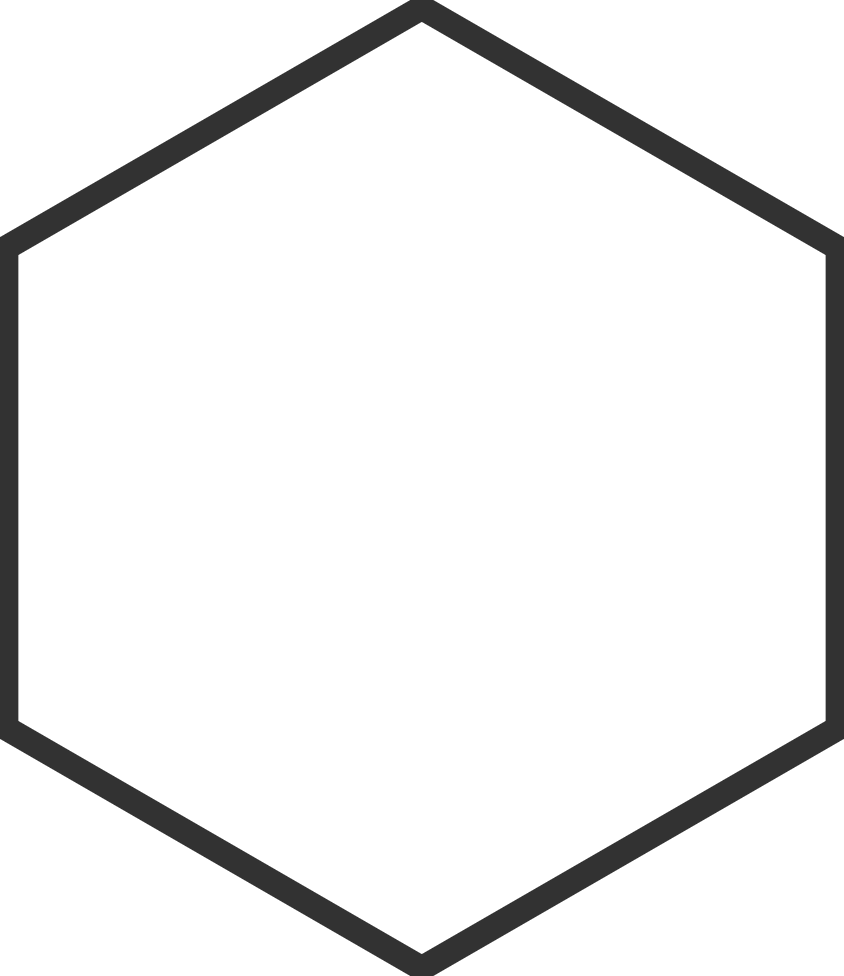
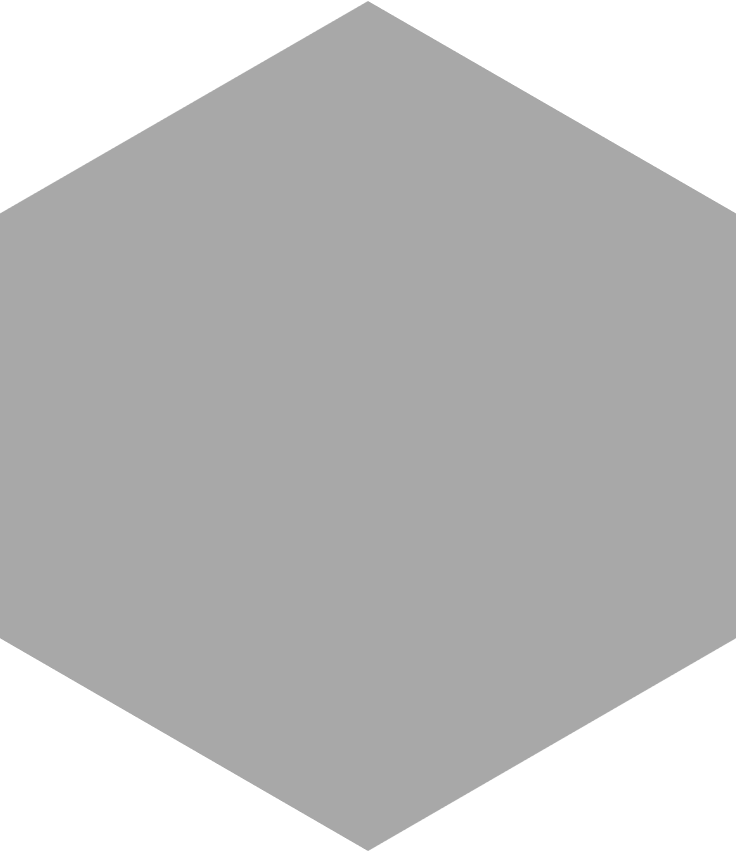
