemin :  [ ]

Dates de départ :  [ + ]

- 06/02/2024 04:35 [ - ]
- 21/02/2024 04:25 [ - ]

Horaires périodiques :

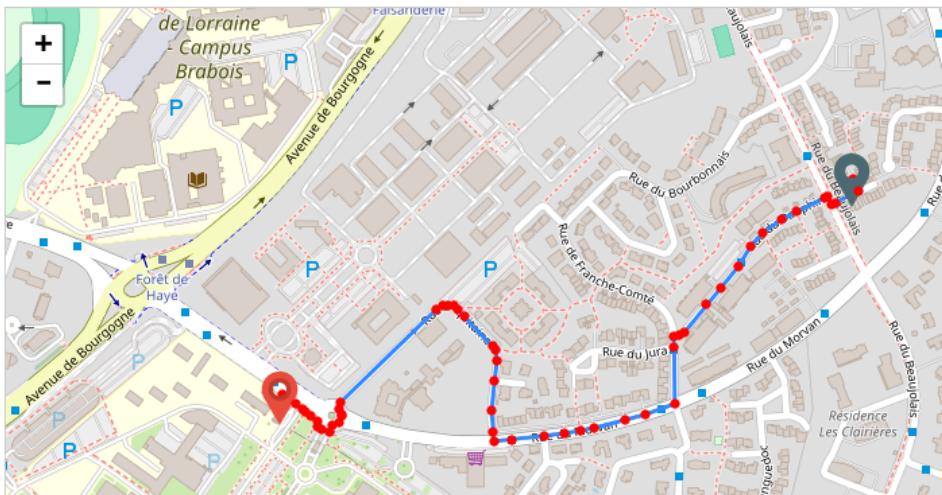
Jour de la semaine :  [ ▾ ]

Heure :  [ X ] [ + ]

- Lundi 06:30 [ - ]
- Vendredi 14:45 [ - ]

**Créer trajet**

- Point de départ  
 Point d'arrivée



### Fonctionnalités

- Ajout de dates ponctuelles
- Ajout de dates périodiques
- Sélection de dates ergonomique



## > Sauvegarde des trajets

### Trajet courant

Nom du chemin :  [+]

Dates de départ :  [+]

- 06/02/2024 04:35 [+]
- 21/02/2024 04:25 [+]

Horaires périodiques :

Jour de la semaine :  [+]

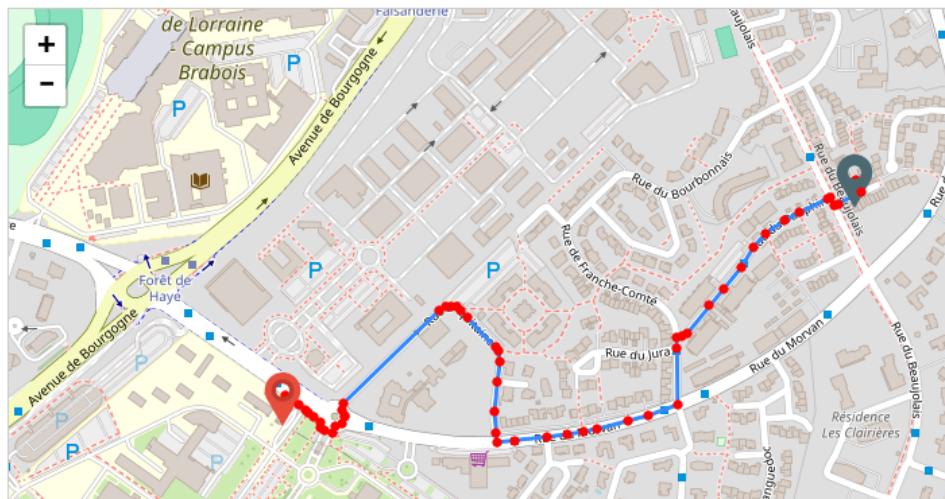
Heure :  x [+]

- Lundi 06:30 [+]
- Vendredi 14:45 [+]

