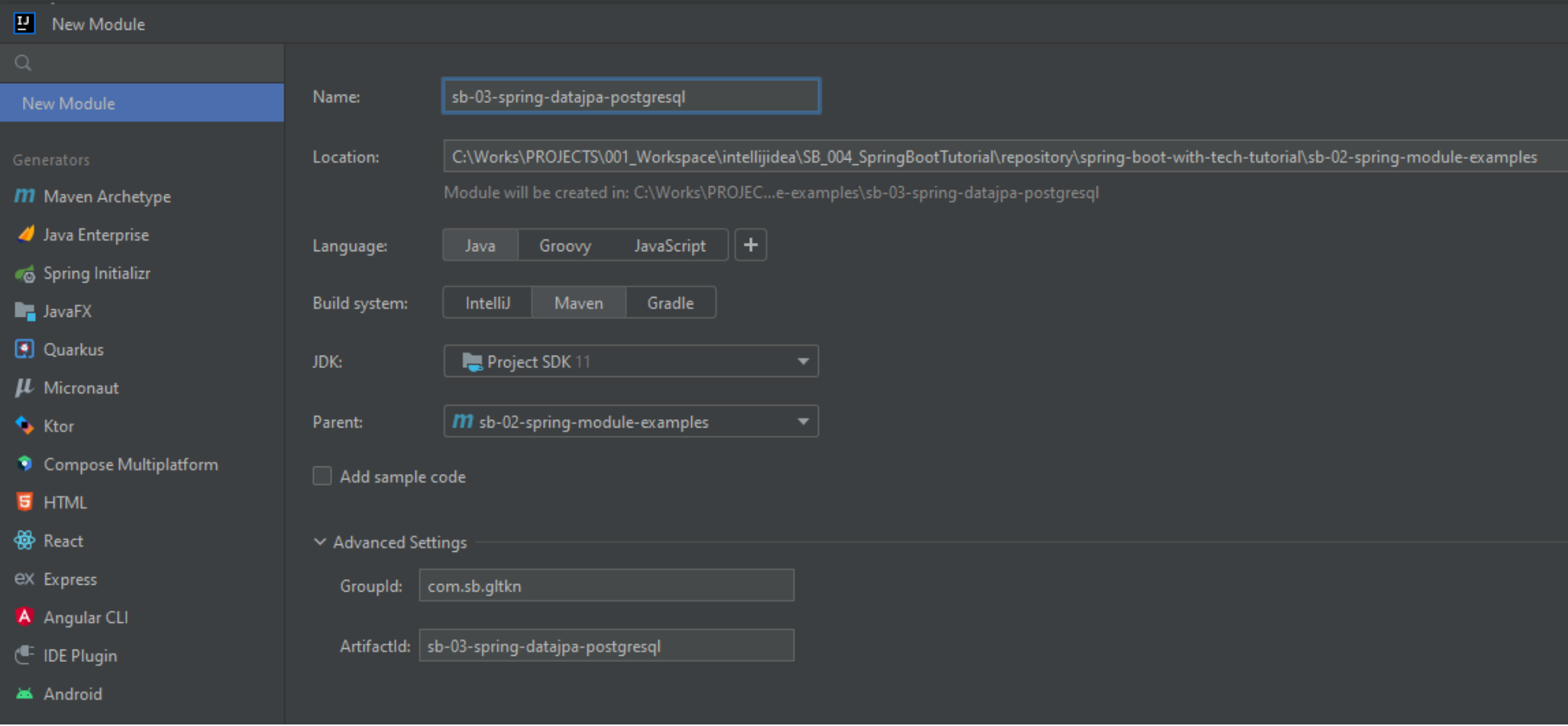
