

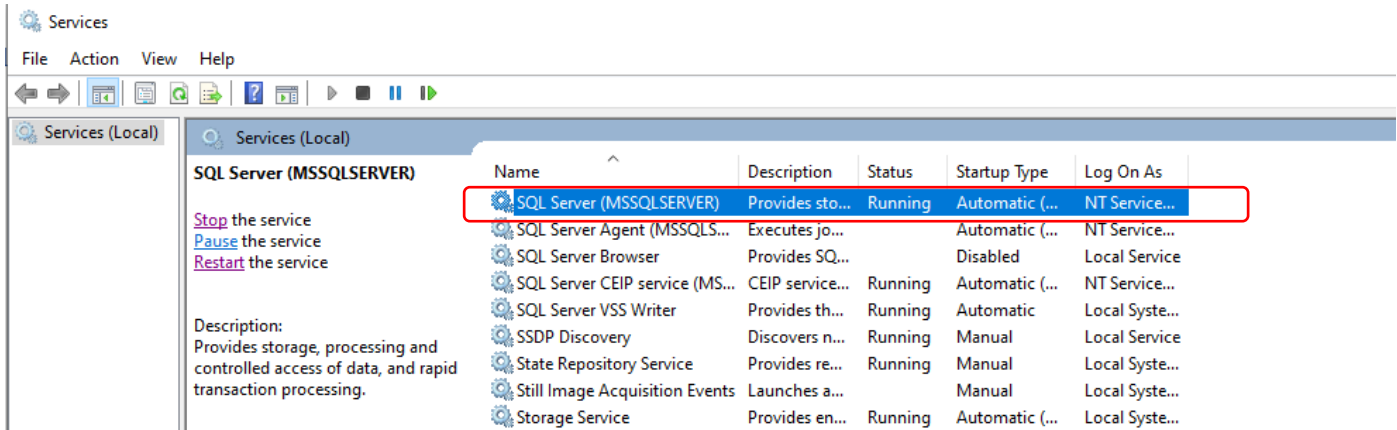
# PROJET FIL ROUGE – DEVELOPPER LES COMPOSANTS D'ACCES AUX DONNEES

## 1. Contenu du fichier .zip

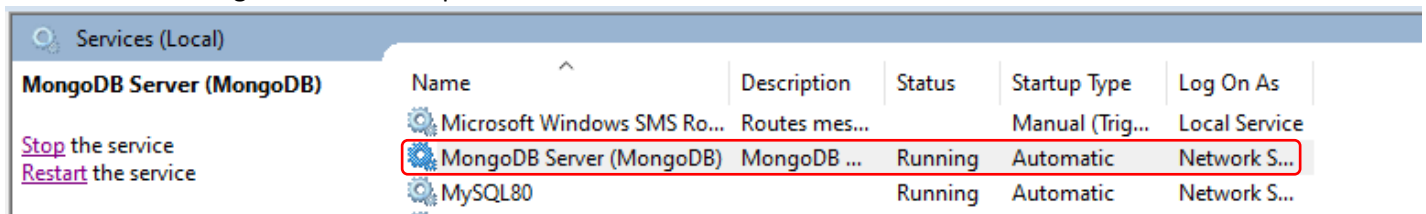
- 'cahier\_des\_charges.pdf' → travail à réaliser
- 'create\_easyloc.sql' → fichier de script de création de la base de données 'easyLoc' à lancer depuis un terminal
- 'ReadMe.pdf' → ce document

