

🔒 Routes protégées 18

GET	api/user/names	Récupérer les informations relatives aux utilisateurs inscrits
GET	api/user	Récupérer les informations relatives à l'utilisateur actif
POST	api/logout	Déconnecter l'utilisateur authentifié de l'application
PATCH	api/settings/profile	Mettre à jour les informations relatives à l'utilisateur authentifié
PATCH	api/settings/password	Mettre à jour le mot de passe de l'utilisateur authentifié
GET	api/projects	Récupérer la totalité des projets
GET	api/projects/{id}	Récupérer un projet
POST	api/projects	Créer un nouveau projet
PATCH	api/projects/{project}	Mettre à jour un projet
DELETE	api/projects/{project}	Supprimer un projet
POST	api/projects/ranking/{category}	Récupérer les meilleurs projets
GET	api/categories	Récupérer l'ensemble des catégories
GET	api/likes	Récupère la totalité des mentions like
POST	api/likes	Ajouter une mention like
DELETE	api/likes/{like}	Supprimer une mention like
GET	api/games	Récupérer l'ensemble des matchs joués
POST	api/games/init/{category}	Récupérer deux projets pour participer à un match
PATCH	api/games/update	Mettre à jour les projets et créer l'historique d'un match

🔓 Routes non-protégées 4

POST	api/login	Authentifier un utilisateur pour accéder à l'application
POST	api/register	Inscrire un nouvel utilisateur à l'application
POST	api/oauth/{driver}	Contacter une application tiers pour obtenir des informations utilisateur
GET	api/oauth/{driver}/callback	Obtenir un jeton d'authentification de la part de l'application tiers et créer un nouvel utilisateur

**POST****api/login**

Authentifier un utilisateur pour accéder à l'application

**Body**

```
{
  email: String      L'e-mail renseigné par l'utilisateur
  password: String    Le mot de passe renseigné par l'utilisateur
}
```

Réponse

```
{
  token: String
  token_type: String
  expires_in: int
}
```

POST**api/logout**

Déconnecter l'utilisateur authentifié de l'application

**POST****api/register**

Inscrire un nouvel utilisateur à l'application

**Body**

```
{
  name: String        Le nom renseigné par l'utilisateur
  email: String        L'e-mail renseigné par l'utilisateur
  password: String     Le mot de passe renseigné par l'utilisateur
  password_confirmation: String  La confirmation du mot de passe renseigné précédemment
}
```

Réponse**User**

```
{
  id: int
  name: String
  email: String
  email_verified_at: timestamp
  photo_url: String
  created_at: timestamp
  updated_at: timestamp
}
```

**POST****api/oauth/{driver}**

Contacter une application tiers pour obtenir des informations utilisateur

**Paramètre****{driver}**

Le nom de l'application tiers

Exemple

google

Réponse

```
{
  url: String    L'url de retour de l'application tiers
}
```

GET**api/oauth/{driver}/callback**

Obtenir un jeton d'authentification de la part de l'application tiers et créer un nouvel utilisateur

**Paramètre****{driver}**

Le nom de l'application tiers

Exemple

github

Réponse

```
{
  token: String
  token_type: String
  expires_in: int
}
```

GET

api/user

Récupérer les informations relatives à l'utilisateur actif



Réponse

User ▾

```
{
  id: int
  name: String
  email: String
  email_verified_at: timestamp
  photo_url: String
  created_at: timestamp
  updated_at: timestamp
}
```

GET

api/user/names

Récupérer les informations relatives aux utilisateurs inscrits



Réponse

[

User ▾

```
{
  id: int
  name: String
  photo_url: String
}
```

]

PATCH

api/settings/profile

Mettre à jour les informations relatives à l'utilisateur authentifié



Body

```
{
  name: String
  email: String
}
```

Réponse

User ▾

```
{
  id: int
  name: String
  email: String
  email_verified_at: timestamp
  photo_url: String
  created_at: timestamp
  updated_at: timestamp
}
```