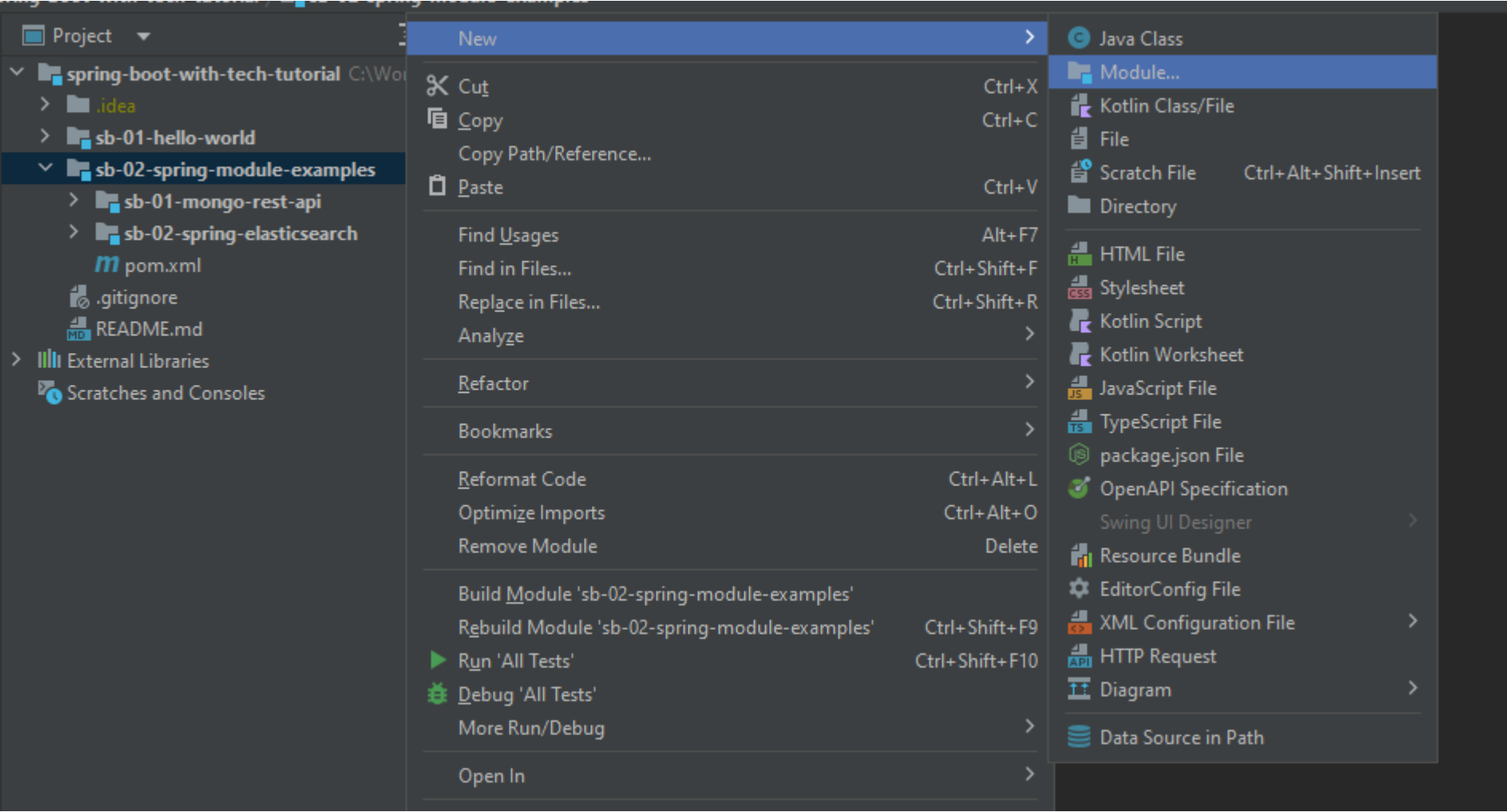