## 2. Pré-requis

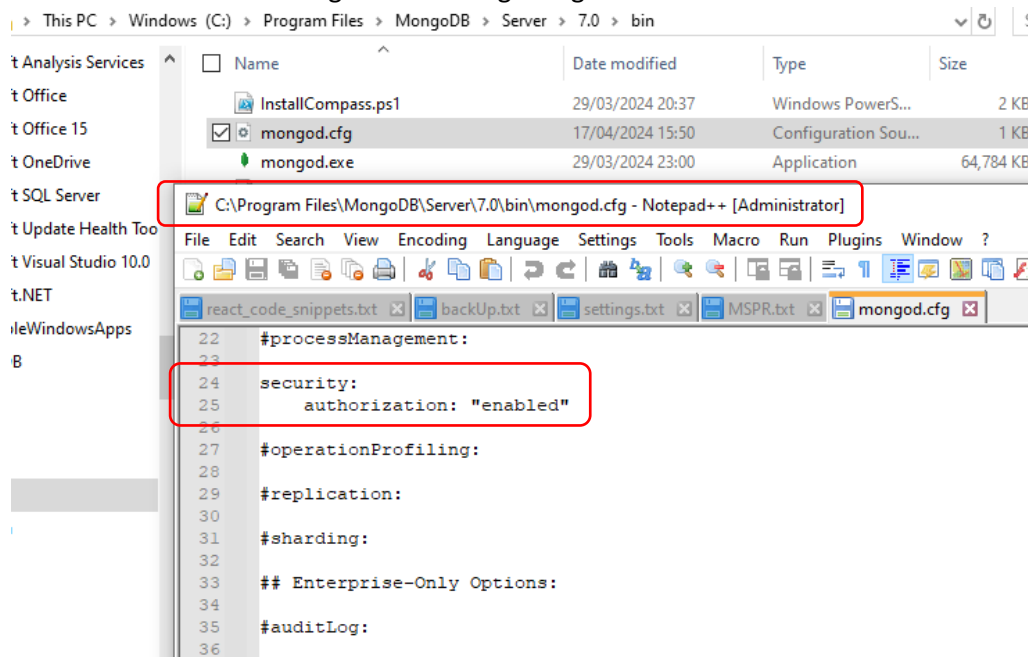
- MS SQL Server installé et opérationnel en mode 'localhost'



- MongoDB installé et opérationnel en mode 'localhost'



- Mongosh (utilitaire de commande en ligne pour MongoDB) installé
- Modifier le fichier de configuration 'mongod.cfg'



## 3. Création de la base de données SQL Server

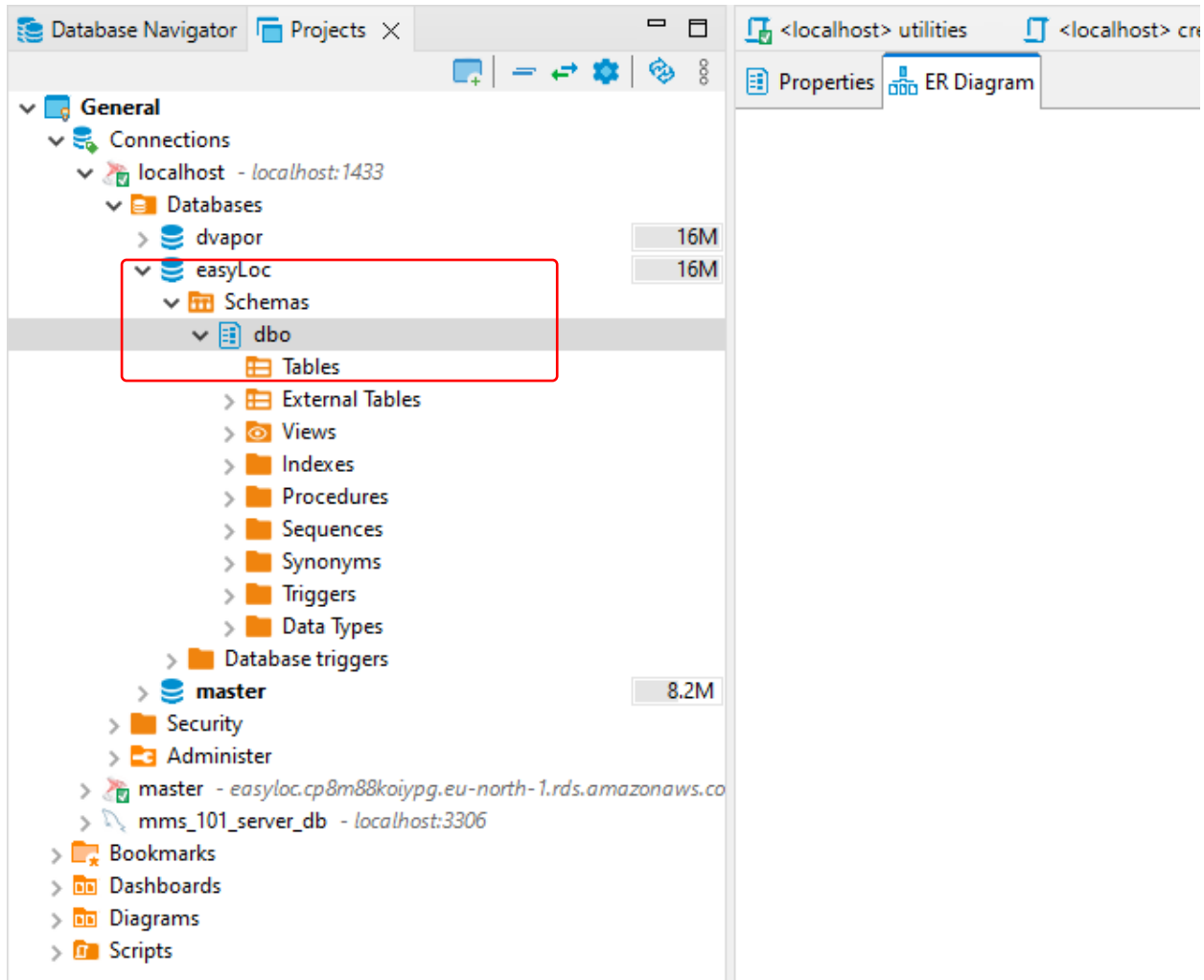
- Décompresser le fichier .zip et placer le fichier 'create\_easyLoc.sql' sur le bureau

