





▷△ ⇨⇨ ▷▷↓ ⇨▽◀ ⇨⇨ ↵→⇨△⇨↑↓◀⇨⇨◀▷△⇨ ⇨▷ ▲△⇨↖⇨  
◀↓ ⇨⇨ ⇒ ▷⇨ U\_⇨⇨

- bookly-hybrid
- config C:\Users\sacha\Desktop\BACKEND\bo
  - JS db.mongo.js
  - JS db.postgres.js
- controllers
  - JS bookController.js
  - JS mixController.js
  - JS profileController.js
  - JS userController.js
- models
  - JS book.model.js
  - JS mix.model.js
  - JS profile.model.js
  - JS user.model.js
- node\_modules
- routes
  - JS bookRoutes.js
  - JS mixRoutes.js
  - JS profileRoutes.js
  - JS userRoutes.js
- .env
- { package-lock.json
- { package.json
- JS server.js

Méthode	Endpoint	Action
GET	/api/users	liste des utilisateurs
POST	/api/users	ajout d'un utilisateur
GET	/api/books	liste des livres
POST	/api/books	ajout d'un livre

Méthode	Endpoint	Action
GET	/api/profiles/:userId	récupère le profil Mongo d'un utilisateur
POST	/api/profiles	crée un profil
PUT	/api/profiles/:userId	met à jour les préférences ou l'historique

The screenshot shows the Postman application interface. The left sidebar contains collections like 'Orium's Workspace', 'API POO', 'TP1', 'TP2', and 'TP BDD'. The main area displays an API request for 'GET /api/user-full/1' with a response body showing a user profile with ID 1, name Alice, email alice@example.com, and preferences for dark theme and French language.

Orium's Workspace

New Import

TP BD GET health GET GetA POST Cre POST Profi GET Profi POST Profi GET FUL FULL

Save Share

Send

GET http://localhost:3000/api/user-full/1

Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (8) Test Results

200 OK 51 ms 487 B Save Response

{ } JSON Preview Visualize

```
1 {  
2   "user": {  
3     "_id": 1,  
4     "name": "Alice",  
5     "email": "alice@example.com"  
6   },  
7   "profile": {  
8     "_id": 1,  
9     "preferences": {  
10       "theme": "dark",  
11       "lang": "fr"  
12     },  
13     "history": [  
14       {}  
15     ],  
16     "createdAt": "2025-11-05T12:53:53.359Z",  
17     "__v": 0  
18   }  
19 }
```

↶↑⇒▶△ ↵⇒ ↴↑⇒◀◀◀ ↶▶ ▶▽⇒△ ⇨⇒ ▶◀↓◀↓▽⇒ ↪↑▽◀→△⇒↖○  
↳ ↷⇒⇒ ↴↙ ↲⇒▶◀ ↶⇒⇒▽ ▶⇒ ▲△⇒↘↓⇒△ ◀⇒↘▲▽ ▽⇒◀◀◀△  
⇒⇒ ▲↙⇒⇒ ↵⇒ ▲↑↑↙ ↶⇒ ⇒ ↶⇒⇒⇒⇒▶↓↑⇒ ⇒⇒ ↳⇒⇒ ↪↑▽◀→△  
⇒▽↘

↶↑↖↶↶↖ ↶↑↖↶↶↖ ↶△↖↶↶↖



```
const mongoose = require('mongoose');

const { Schema } = mongoose;

const ProfileSchema = new Schema({
  _id: { type: Number, required: true },
  preferences: { type: Schema.Types.Mixed, default: {} },
  history: { type: [Schema.Types.Mixed], default: [] },
  createdAt: { type: Date, default: Date.now }
});

module.exports = mongoose.models.Profile || mongoose.model('Profile', ProfileSchema);
```

↑↔↔ ⌄△▶↔ ↔▽◀ ↔↕↔↔ ↔↔↔▽ ↵↔ ⌄↕↔◀△↕↙↙↔△ ▽↔↔▽  
▶◀↓↓▽↔△ ↵↔▽ ↔▶▲↕△◀▽ ↔↕↙↙↔ ↵→↔↓ ▲▶ ←↔↓△↔ ↔◀  
↔↔ ▶▽↔△▽

A set of small, colorful navigation icons used for navigating through the presentation slides.

↵⇒↓◀ ▲⇒△ ↗⇒↑⇒ ↗⇒↑↖  
 ↙↖↔▶◀ ⇨⇒ ↵⇒ ⇨↖⇒▽▽⇒ ↗⇒⇒ ⋯ ↙⇒◀ ↪  
 ↤⇒↔⇨◀◀⇒▶△ ⇨⇒⇨△↓▶↙ ↤⇒⇒↔⇨⇒  
▶▶▶↘\$△↓▶↙⇒→⇒⇒⇒↘↖◀△  
 ▽⇒↑⇒ ↵\$△↓▶↙⇒→⇒⇒⇒↘↖◀△