Projemizi aşağıdaki gibi oluşturalım.



```
14 <groupId>org.sb.gltkn</groupId>
15 <properties>
16   <maven.compiler.source>11</maven.compiler.source>
17   <maven.compiler.target>11</maven.compiler.target>
18 </properties>
19
20 <dependencies>
21   <!-- spring-boot web -->
22   <dependency>
23     <groupId>org.springframework.boot</groupId>
24     <artifactId>spring-boot-starter-web</artifactId>
25     <version>2.7.0</version>
26   </dependency>
27
28   <!-- spring-boot data-jpa -->
29   <dependency>
30     <groupId>org.springframework.boot</groupId>
31     <artifactId>spring-boot-starter-data-jpa</artifactId>
32   </dependency>
33   <!-- postgresql -->
34   <dependency>
35     <groupId>org.postgresql</groupId>
36     <artifactId>postgresql</artifactId>
37     <scope>runtime</scope>
38   </dependency>
39   <!-- lombok -->
40   <dependency>
41     <groupId>org.projectlombok</groupId>
42     <artifactId>lombok</artifactId>
43     <optional>true</optional>
44   </dependency>
45 </dependencies>
```

Postgresql için **user:** postgres **password:** pass1234

```
1 version: '3.1'
2
3 services:
4
5   db:
6     image: postgres
7     restart: always
8     ports:
9       - '5432:5432'
10    environment:
11      POSTGRES_PASSWORD: pass1234
12
13   adminer:
14     image: adminer
15     restart: always
16     ports:
17       - 8080:8080
```

```
EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED        STATUS        PORTS                               NAMES
e064e995b68b   elasticsearch:7.4.2 "/usr/local/bin/dock..." 2 hours ago   Up 4 seconds   0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp   resources_elasticsearch_1

EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker stop e0
e0

EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED        STATUS        PORTS   NAMES

EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$
```

docker-compose -f /c/Works/PROJECTS/001_Workspace/intellijidea/SB_004_SpringBootTutorial/repository/spring-boot-with-tech-tutorial/sb-02-spring-module-examples/sb-03-spring-datajpa-postgresql/src/main/resources/docker-compose.yml up -d

```
EmreG1tkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker-compose -f /c/Works/PROJECTS/001_Workspace/intellijidea/SB_004_SpringBootTutorial/repository/spring-boot-with-tech-tutorial/sb-02-spring-module-examples/sb-03-spring-datajpa-postgresql/src/main/resources/docker-compose.yml up -d
Found orphan containers (resources_elasticsearch_1) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphan flag to clean it up.
Pulling db (postgres:...)
latest: Pulling from library/postgres
Digest: sha256:2d1e636f07781d4799b3f2edbff78a0a5494f24c4512cb56a83ebfd0e04ec074
Status: Downloaded newer image for postgres:latest
Pulling adminer (adminer:...)
latest: Pulling from library/adminer
Digest: sha256:9b171348a0278a3f63e3be48ecd0da57ea9ad006f9f22126c9549b435b9eede9
Status: Downloaded newer image for adminer:latest
Creating resources_db_1 ...
Creating resources_adminer_1 ...
Creating resources_adminer_1 ... done
Creating resources_db_1 ... done
```

```
EmreG1tkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
6d4ecef30df5   adminer    "entrypoint.sh docke..." About a minute ago Up About a minute 0.0.0.0:8080->8080/tcp resources_adminer_1
7f8cc8363ec5   postgres  "docker-entrypoint.s..." About a minute ago Up About a minute 5432/tcp resources_db_1
```

```
EmreG1tkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker-machine ip
192.168.99.102
```

İstersek şu komut ile tüm container'ları durdurabiliriz.

docker stop \$(docker ps -qa)

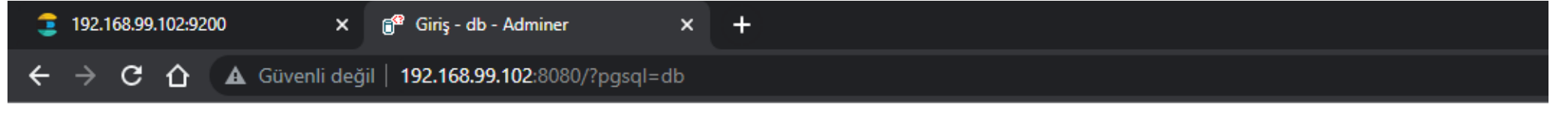
```
EmreG1tkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker stop $(docker ps -qa)
6d4ecef30df5
7f8cc8363ec5
e064e995b68b
```

Şimdi tekrar ayağı kaldıralım.

```
EmreG1tkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker-compose -f /c/Works/PROJECTS/001_Workspace/intellijidea/SB_004_SpringBootTutorial/repository/spring-boot-with-tech-tutorial/sb-02-spring-module-examples/sb-03-spring-datajpa-postgresql/src/main/resources/docker-compose.yml up -d
Found orphan containers (resources_elasticsearch_1) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphan flag to clean it up.
Starting resources_adminer_1 ...
Recreating resources_db_1 ...
Starting resources_adminer_1 ... done
Recreating resources_db_1 ... done

EmreG1tkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
b8b3b9a2c43a   postgres  "docker-entrypoint.s..." 12 seconds ago Up 4 seconds  0.0.0.0:5432->5432/tcp resources_db_1
6d4ecef30df5   adminer    "entrypoint.sh docke..." 2 hours ago    Up 5 seconds  0.0.0.0:8080->8080/tcp resources_adminer_1
```