## PROJET FIL ROUGE – DEVELOPPER LES COMPOSANTS D'ACCES AUX DONNEES

- Ouvrir un terminal et lancer le script en prenant soin d'indiquer le chemin vers le fichier de script

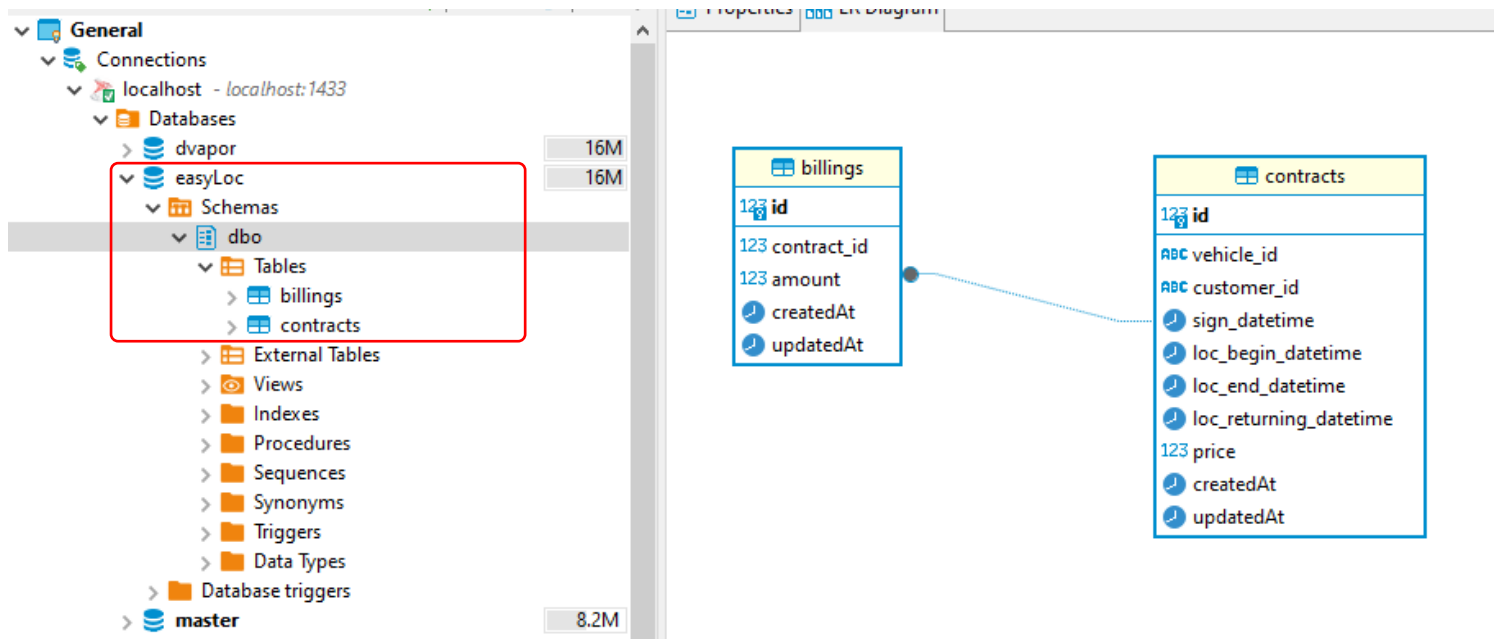
```
C:\Users\henri>sqlcmd -S localhost -i C:\Users\henri\Desktop\create_easyLoc.sql
Changed database context to 'master'.
Changed database context to 'easyLoc'.
```

