
API

Documentation des interfaces API du projet [etudiant.bj](#), décrivant les échanges entre le frontend, l'API Gateway et les services backend.

API

Bienvenue dans la documentation des API du projet [etudiant.bj](#) .

Cette section présente les interfaces de communication utilisées par la plateforme afin de permettre les échanges entre le frontend, l'API Gateway et les différents services backend.

Présentation générale

Les API constituent le point de communication entre les différentes parties du système.

Elles permettent notamment :

- au frontend d'accéder aux fonctionnalités backend ;
- aux différents services backend d'échanger des informations ;
- de transmettre et récupérer les données nécessaires au fonctionnement de la plateforme.

Les API permettent ainsi de garantir une communication structurée, sécurisée et évolutive entre les composants du système.

Architecture des échanges

Les échanges entre les différents composants suivent généralement le flux suivant :

Utilisateur



Frontend



API Gateway



Services Backend



Base de données

L'API Gateway constitue le point d'entrée principal des requêtes provenant des applications clientes.

Elle assure la communication entre le frontend et les différents services backend.

API Gateway

L'API Gateway est responsable de la centralisation des accès aux services backend.

Elle joue plusieurs rôles :

- réception des requêtes provenant du frontend ;
- routage des requêtes vers les services concernés ;
- centralisation de certaines règles communes ;
- sécurisation des échanges ;
- gestion de la communication avec les microservices.

Technologie utilisée :

- **Go**

Types d'API

Les communications entre les composants reposent principalement sur des API HTTP.

Les opérations courantes utilisent les méthodes HTTP suivantes :

MÉTHODE HTTP	DESCRIPTION
GET	Récupération d'informations
POST	Création d'une nouvelle ressource
PUT	Modification complète d'une ressource
PATCH	Modification partielle d'une ressource
DELETE	Suppression d'une ressource

Structure d'une requête API

Une requête envoyée à une API contient généralement plusieurs éléments.

URL

L'URL permet d'identifier la ressource ciblée.

Exemple :

GET /api/users

Headers

Les headers permettent de transmettre des informations complémentaires.

Ils peuvent contenir notamment :

- le token d'authentification ;

- le type de contenu envoyé ;
- des informations nécessaires à la sécurité des échanges.

Exemple :

Authorization: Bearer

Content-Type: application/json

Body

Le body contient les données envoyées lors des opérations de création ou de modification.

Exemple :

```
{  
  "name": "Utilisateur",  
  "email": "utilisateur@example.com"  
}
```

Format des réponses API

Les API retournent généralement les données au format JSON.

Exemple :

```
{  
  "success": true,  
  "data": {  
    "id": 1,  
    "name": "Utilisateur"  
  },  
  "message": "Opération réussie"  
}
```

Une réponse API peut contenir :

- le résultat de l'opération ;

- les données demandées ;
- un message d'information ;
- les éventuelles erreurs rencontrées.

Authentification et sécurité

Les API du projet utilisent des mécanismes d'authentification afin de protéger les ressources accessibles.

La sécurité comprend notamment :

- l'identification des utilisateurs ;
- la gestion des rôles et permissions ;
- la protection des routes sensibles ;
- la sécurisation des échanges entre les composants.

Les composants liés à l'authentification comprennent notamment :

- **Keycloak** ;
- **RBER Connect**.

Communication entre les microservices

Les différents services backend communiquent entre eux afin d'échanger les informations nécessaires au fonctionnement de la plateforme.

La communication inter-services repose notamment sur :

RabbitMQ

RabbitMQ permet :

- la communication asynchrone entre services ;
- l'échange d'événements ;
- le découplage entre les composants ;

- l'amélioration de la fiabilité du système.

Cette approche permet aux services de fonctionner de manière indépendante tout en collaborant efficacement.

Organisation des endpoints

Chaque service backend expose des endpoints permettant d'accéder aux fonctionnalités qu'il fournit.

Chaque endpoint doit être documenté avec les informations suivantes :

- méthode HTTP utilisée ;
- URL de l'endpoint ;
- description de la fonctionnalité ;
- paramètres nécessaires ;
- format des données envoyées ;
- format de la réponse ;
- erreurs possibles.

Exemple de documentation d'un endpoint

Récupération d'une ressource

Méthode :

GET

URL :

/api/ressource

Réponse :

```
{  
  "id": 1,
```

```
"name": "Exemple"  
}
```

Gestion des erreurs

Les API doivent retourner des réponses explicites lorsqu'une erreur survient.

Les erreurs peuvent concerner :

- une requête invalide ;
- une ressource inexistante ;
- un problème d'authentification ;
- une erreur interne du serveur.

Exemple :

```
{  
  "success": false,  
  "message": "Ressource introuvable"  
}
```

Documentation et tests des API

Les API doivent être testées et documentées afin de faciliter leur utilisation et leur maintenance.

Les outils pouvant être utilisés sont notamment :

- Postman ;
- Swagger / OpenAPI ;
- outils de tests automatisés.

Bonnes pratiques API

Les développements API doivent respecter plusieurs bonnes pratiques :

- utiliser des noms de ressources clairs ;
- respecter les conventions REST ;
- retourner des réponses cohérentes ;
- sécuriser les endpoints sensibles ;
- documenter chaque modification ;
- maintenir une compatibilité entre les versions.

Maintenance de la documentation API

Toute modification concernant les API doit être accompagnée d'une mise à jour de cette documentation.

Les développeurs doivent documenter :

- les nouveaux endpoints ;
- les modifications de paramètres ;
- les changements de structure des réponses ;
- les nouvelles règles de sécurité ;
- les évolutions des services.

Cette page constitue la documentation générale des API du projet [etudiant.bj](#) .

Les détails spécifiques de chaque endpoint seront ajoutés dans les sections dédiées aux différents services backend.