<http://192.168.99.102:8080/>



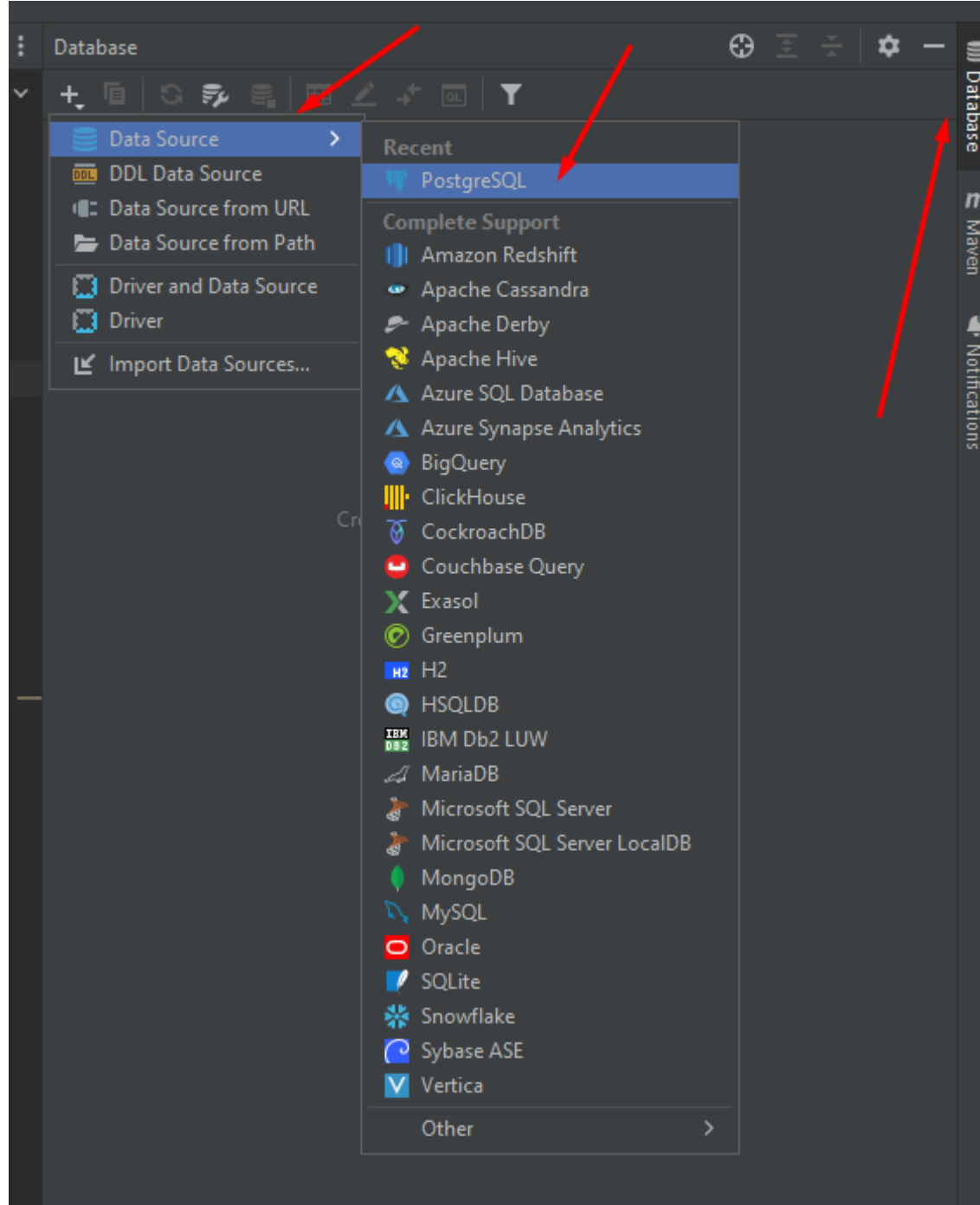
Adminer 4.8.1