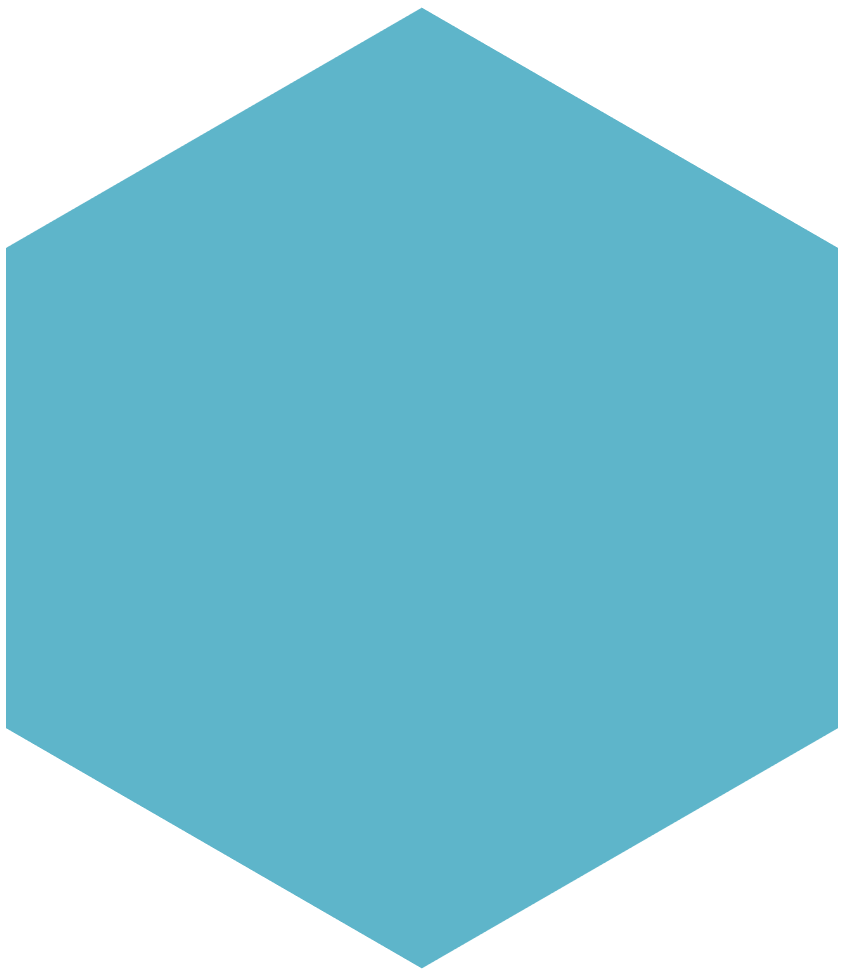


