

Projet GSB

I. Présentation.....	1
II. Partie API.....	2
Structure du projet.....	2
Authentification	3
Gestion de Visiteur.....	5
Gestion de Praticien.....	5
Gestion des portefeuilles.....	6
III. Partie Application.....	7
Page Login.....	7
Page Home.....	8

I. Présentation

Cette documentation a pour objectif de présenter et d'expliquer les fonctionnalités mises en place dans l'API REST **GSB** et l'application mobile Android **GSB**.

Elle permet de comprendre :

- Le fonctionnement global de l'API REST
- L'utilisation de l'application mobile
- Les parcours utilisateurs (user stories)
- L'organisation des fonctionnalités

Ressources associées :

[Lien GitHub API REST](#)

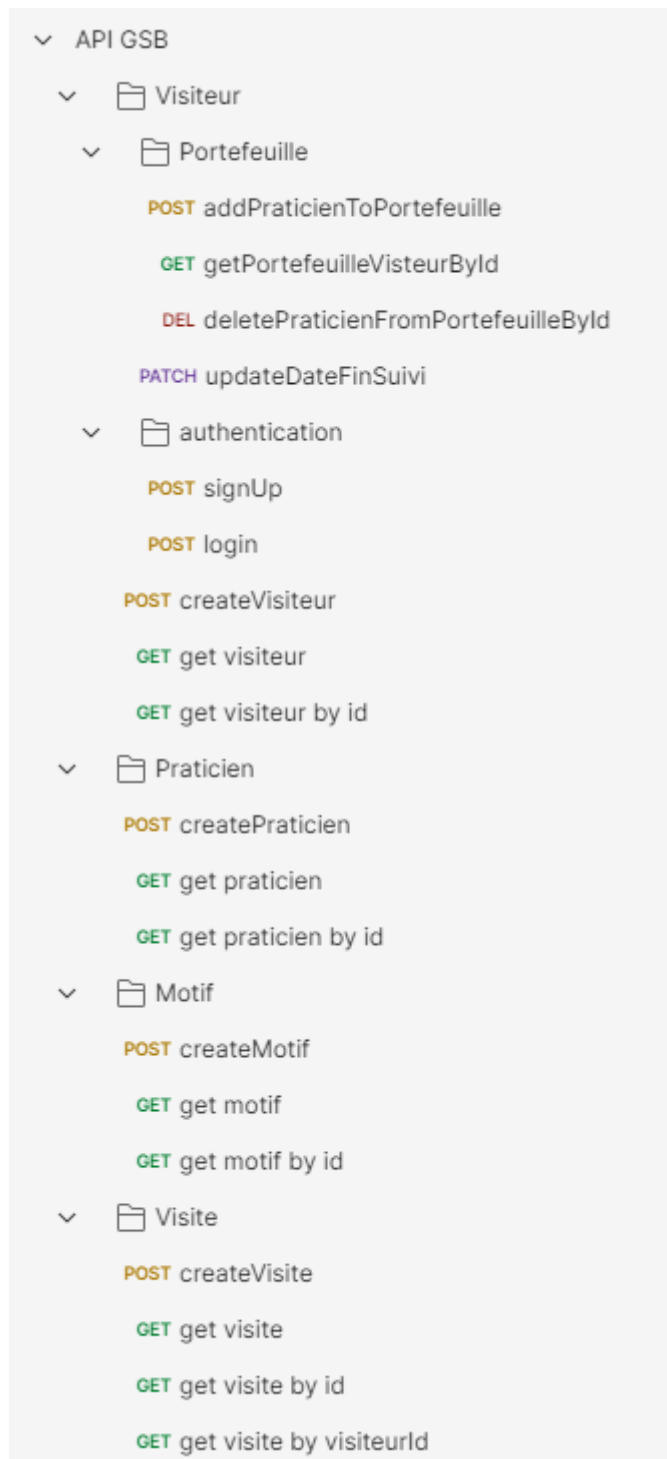
[Lien GitHub Application](#) - [Readme](#)

[Présentation du projet](#)

[User stories du projet](#)

II. Partie API

Structure du projet



Pour tester les fonctionnalités, on initialise les variables dans l'environnement DEV de Postman:

- baseUrl
- Nom
- Prenom
- visiteurId
- praticienId
- email
- token

Les données récupérées seraient enregistrées automatiquement dans l'environnement afin de réutiliser.

