Partie 2 — Version avancée (Architecture + Auth JWT)

Objectif :

- Structurer proprement le projet
- Séparer routes / controllers / services
- Ajouter authentification JWT
- Protéger les routes

---

1. Installer dépendances supplémentaires

```
npm install @fastify/jwt bcrypt
npm install -D @types/bcrypt
```

---

2. Nouvelle structure

```
src/
 ├── server.ts
 ├── prisma.ts
 ├── plugins/
 │     └── jwt.ts
 ├── controllers/
 ├── services/
 └── routes/
```

---

3. Ajouter modèle User

Modifier `schema.prisma` :

```
model User {
  id        Int      @id @default(autoincrement())
  name      String
  email     String   @unique
  password  String
  createdAt DateTime @default(now())
}
```

Migration :

```
npx prisma migrate dev --name add_user
```

---

4. Plugin JWT

`src/plugins/jwt.ts`

```
import fp from "fastify-plugin"
import jwt from "@fastify/jwt"

export default fp(async (app) => {
  app.register(jwt, {
    secret: "supersecret"
  })

  app.decorate("authenticate", async (request: any, reply: any) => {
    await request.jwtVerify()
  })
})
```

---

5. Séparation Service / Controller

Service = logique métier Controller = interface HTTP

Exemple logique :

Service :

```
async registerUser(data)
```

Controller :

```
async function register(request, reply)
```

---

6. Protection de route

```
app.get("/posts", {
  preHandler: [app.authenticate]
}, handler)
```

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/web/api/prisma-fastify/p2?rev=1772586176

Last update: **2026/03/04 02:02**