**Créer trajet**

Point de départ

Point d'arrivée

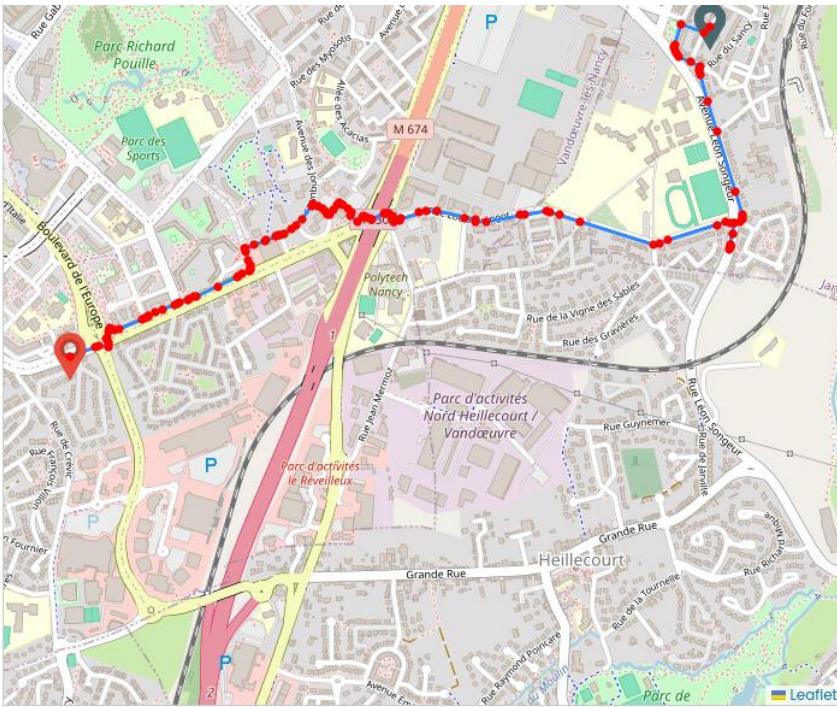


✓ Route added

```
_id: ObjectId('65c991ab6958c07dc21326fc')
username: "john177"
name: "Trajet_1"
route: Array (53)
  planning: Object
    dates: Array (2)
      0: 2024-02-06T03:35:00.000+00:00
      1: 2024-02-21T03:25:00.000+00:00
periodic: Array (2)
```



## > Chargement des trajets



### Mes Trajets

• 05/02/2024 04:33

◦ 21/02/2024 04:25

Horaires hebdomadaires :

◦ Lundi 06:30

◦ Vendredi 14:45

[Supprimer](#)

• Trajet\_2

Dates :

◦ 29/01/2024 04:40

Horaires hebdomadaires :

[Supprimer](#)

### Trajet courant

Nom du chemin :

Dates de départ :  [+]

• 29/01/2024 04:40 [-]

Horaires périodiques :

Jour de la semaine :  [▼]

Heure :  [+]

[Créer trajet](#) [Modifier le trajet](#)

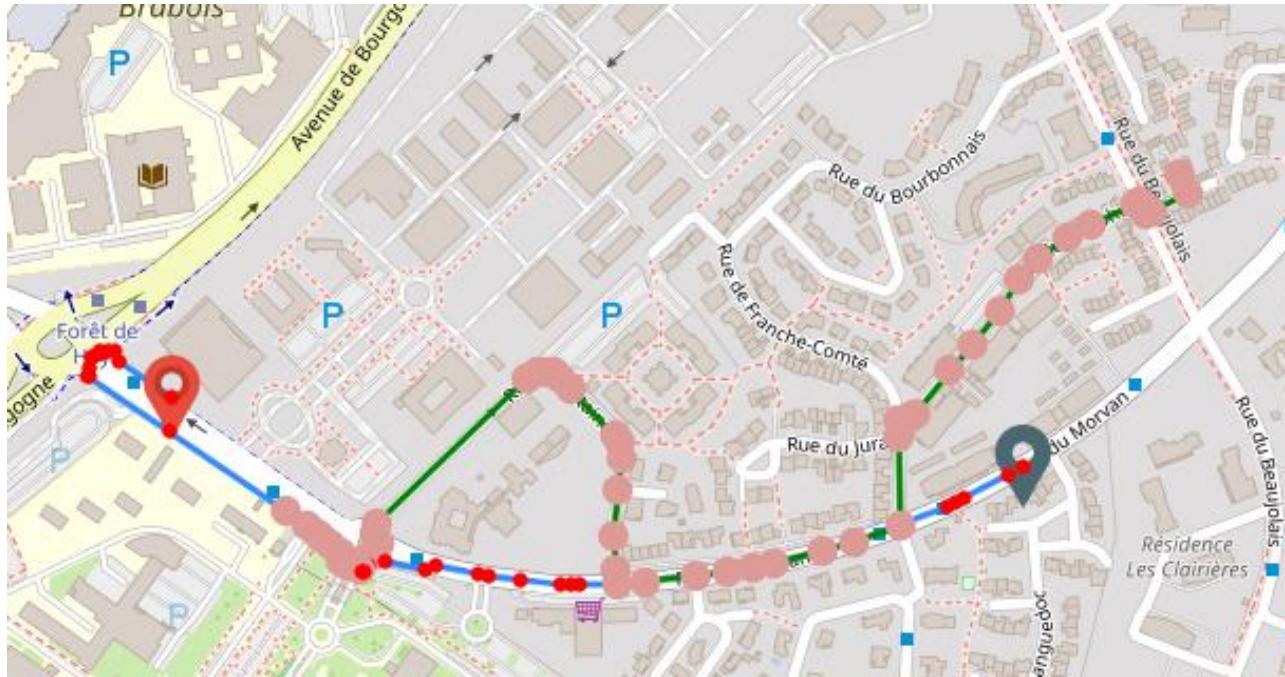
- Remplissage des champs automatique
- Mise à jour du trajet
- Création d'un nouveau trajet