OSB Dev Fork | 0 Save Share

Filter variables

Variable	Type	Initial value	Current value
<input checked="" type="checkbox"/> baseUrl	default		https://orange-bassoon-r4pju56wlgv7fxqpp-3000.app.github.dev/api
<input checked="" type="checkbox"/> randomNom	default		Schmitt
<input checked="" type="checkbox"/> randomPrenom	default		Elvera
<input checked="" type="checkbox"/> visiteurId	default		69f8a497910eda2d95d1b431
<input checked="" type="checkbox"/> praticienId	default		69f895689590a881de034c9f
<input checked="" type="checkbox"/> motifId	default		
<input checked="" type="checkbox"/> visiteId	default		
<input checked="" type="checkbox"/> email	default		elvera.schmitt@gsb.com
<input checked="" type="checkbox"/> token	default		

Authentification

- POST {{baseUrl}}/visiteur/signup

On initialise les variables saisies dans le Body:

```
{
  "nom": "{{randomNom}}",
  "prenom": "{{randomPrenom}}",
  "tel": "{{randomPhone}}",
  "email": "{{randomEmail}}",
  "date_embauche": "{{randomDatePast}}",
  "password": "Password123!"
}
```

POST {{baseUrl}}/visiteur/signup

Params Authorization Headers (10) Body **Pre-request Script** Tests Settings

```
1 // Générez le nom et prénom aléatoires
2 pm.environment.set("randomNom", pm.variables.replaceIn("{{randomLastName}}");
3 pm.environment.set("randomPrenom", pm.variables.replaceIn("{{randomFirstName}}");
4
5 // Générez un numéro français valide (10 chiffres)
6 const prefixes = ['06', '07']; //mobile français
7 const prefix = prefixes[Math.floor(Math.random() * prefixes.length)];
8 const number = prefix + Math.floor(10000000 + Math.random() * 90000000);
9 pm.variables.set("randomPhone", number);
10
11 // Générez un email avec le nom et prénom (en minuscules)
12 const nom = pm.environment.get("randomNom");
13 const prenom = pm.environment.get("randomPrenom");
14 const email = prenom.toLowerCase() + "." + nom.toLowerCase() + "@gsb.com";
15 pm.variables.set("randomEmail", email);
16
```

Body Cookies (1) Headers (38) Test Results Status: 201 Created Time: 500 ms Size: 3 KB Save

Pretty Raw Preview Visualize JSON Save

```
1
2 {
3   "success": true,
4   "message": "Visiteur créé avec succès",
5   "data": {
6     "nom": "Schmitt",
7     "prenom": "Elvera",
8     "tel": "06608389774",
9     "email": "elvera.schmitt@gsb.com",
10    "password": "32b5109n657awC0z1uWY6zevNg.WsJTIsaCMXPg3hL6MmSd4cT.yEvnVas",
11    "id": "69f8a497910eda2d95d1b431",
12    "dateEmbauche": "2026-05-04T13:52:23.289Z",
13    "createdAt": "2026-05-04T13:52:23.292Z",
14    "updatedAt": "2026-05-04T13:52:23.292Z",
15    "id": "69f8a497910eda2d95d1b431"
16  }
17 }
```


Gestion de Visiteur

- GET {{baseUrl}}/visiteur

Récupérer toutes les visiteurs enregistrés.

GET {{baseUrl}}/visiteur

Authorization: Bearer Token ({{token}})

```

1 {
2   "success": true,
3   "count": 4,
4   "data": [
5     {
6       "_id": "69c143829ef78ffa55e8f2e4",
7       "nom": "Hezman",
8       "prenom": "Dax",
9       "tel": "0629455436",
10      "email": "dax.hezman@gsb.com",
11      "id": "69c143829ef78ffa55e8f2e4"
12    },
13    {
14      "_id": "69ce63bd887236aaca573b88",
15      "nom": "Farrell",
16      "prenom": "Lonny",
17      "tel": "0638834323",
18      "email": "lonny.farrell@gsb.com",
19      "id": "69ce63bd887236aaca573b88"
20    }
21  ]
22 }
    
```

Gestion de Praticien

- GET {{baseUrl}}/praticien

Récupérer toutes les praticiens.