Digital HaïTian Gourde



ANNEXES

2019-2020

Auteurs :

Thomas BEATINI  
Arnaud FERNANDEZ  
Chloé MACCARINELLI  
Cédric ORTEGA

Tuteurs :

Gabriel MOPOLO-MOKE

Alexandre MAISONOBE

Gaëtan LESCOUFLAIR

Table des matières

[**I.** **Documentation Technique de l’Environnement existant** 2](#_Toc31184907)

[A. MySQL 2](#_Toc31184908)

[B. API REST 2](#_Toc31184909)

[1. Liste des routes en GET 2](#_Toc31184910)

[2. Liste des routes en POST 3](#_Toc31184911)

[3. Liste des routes en PUT 4](#_Toc31184912)

[4. Liste des routes en DELETE 4](#_Toc31184913)

[C. Angular 7 4](#_Toc31184914)

[**II.** **Documentation Profil BRH** **Erreur ! Signet non défini.**](#_Toc31184915)

[**III.** **Documentation technique dockerisation** **Erreur ! Signet non défini.**](#_Toc31184916)

# Documentation Technique de l’Environnement existant

## MySQL

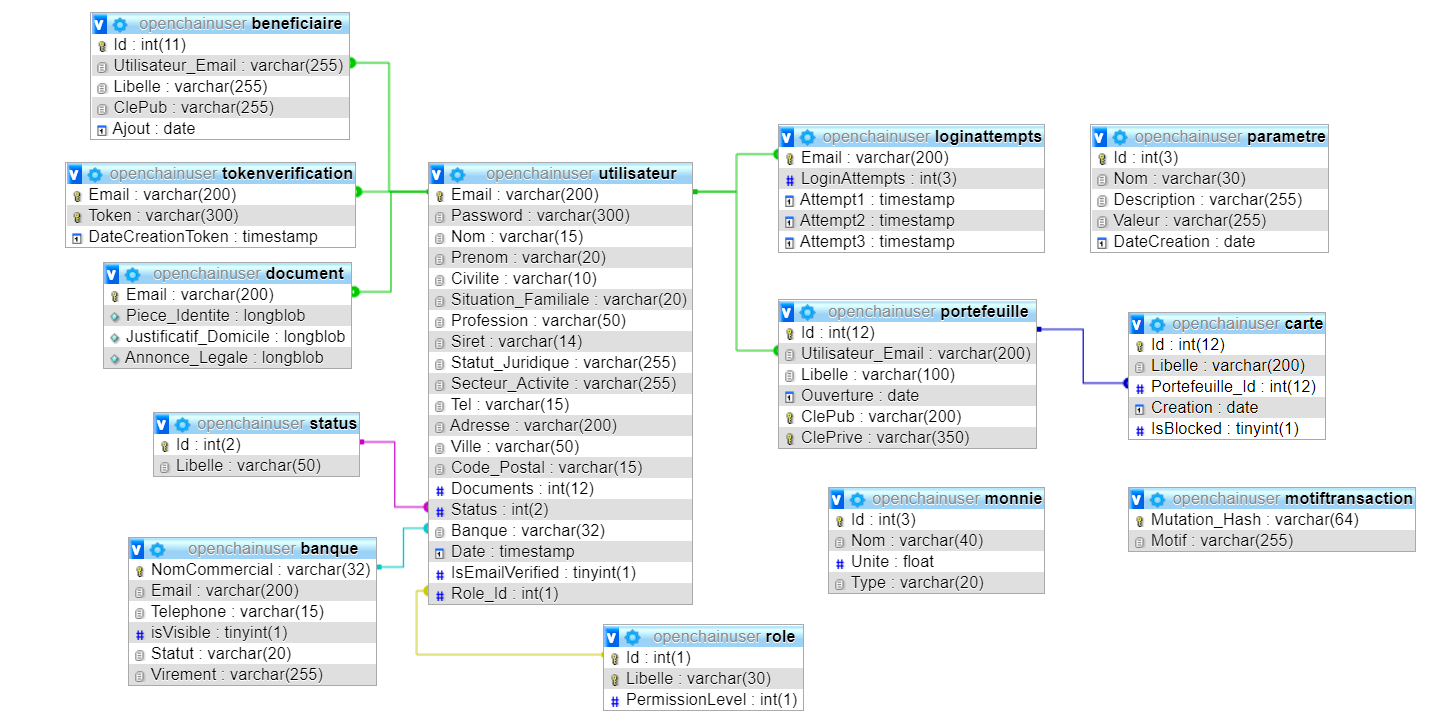
La base des données MySQL est utilisée pour gérer les informations relatives aux utilisateurs. Dans le cadre du projet cette base des données est déployée par la BRH qui a le contrôle absolu sur les données.

Figure 1 : Visualisation de la BDD MySQL déjà en place

## API REST

* index.js : point d'entrée, contient toutes les routes
* env.config.js : contient les configurations de l’api
* sql.js : contient les requêtes SQL
* outils.js : contient les fonctions utilitaires

### Liste des routes en GET

|  |  |
| --- | --- |
| /bienvenue | Affiche un page de bienvenue avec 2 liens – un vers la page web de la BRH |
| /allMonnies | Récupère la liste des taux de conversion des crypto monnaies |
| /allBanks | Récupère la liste des banques |
| /allBanksUtilisateurs |  |
| /allParametres | Récupère la liste des paramètres |
| /allBanksValid | Récupère la liste des banques (statut == Validé) |
| /allBanksNotValid | Récupère la liste des banques (statut != Validé) |
| /clients | Récupère la liste des clients par banque |
| /allClients | Récupère la liste des clients de toutes les banques |
| /allClientsBanque | Récupère la liste des clients des toutes les banques si l’on est admin |
| /contactsByUserEmail | Récupère la liste des contacts pour l’utilisateur donné |
| /clientDocuments | Récupère la liste des documents pour un client donné |
| /reValidateEmailIfEmailExpired | Renvoye l’email de vérification dans le cas où le  token de vérification a expiré (le token a une durée de vie de 24 heures) |