## > Ajout de la correspondance des trajets

L>Travail partiellement terminé

=>Correspondance uniquement par date

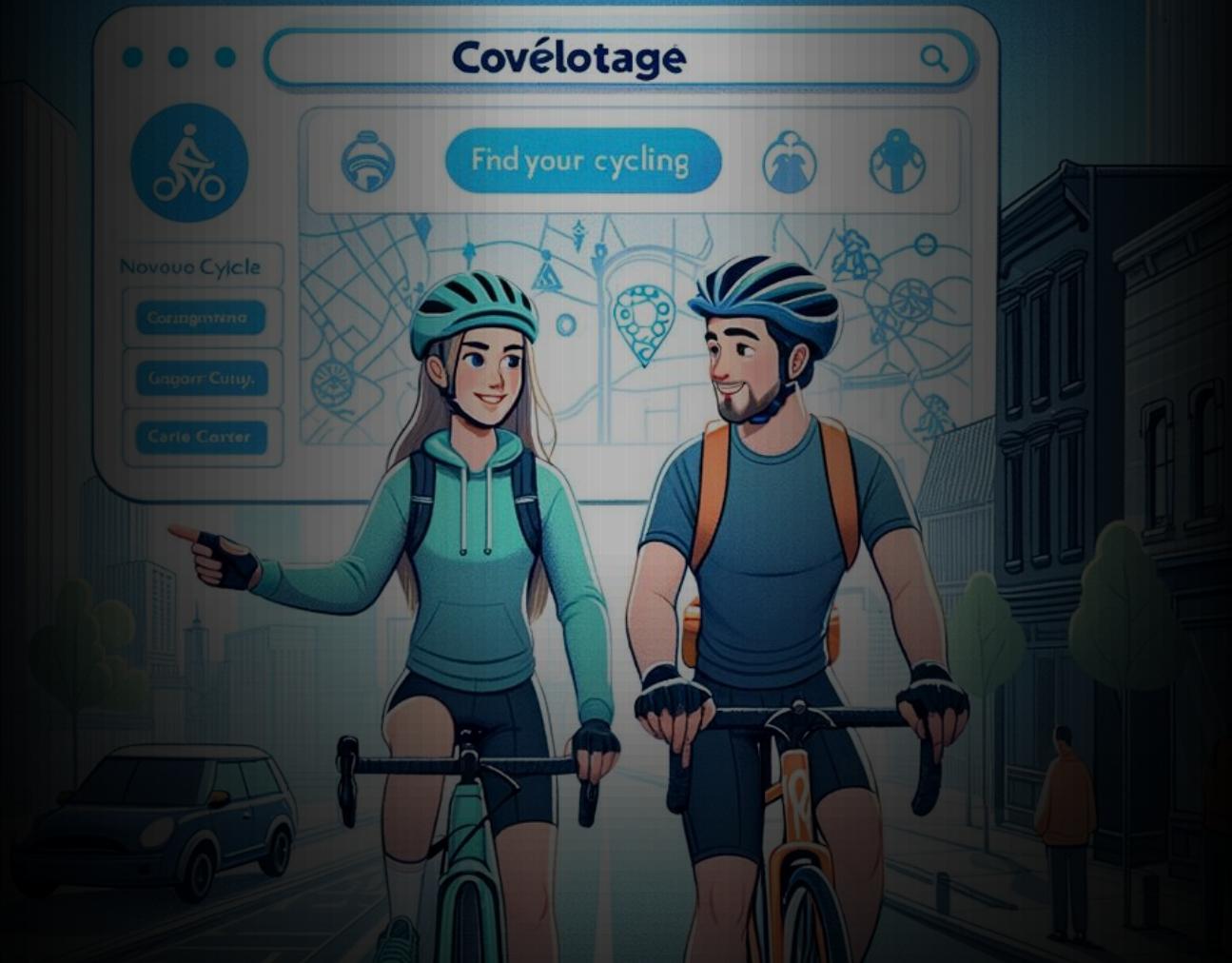


### Trajets correspondant

- 06/02/2024 22:30  
Horaires hebdomadaires :
  - Dimanche 18:27
  - Mardi 10:40

- Trajet\_1

- Dates :
  - 06/02/2024 04:35
  - 21/02/2024 04:25
- Horaires hebdomadaires :
  - Lundi 06:30
  - Vendredi 14:45



# CONCLUSION

Démonstration

DFM