- Une base de données vide a été créée ainsi que l'utilisateur 'user1' mot de passe 'Epsi2024'



- Les tables 'contracts' et 'billings' ainsi que la relation 'one to many' entre les 2 tables seront créées par les modèles de l'API dès son lancement (voir § 5)

# PROJET FIL ROUGE – DEVELOPPER LES COMPOSANTS D'ACCES AUX DONNEES



## 4. Création de la base de données MongoDB

- Ouvrir un Command Prompt en ligne de commande

```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\henri>mongosh
Current Mongosh Log ID: 661f9a73762ff232d9117b7a
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.4
Using MongoDB:      7.0.8
Using Mongosh:       2.2.4

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

-----
The server generated these startup warnings when booting
2024-04-17T10:15:33.268+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test>
```

- Se connecter sur la base 'admin' (système) et créer l'utilisateur 'root' mot de passe 'root' avec le rôle 'root' → super utilisateur avec tous les privilèges

```
easyLoc> use admin
switched to db admin

admin> db.createUser({user:'root',pwd:'root',roles:[{role:'root',db:'admin'}]})
{ ok: 1 }
```

- S'authentifier en tant que 'root'

## PROJET FIL ROUGE – DEVELOPPER LES COMPOSANTS D'ACCES AUX DONNEES

```
admin> db.auth('root','root')
{ ok: 1 }
admin> db.runCommand({connectionStatus:1})
{
  authInfo: {
    authenticatedUsers: [ { user: 'root', db: 'admin' } ],
    authenticatedUserRoles: [ { role: 'root', db: 'admin' } ]
  },
  ok: 1
}
admin>
```

- Créer la base de données 'easyLoc' et les collections 'Contracts' et 'Billings'

```
admin> use easyLoc
switched to db easyLoc
easyLoc> db.createCollection('Contracts')
{ ok: 1 }
easyLoc> db.createCollection('Billings')
{ ok: 1 }
easyLoc>
```

- Créer l'utilisateur 'user1' avec le mot de passe 'Epsi2024'

```
easyLoc> db.createUser({user:'user1',pwd:'Epsi2024',roles:[{role:'readWrite',
db:'easyLoc'},{role:'dbAdmin',db:'easyLoc'}]})
{ ok: 1 }
```

- Les modèles Customer et Vehicle seront utilisés lors de chaque requête vers l'API