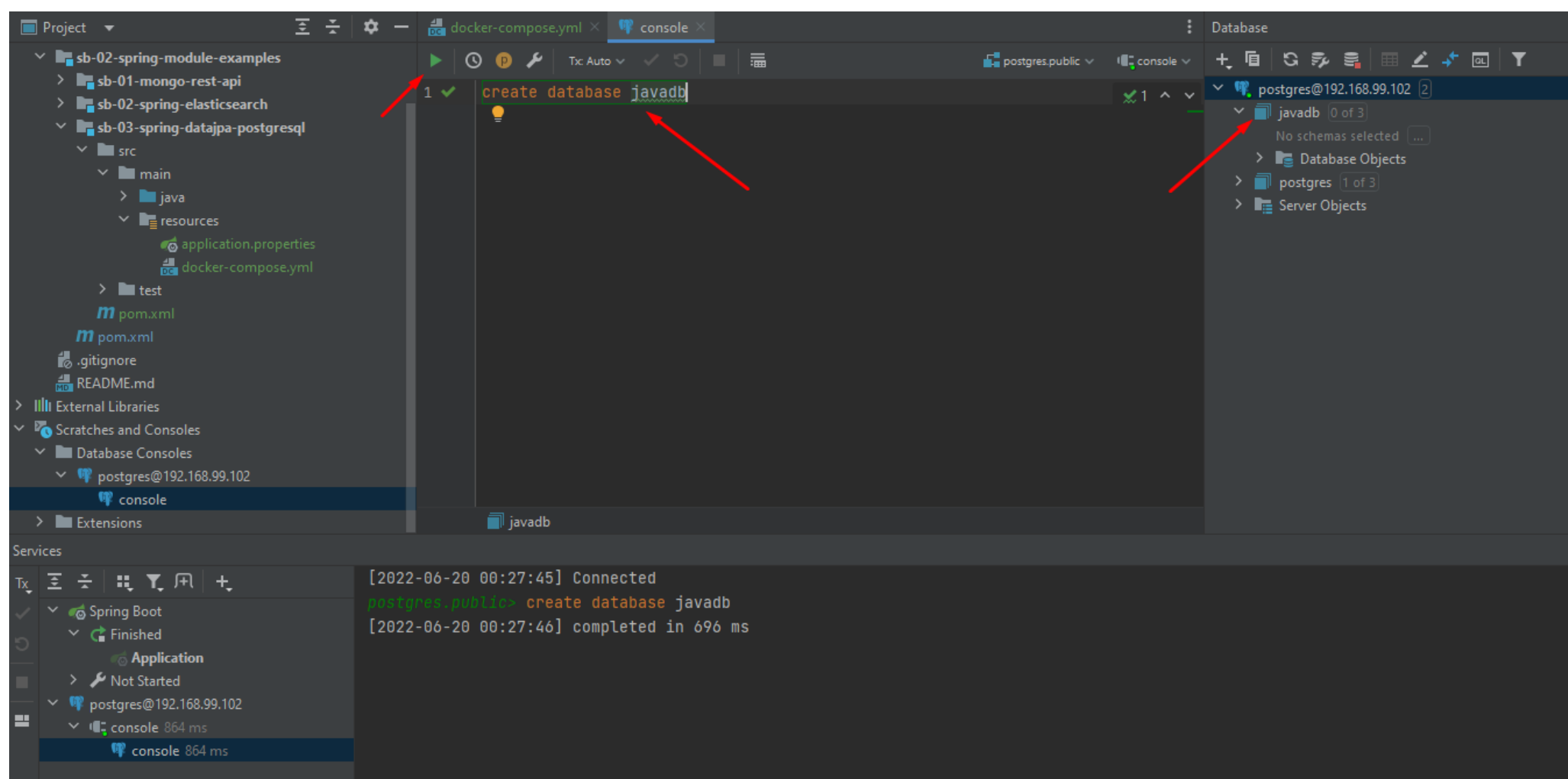
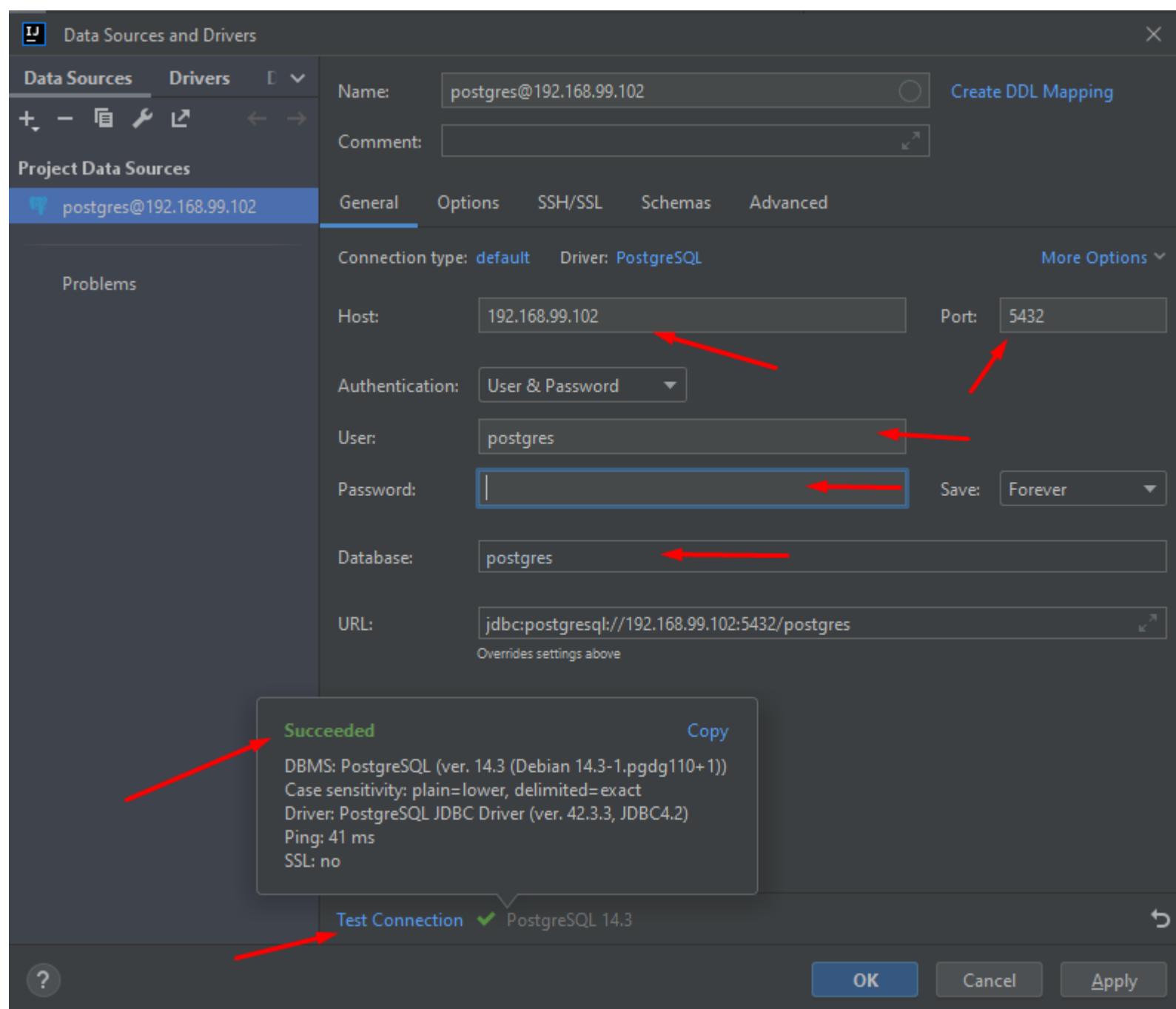
Giriş