### Liste des routes en POST

|  |  |
| --- | --- |
| /cardsByPortefeuilleIds | Récupère la liste des cartes pour un id de portefeuille donné |
| /motifsByMutationHashes | Récupère la liste des motifs pour les hash des transactions (MutationHashes) données |
| /portefeuillesByBanqueEmail | Récupère la liste des portefeuilles pour un email utilisateur donné |
| /portefeuillesByUserEmail | Récupère la liste des portefeuilles pour un email utilisateur donné |
| /createMonnie | Permet de créer un taux de conversion |
| /createParametre | Créer un nouveau paramètre |
| / createClient | Créer un nouveau client |
| /sendDocumentsValidatedEmailToClient | Envoyer un mail de confirmation au client pour lui informer que ses documents ont été validés |
| /createPortefeuille | Créer un nouveau portefeuille |
| / createCarte | Créer une nouvelle carte |
| / createBankClient | Créer un nouveau client ayant le niveau de permission BANQUE |
| / createContact | Créer un nouveau bénéficiaire |
| /createContactByUserEmail |  |
| /insertTransactionMotif | Ajouter un motif pour un hash de transaction donne |
| /insertCommercantDocs | Insérer les documents pour un nouveau client commerçant |
| /updateCommercantDocs | Mettre à jour les documents d’un commerçant |
| /insertParticulierDocs | Insérer les documents pour un nouveau client particulier |
| /updateParticulierDocs | Mettre à jour les documents d’un particulier |
| /submit | Effectuer une transaction vers un client connaissant son email |
| /transferTo | Effectuer une transaction vers un client connaissant sa clé publique |
| /transferToUserEmail | Effectuer une transaction vers un client connaissant son email |
| /issueDHTG | Créer des DHTG et les envoyer sur son compte |
| /validateEmail | Valider une adresse email si le champ `token` est correct |
| /auth | Permet s’authentifier et récupérer son JWT |

### Liste des routes en PUT

|  |  |
| --- | --- |
| /blockClient | Bloquer un client donné connaissant son email |
| /blockCarte | Bloquer une carte connaissant son id |
| / unblockCarte | Débloquer une carte connaissant son id |
| /updateParametre | Mettre à jour des paramètres |
| /updateBanque | Mettre à jour des informations relatives à une banque |
| /updateMonnie | Mettre à jour un taux de conversion |
| / updateCarte | Mettre à jour le libelle d’une carte connaissant son id |
| /updatePortefeuilleLibelle | Mettre à jour le libellé d’un portefeuille connaissant son id |
| /updateContact | Mettre à jour le libellé et la clé publique d’un bénéficiaire connaissant son id |
| /updateClient | Mettre à jour les informations relatives au client |
| /unBlockClient | Débloquer un client |
| /unBlockBanque | Débloquer une banque |
| /unBlockOrBlockClient | Débloquer ou bloquer un client |
| /changeStatusClient | Changer le statut d’un client |

### Liste des routes en DELETE

|  |  |
| --- | --- |
| /deleteBank | Supprimer une banque |
| /deleteMonnieElectronique | Supprimer un taux de conversion |
| /deleteParametre | Supprimer un paramètre |
| /deletePortefeuille | Supprimer un portefeuille |
| /deleteClient | Supprimer un client |
| /deleteContact | Supprimer un bénéficiaire |
| /deleteCarte | Supprimer une carte client |

## Angular 7

Découpage de la partie applicative :

* Package sogebank
* Package brh
* Package common
* Package portail

Le package sogebank ainsi que le service *sogebank.service* contient toutes les interfaces relatives aux profils Particulier et Commerçant.

Le package brh contient toutes les interfaces relatives aux profils Banque privée et Banque centrale (BRH)

Les packages common et portail contient les interfaces / composants communs aux différents profils

Le service *nodeapi.service* stocke la session du client et permet de faire les appels réseau vers l’API node.