MongoDB Compass - easyLoc\_user1/easyLoc.customers

Connect Edit View Collection Help

easyLoc\_user1

My Queries Performance Databases Search

easyLoc

customers vehicles

My Queries customers

easyLoc > customers

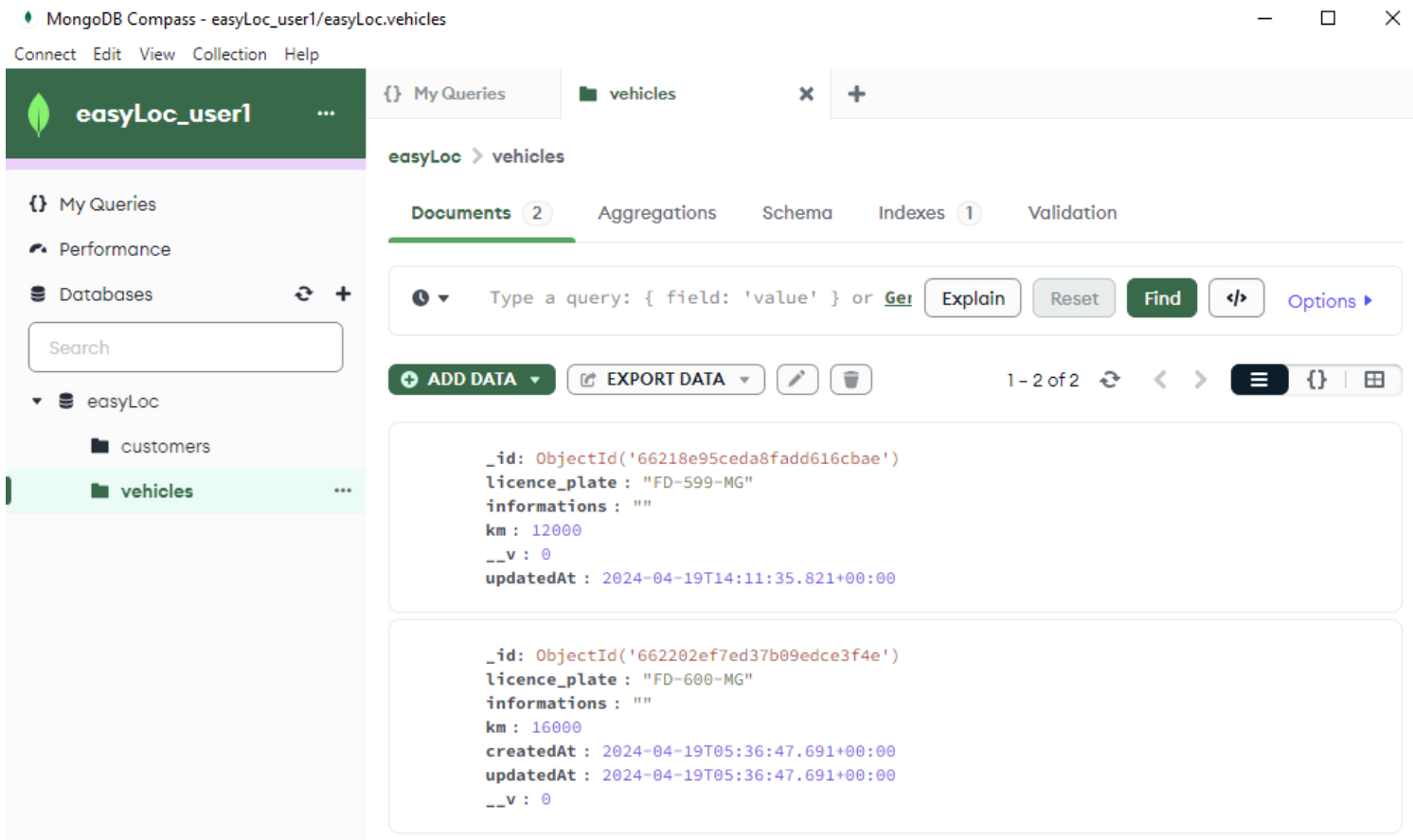
Documents 1 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Get](#) [Explain](#) [Reset](#) [Find](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#) 1-1 of 1

```
{
  "_id": ObjectId("6628cfff987ba4b7dbb2f937"),
  "first_name": "Henri",
  "last_name": "MARTIN",
  "address": "4, impasse du pigeonier - 75015 Paris",
  "permit_number": "451228"
}
```