Oturum başarıyla sonlandı. Adminer kullandığınız için teşekkür ederiz [bağış](#) yapmayı düşünün.

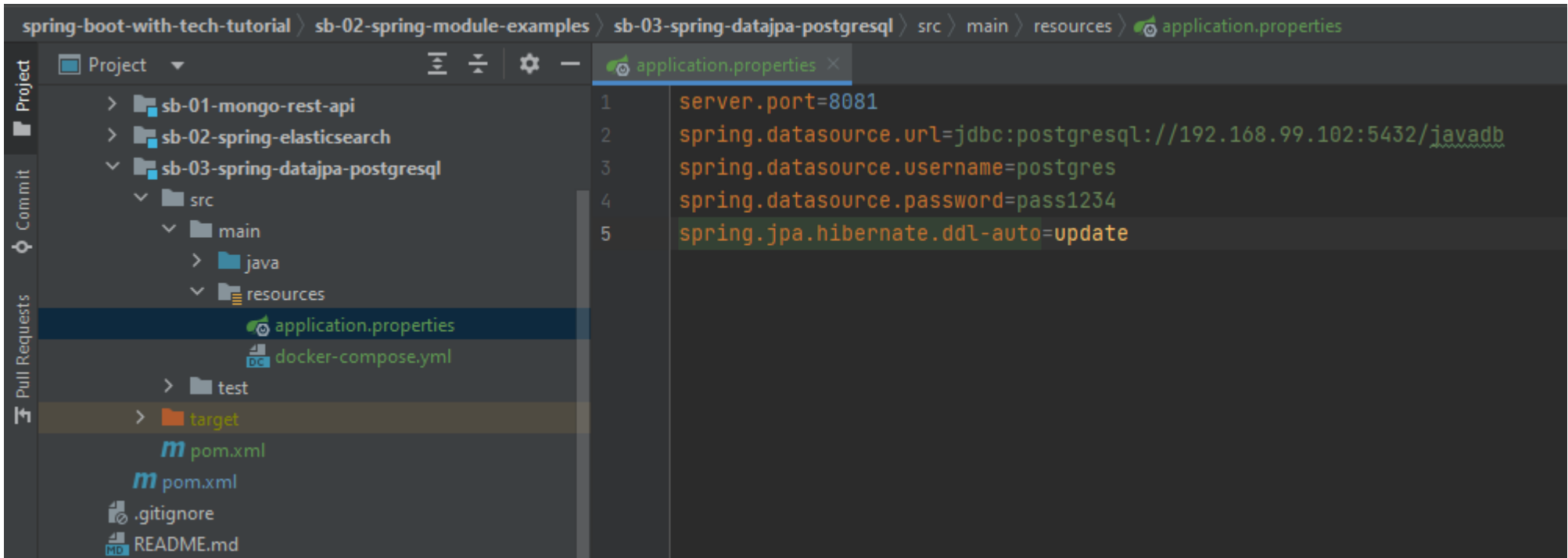
Sistem	PostgreSQL
Sunucu	db
Kullanıcı	postgres
Parola
Veri Tabanı	postgres

