

Backend step

Wednesday, April 09, 2025 2:15 PM

Step 1: Create the Backend Project

npm install -g @nestjs/cli

nest new projet-backend

choose npm in installation

Cd projet-backend->terminal

npm install @nestjs/typeorm typeorm pg

This gives you:

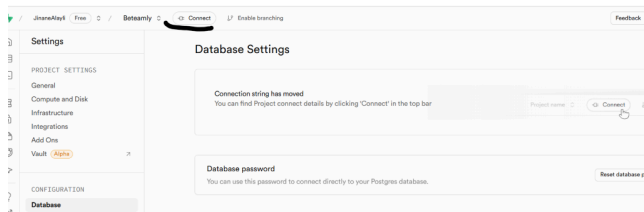
- @nestjs/typeorm: NestJS integration for TypeORM
- typeorm: the ORM (connects NestJS to PostgreSQL)
- pg: PostgreSQL driver

Create Your Project on Supabase

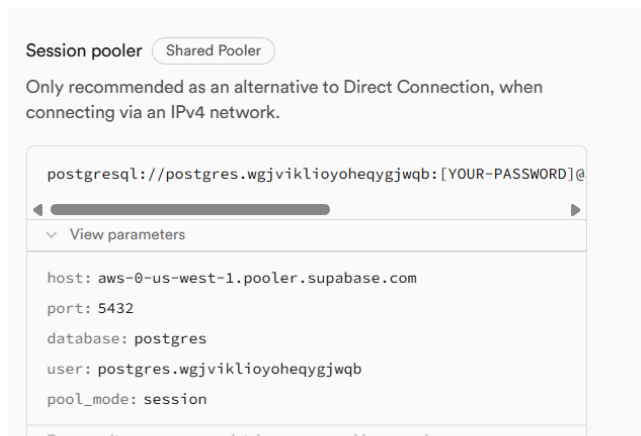
Grab Your Database Credentials

1. Go to the **Settings > Database** tab
2. Copy the following:

- **Host**
- **Port** (usually 5432)
- **Database name**
- **Username** (default is postgres)
- **Password** (you chose this during project setup)



Telit whde



```
.env exp
```

```
DB_HOST=aws-0-us-west-1.pooler.supabase.com
DB_PORT=5432
DB_USERNAME=postgres.wgjjviklioyoheqygjwqb
DB_PASSWORD=*****
DB_NAME=postgres
NODE_ENV=development
```

```
App.module.ts
```

```
import { Module } from '@nestjs/common';
import { ConfigModule, ConfigService } from '@nestjs/config';
import { TypeOrmModule } from '@nestjs/typeorm';

@Module({
  imports: [
    ConfigModule.forRoot({
      isGlobal: true,
    }),

    TypeOrmModule.forRootAsync({
      imports: [ConfigModule],
      inject: [ConfigService],
      useFactory: (config: ConfigService) => {
        const isProd = config.get<string>('NODE_ENV') === 'production';

        return {
          type: 'postgres',
          host: config.get<string>('DB_HOST', 'localhost'),
          port: parseInt(config.get<string>('DB_PORT', '5432')),
          username: config.get<string>('DB_USERNAME', 'postgres'),
          password: config.get<string>('DB_PASSWORD', ''),
          database: config.get<string>('DB_NAME', 'postgres'),

          synchronize: !isProd,

          autoLoadEntities: true,

          ssl: true,
          extra: {
            ssl: {
              rejectUnauthorized: false,
            },
          },
        };
      },
    }),
  ],
})
export class AppModule {}
```

Btrj3e bt5la2e l tables module

Role example

terminal

nest g module users

nest g service users

nest g class users/user.entity --no-spec

Kermet fastify

npm install @nestjs/platform-fastify fastify

W

bt3ytilo main.ts

```
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';
import { FastifyAdapter, NestFastifyApplication, } from '@nestjs/platform-fastify';
async function bootstrap() {
  const app = await NestFactory.create<NestFastifyApplication>(AppModule, new
FastifyAdapter(),);
  await app.listen(process.env.PORT ?? 3000);
}
bootstrap();
```