# PROJET FIL ROUGE – DEVELOPPER LES COMPOSANTS D'ACCES AUX DONNEES



## 5. Lancement du serveur en mode local


- Récupérer les fichiers source sur le répertoire GitHub à l'adresse [https://github.com/talfumier/easyLoc\\_api](https://github.com/talfumier/easyLoc_api) sous forme de fichier .zip
- Décompresser le fichier .zip dans un dossier local
- Ouvrir un terminal en ligne de commande et se rendre dans le dossier 'src' puis lancer la commande 'node index.js'

```
Command Prompt - node index.js
(c) Microsoft Corporation. All rights reserved.
C:\Users\henri>cd C:\Users\henri\Documents\WebDevProjects\EPSI_projects\easyLoc_api\src
C:\Users\henri\Documents\WebDevProjects\EPSI_projects\easyLoc_api\src>node index.js
(node:6648) ExperimentalWarning: Importing JSON modules is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
Express server is listening at http://localhost:8000
[API]: successfully connected to MongoDB server !
[API]: successfully connected to MS SQL Server !
SqlServer tables and models successfully synced !
```


- Le serveur est maintenant opérationnel et prêt à recevoir des requêtes sur <http://localhost:8000>

6. Déploiement de l’API en production

- Les bases de données ont été déployées sur AWS RDS pour la base SQL et MongoDB Atlas pour la base NoSQL
- L’API globale est sur Render et peut être utilisée comme dans le mode local en utilisant le domaine <https://easyloc-api.onrender.com/>
- Il peut y avoir un certain temps de latence à la première connexion car les serveurs sont mis au repos en cas d’inactivité
- Un jeu de données minimal a été créé et est disponible en ligne



DashboardBlueprintsEnv GroupsDocsCommunityHelp

New +

WEB SERVICE

easyLoc\_api

NodeFreeUpgrade your instance →


Connexion

talumier / easyLoc\_api

dev

<https://easyloc-api.onrender.com>

Events



Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. [Upgrade now.](#)

7. Fonctionnalités de l’API

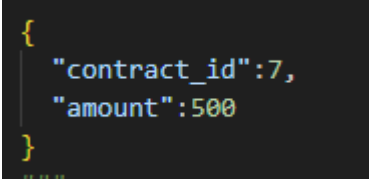
- Pour accéder aux fonctionnalités et aux données, il suffit de créer des requêtes http ayant pour url le domaine et le chemin correspondant: <http://localhost:8000/> + **chemin** en mode local et : <https://easyloc-api.onrender.com/> + **chemin** pour la version de production
- Les requêtes de type POST ou PATCH nécessitent l’envoi dans le corps de la requête, d’un objet au format JSON correspondant au modèle de données (complet ou partiel)

Modèle	Chemin	Type	Résultats
Customer	/api/customers	GET	<div>– Liste de tous les clients</div> <div>– Possibilité d’ajouter des paramètres dans l’url pour modifier les critères de filtre et/ou de tri : <a href="#">/api/customers?first_name=/^henri\$/i&amp;sort=last_name</a> → asc <a href="#">/api/customers?first_name=/^henri\$/i&amp;sort=-last_name</a> → desc</div>
	/api/customers	POST	<div>– Création d’un nouveau client</div> <div>– Client à créer doit être fourni dans le corps de la requête au format JSON</div> <div><pre>{   "first_name": "xxxx",   "last_name": "yyyyyy",   "address": "zzzzzz",   "permit_number": "4aaaaa" }</pre></div>