GET {{baseUrl}}/praticien

Key	Value	Description
Key	Value	Description

```

1 {
2   "success": true,
3   "count": 1,
4   "data": [
5     {
6       "_id": "69f89568990a881de034c9f",
7       "nom": "Bernier",
8       "prenom": "Cydney",
9       "tel": "0758357593",
10      "email": "cydney.bernier@gsb.com",
11      "rue": "Avenue des Champs-Élysées",
12      "codePostal": "75001",
13      "ville": "Paris",
14      "v": 0
15    }
16  ]
17 }
    
```

Gestion des portefeuilles

- POST {{baseUrl}}/visiteur/{{visiteurId}}/portfeuille

Ajouter un praticien dans le portefeuille d'un visiteur.

```
{
  "praticienId": " "
}
```

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** {{baseUrl}}/visiteur/{{visiteurId}}/portfeuille
- Body:**

```
{
  "praticienId": " "
}
```
- Status:** 201 Created
- Time:** 264 ms
- Response Body (JSON):**

```
{
  "success": true,
  "message": "Praticien ajout\u00e9 au portefeuille avec succ\u00e8s",
  "data": {
    "dateDebutSuivi": "2026-05-04T15:10:20.673Z",
    "dateFinSuivi": null,
    "visiteur": "69f8a497910eda2d95d1b431",
    "praticien": "69318009eb8a9b3e0ef9c42",
    "_id": "69f8b6dce3bf3275b497768f",
    "createdAt": "2026-05-04T15:10:20.677Z",
    "updatedAt": "2026-05-04T15:10:20.677Z",
    "id": "69f8b6dce3bf3275b497768f"
  }
}
```

- GET {{baseUrl}}/visiteur/{{visiteurId}}/portfeuille

Recup\u00e9rer toutes les portefeuilles d'un visiteur.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** {{baseUrl}}/visiteur/{{visiteurId}}/portfeuille
- Status:** 200 OK
- Response Body (JSON):**

```
{
  "success": true,
  "data": [
    {
      "_id": "69f8b6dce3bf3275b497768f",
      "dateDebutSuivi": "2026-05-04T15:10:20.673Z",
      "dateFinSuivi": null,
      "visiteur": "69f8a497910eda2d95d1b431",
      "praticien": null,
      "id": "69f8b6dce3bf3275b497768f"
    }
  ]
}
```

III. Partie Application

Pour assurer le fonctionnement, l'api fonctionne en local (codespace).

Avant de réaliser les actions, vérifier que le codespace de l'api est activé et que le port 3000 est public.

```
@Juy246 → /workspaces/api_rest_gsb_visite (AWS_migration) $ npm run dev

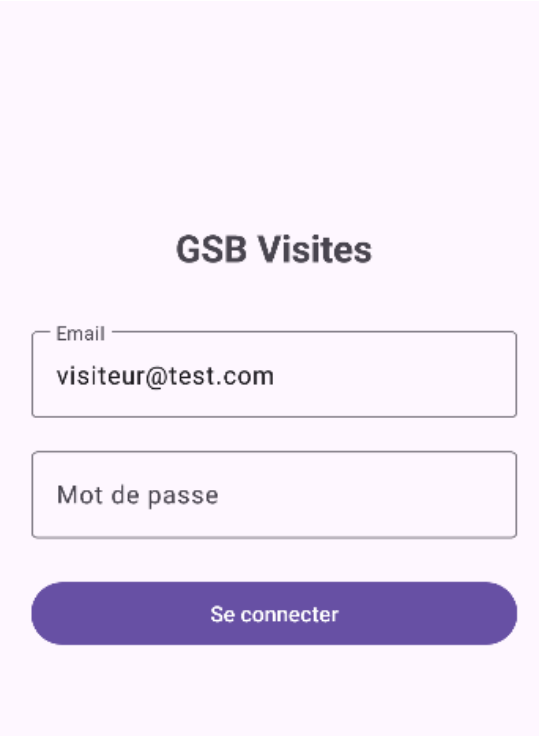
> api_rest_gsb_visite@1.0.0 dev
> nodemon

[nodemon] 3.1.11
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): src/**/*
[nodemon] watching extensions: ts,json
[nodemon] starting `tsx src/server.ts`
[dotenv@17.3.1] injecting env (5) from .env -- tip: ⚙️ suppress all logs with { quiet: true }
[dotenv@17.3.1] injecting env (0) from .env -- tip: 🔐 agentic secret storage: https://dotenvx.com/as2
=====
Serveur démarré sur le port 3000
Environnement: development
=====
Connexion à MongoDB Atlas réussie
```

Page Login

- Saisir l'adresse mail et le mot de passe de l'utilisateur
- Bouton "Se connecter"

-> Si l'utilisateur est validé, un token se génère pour récupérer les informations de l'utilisateur



```
Données reçues pour la connexion du visiteur: { email: 'dax.herman@gsb.com', password: 'Password123!' }
```

Page Home

- Affichage de prénom, nom, adresse mail de l'utilisateur connecté
- Affichage de la liste de portefeuille actifs du visiteur
- Bouton "Créer un praticien"