☐ Beni hatırla

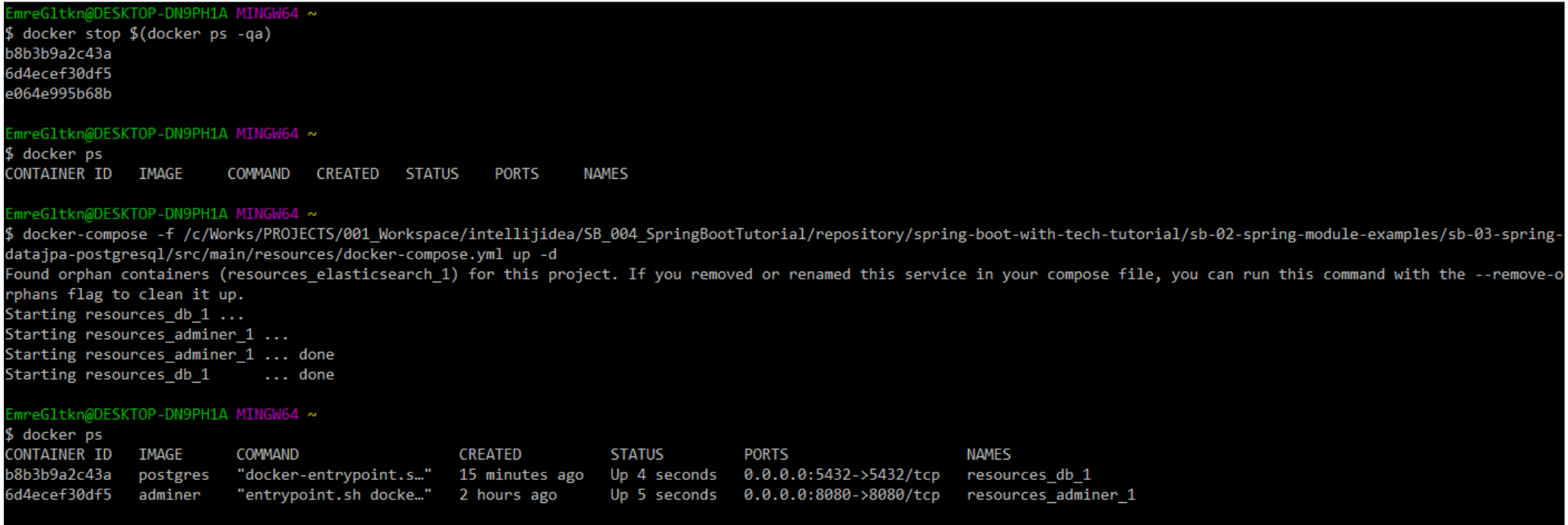
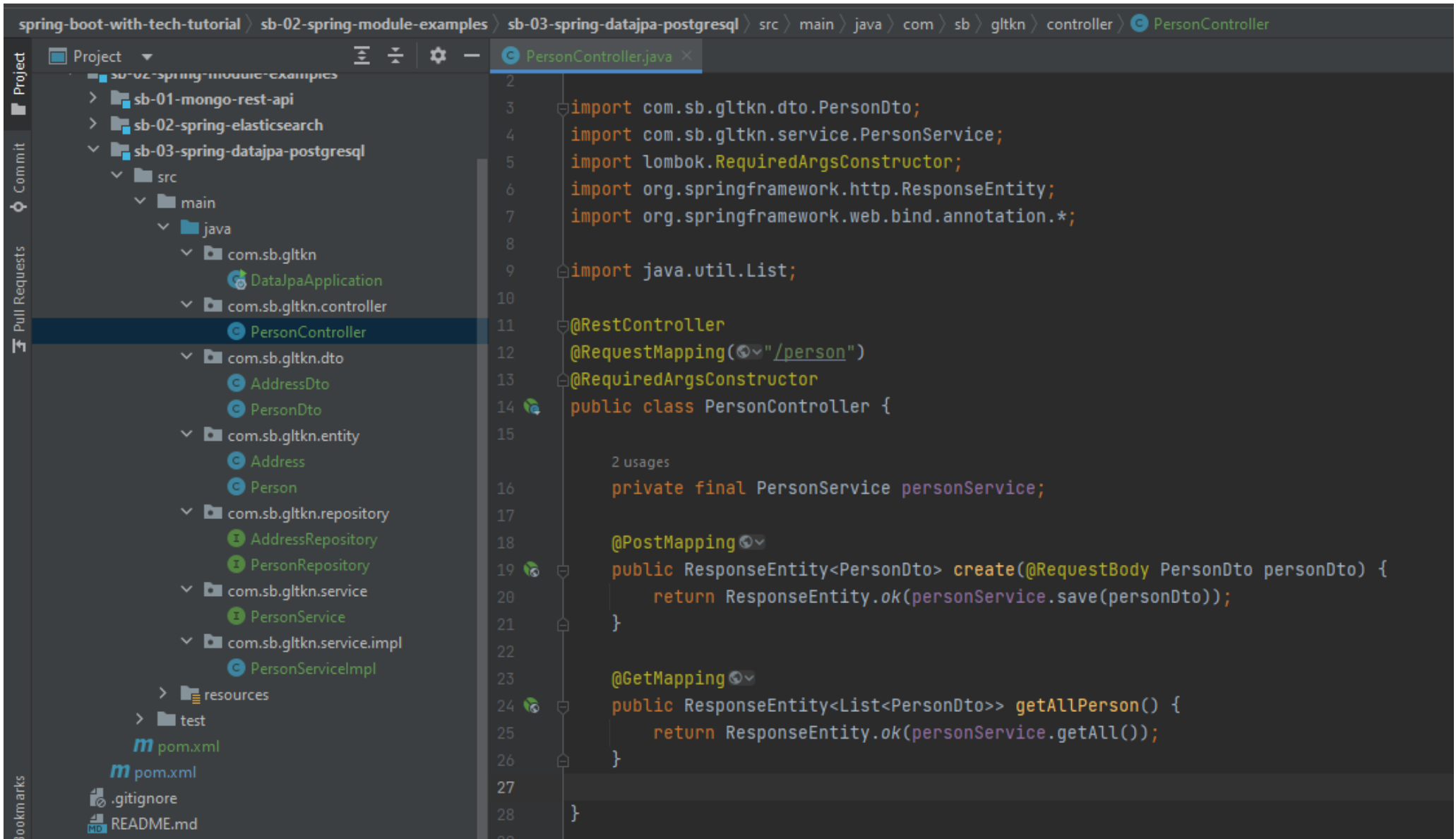