# PROJET FIL ROUGE – DEVELOPPER LES COMPOSANTS D'ACCES AUX DONNEES

Modèle	Chemin	Type	Résultats
	/api/customers/:id	PATCH	<ul style="list-style-type: none"> <li>– Modification d'un client existant</li> <li>– Le ou les champs à modifier à fournir dans le corps de la requête au format JSON</li> <li>– id du client à mettre dans l'url <a href="/api/customers/66212a1b33d48a71897191f6">/api/customers/66212a1b33d48a71897191f6</a></li> </ul>
	/api/customers/:last_name/:first_name	GET	<ul style="list-style-type: none"> <li>– Rechercher un client à partir de son nom et prénom (paramètres à mettre dans l'url)</li> </ul>
	/api/customers/:id	DELETE	<ul style="list-style-type: none"> <li>– Supprimer un client</li> <li>– id du client à mettre dans l'url <a href="/api/customers/66212a1b33d48a71897191f6">/api/customers/66212a1b33d48a71897191f6</a></li> </ul>
Vehicle	/api/vehicles	GET	<ul style="list-style-type: none"> <li>– Liste de tous les véhicules</li> <li>– Possibilité d'ajouter des paramètres dans l'url pour modifier les critères de filtre et/ou de tri : <a href="/api/vehicles?km&lt;=86000&amp;km&gt;=16000&amp;sort=km">/api/vehicles?km&lt;=86000&amp;km&gt;=16000&amp;sort=km</a></li> </ul>
	/api/vehicles	POST	<ul style="list-style-type: none"> <li>– Création d'un nouveau véhicule</li> <li>– Véhicule à créer doit être fourni dans le corps de la requête au format JSON</li> </ul> <pre>{   "licence_plate": "FD-600-MG",   "km": 16000 }</pre>
	/api/vehicles/:id	PATCH	<ul style="list-style-type: none"> <li>– Modification d'un client existant</li> <li>– Le ou les champs à modifier à fournir dans le corps de la requête au format JSON</li> <li>– id du client à mettre dans l'url <a href="/api/vehicles/66212a1b33d48a71897191f6">/api/vehicles/66212a1b33d48a71897191f6</a></li> </ul>
	/api/vehicles/:licence_plate	GET	<ul style="list-style-type: none"> <li>– Rechercher un véhicule à partir de sa plaque d'immatriculation (paramètre à mettre dans l'url)</li> </ul>
	/api/vehicles/:id	DELETE	<ul style="list-style-type: none"> <li>– Supprimer un véhicule</li> <li>– id du véhicule à mettre dans l'url <a href="/api/vehicles/66212a1b33d48a71897191f6">/api/vehicles/66212a1b33d48a71897191f6</a></li> </ul>
Contract	/api/contracts	GET	<ul style="list-style-type: none"> <li>– Liste de tous les contrats</li> </ul>
	/api/contracts/:id	GET	<ul style="list-style-type: none"> <li>– Liste le contrat correspondant à l'id fourni dans l'url</li> </ul>
	/api/contracts	POST	<ul style="list-style-type: none"> <li>– Création d'un nouveau contrat</li> <li>– Contrat à créer doit être fourni dans le corps de la requête au format JSON</li> </ul> <pre>{   "vehicle_id": "662202ef7ed37b09edce3f4e",   "customer_id": "662129de0824fcaf827b2209",   "loc_begin_datetime": "2024-04-19 18:50:00",   "loc_end_datetime": "2024-04-21 10:00:00",   "price": 12 }</pre>