PATCH

api/settings/password

Mettre à jour le mot de passe de l'utilisateur authentifié



Body

```
{  
  password: String  
  password_confirmation: String  
}
```

GET

api/projects

Récupérer la totalité des projets



Réponse

```
[
  Project ▾
    {
      id: int
      title: String
      description: String
      filename: String
      wins: int
      losses: int
      rank: int
      score: int
      created_at: timestamp
      updated_at: timestamp
      user_id: int
      category_id: int
    }
]
```

GET

api/projects/{id}

Récupérer un projet



Paramètre

{id}

L'identifiant du projet à récupérer

Réponse

```
Project ▾
{
  id: int
  title: String
  description: String
  filename: String
  wins: int
  losses: int
  rank: int
  score: int
  created_at: timestamp
  updated_at: timestamp
  user_id: int
  category_id: int
}
```



POST

api/projects

Créer un nouveau projet



Body

```
{
  user_id: int      L'identifiant de l'utilisateur actif
  title: String     Le titre du projet
  description: String La description du projet
  image: String     Le nom du fichier encodé du projet
  category: String  Le nom de la catégorie du projet
}
```

Réponse

Project ▾

```
{
  id: int
  title: String
  description: String
  filename: String
  wins: int
  losses: int
  rank: int
  score: int
  created_at: timestamp
  updated_at: timestamp
  user_id: int
  category_id: int
}
```

DELETE

api/projects/{project}

Supprimer un projet

Paramètre

{project}

L'identifiant du projet à supprimer

Réponse

Statut **String**

Exemple "Deleted project item"

PATCH
api/projects/{project}

Mettre à jour un projet


Paramètre
{project}

L'identifiant du projet à mettre à jour

Body

```
{
  title: String      Le titre du projet
  description: String La description du projet
}
```

Réponse
Project

```
{
  id: int
  title: String
  description: String
  filename: String
  wins: int
  losses: int
  rank: int
  score: int
  created_at: timestamp
  updated_at: timestamp
  user_id: int
  category_id: int
}
```

POST
api/projects/ranking/{category}

Récupérer les meilleurs projets


Paramètre
{category}

Le slug de la catégorie demandée

Réponse

```
[
  Project {
    id: int
    title: String
    description: String
    filename: String
    wins: int
    losses: int
    rank: int
    score: int
    created_at: timestamp
    updated_at: timestamp
    user_id: int
    category_id: int
  }
]
```




POST

api/projects/ranking/{category}

Récupérer les meilleurs projets



Paramètre

{category}

Le slug de la catégorie demandée

Réponse

```
[
  Project {
    id: int
    title: String
    description: String
    filename: String
    wins: int
    losses: int
    rank: int
    score: int
    created_at: timestamp
    updated_at: timestamp
    user_id: int
    category_id: int
  }
]
```

GET

api/categories

Récupérer l'ensemble des catégories



Réponse

```
[
  Category ▾
    {
      id: int
      name: String
      slug: String
      created_at: timestamp
      updated_at: timestamp
    }
]
```

GET

api/games

Récupérer l'ensemble des matchs joués



Réponse

```
[
  Game ▾
  {
    id: int
    winner: id
    loser: id
    created_at: timestamp
    updated_at: timestamp
  }
]
```

POST

api/games/init/{category}

Récupérer deux projets pour participer à un match



Paramètre

{category}

Le slug de la catégorie demandée

Réponse

Les deux projets allant participé au match

```
[
  Project ▾
  {
    id: int
    title: String
    description: String
    filename: String
    wins: int
    losses: int
    rank: int
    score: int
    created_at: timestamp
    updated_at: timestamp
    user_id: int
    category_id: int
  }
]
```

GET

api/likes

Récupère la totalité des mentions like



Réponse

```
[
  Like ▾
    {
      id: int
      created_at: timestamp
      updated_at: timestamp
      project_id: id
      user_id: id
    }
]
```

POST

api/likes

Ajouter une mention like



Body

```
{
  project: int      L'identifiant du projet
  user: int         L'identifiant de l'utilisateur
}
```

Réponse

```
Like ▾
{
  id: int
  created_at: timestamp
  updated_at: timestamp
  project_id: id
  user_id: id
}
```

DELETE

api/likes/{like}

Supprimer une mention like



Paramètre

{like} L'identifiant du like à supprimer

Réponse

Statut String

Exemple "Deleted like"