Şimdi bu oluşturduğumuz veritabanı ile application.properties dosyamızı güncelleyelim.



Projemizdeki API'yi oluşturduk. Tüm geliştirmeyi tamamladık.



Şimdi projemizi çalıştıralım ve postman'den request gönderelim.

<http://localhost:8081/person>

GET http://localhost:8081/person

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK

Şimdi veri kaydedelim.

```
{
  "name": "Emre",
  "surname": "Gltkn",
  "addresses": ["İstanbul", "Adana"]
}
```

POST http://localhost:8081/person

Body

```
1 {
2   "name": "Emre",
3   "surname": "Gltkn",
4   "addresses": ["İstanbul", "Adana"]
5 }
```

Status: 200 OK

Şimdi tekrar GET isteği yapalım.

GET http://localhost:8081/person

POST http://localhost:8081/person

+

...

▶ http://localhost:8081/person

GET

http://localhost:8081/person

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE	DESCRIPTION
	Key	Value	Description

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

[

{

"id": 2,

"name": "Emre",

"surname": "Gltkn",

"addresses": [

"İstanbul",

"Adana"

]

}

]

İntelij üzerinden tablomuzu kontrol edelim.

console_1

▶

⌚

P

⚙️

Txc Auto

✓

↺

■

📄

javadb.public

console_1

1

select * from person;

✓

Output

person

⏪

<

2 rows

>

⏩

↺

■

+

-

🔍

↑

Txc Auto

✓

↺

🌟

	id	name	surname
1	2	Emre	Gltkn
2	3	Emre	Gltkn