# PROJET FIL ROUGE – DEVELOPPER LES COMPOSANTS D'ACCES AUX DONNEES

Modèle	Chemin	Type	Résultats
	/api/contracts/:id	PATCH	<ul style="list-style-type: none"> <li>– Modification d'un contrat existant</li> <li>– Le ou les champs à modifier à fournir dans le corps de la requête au format JSON</li> <li>– id du contrat à mettre dans l'url <a href="/api/contracts/5">/api/contracts/5</a></li> </ul>
	/api/contracts/:id	DELETE	<ul style="list-style-type: none"> <li>– Supprimer un contrat</li> <li>– id du contrat à mettre dans l'url <a href="/api/contracts/7">/api/contracts/7</a></li> </ul>
	/api/contracts/search/queryparams	GET	<ul style="list-style-type: none"> <li>– Liste tous les contrats en l'absence de paramètre <a href="/api/contracts/search/queryparams">api/contracts/search/queryparams</a></li> <li>– Liste tous les contrats correspondant à un customer_id et/ou un vehicle_id et/ou un statut particulier dont les valeurs possibles sont [ongoing, completed, late] <a href="/api/contracts/search/queryparams?customer_id=662129de0824fc827b2209&amp;status=ongoing">api/contracts/search/queryparams?customer_id=662129de0824fc827b2209&amp;status=ongoing</a></li> </ul>
	api/contracts/search/delays	GET	<ul style="list-style-type: none"> <li>– Liste le nombre de contrat(s) en retard (date de retour &gt; date de fin de location + 1 heure) dans une période de temps donnée ; les paramètres date1 et date2 (au format YYYY/MM/DD ) sont à renseigner dans l'url <a href="/api/contracts/search/delays?date1=2024/04/19&amp;date2=2024/04/23">api/contracts/search/delays?date1=2024/04/19&amp;date2=2024/04/23</a></li> </ul>
	api/contracts/search/groupby/customer	GET	– Liste le nombre de contrats par customer_id
	api/contracts/search/groupby/vehicle	GET	– Liste le nombre de contrats par vehicle_id
	api/contracts/search/groupby/delay/customer	GET	– Liste le nombre de contrats, le nombre de retards et le retard moyen par customer_id
	api/contracts/search/groupby/delay/vehicle	GET	– Liste le retard moyen en minutes par vehicle_id
Billing	/api/billings	GET	– Liste de tous les paiements
	/api/billings/:id	GET	– Liste le paiement correspondant à l'id fourni dans l'url
	/api/billings	POST	<ul style="list-style-type: none"> <li>– Création d'un nouveau paiement</li> <li>– Paiement à créer doit être fourni dans le corps de la requête au format JSON</li> </ul> 
	/api/billings/:id	PATCH	<ul style="list-style-type: none"> <li>– Modification d'un paiement existant</li> <li>– Le ou les champs à modifier à fournir dans le corps de la requête au format JSON</li> <li>– id du paiement à mettre dans l'url <a href="/api/billings/5">/api/billings/5</a></li> </ul>



## PROJET FIL ROUGE – DEVELOPPER LES COMPOSANTS D'ACCES AUX DONNEES

Modèle	Chemin	Type	Résultats
	<a href="#">/api/billings/:id</a>	DELETE	<ul style="list-style-type: none"> <li>– Supprimer un paiement</li> <li>– id du paiement à mettre dans l'url <a href="#">/api/billings/7</a></li> </ul>
	<a href="#">api/billings/search/queryparams</a>	GET	<ul style="list-style-type: none"> <li>– Liste de tous les paiements <a href="#">api/billings/search/queryparams</a></li> <li>– Liste des paiements pour un contrat particuliers <a href="#">api/billings/search/queryparams?contract_id=7</a></li> </ul>
	<a href="#">api/billings/search/groupby/contract</a>	GET	<ul style="list-style-type: none"> <li>– Liste le récapitulatif des paiements effectués pour chaque contrat ainsi que le ratio paiements effectués/prix de la location <a href="#">api/billings/search/groupby/contract</a></li> <li>– Liste le récapitulatif des paiements pour un contrat particulier si contract_id fourni dans l'url <a href="#">api/billings/search/groupby/contract?contract_id=7</a></li> <li>– Permet aussi d'avoir les récapitulatifs de paiement effectués dans leur totalité 'payment=settled' <a href="#">api/billings/search/groupby/contract?payment=settled</a> ou ceux qui ne sont pas ou partiellement payés <a href="#">api/billings/search/groupby/contract?payment=ongoing</a> (dans ce cas le paramètre payment peut prendre n'importe quelle valeur ≠ 'settled')</li> </ul>