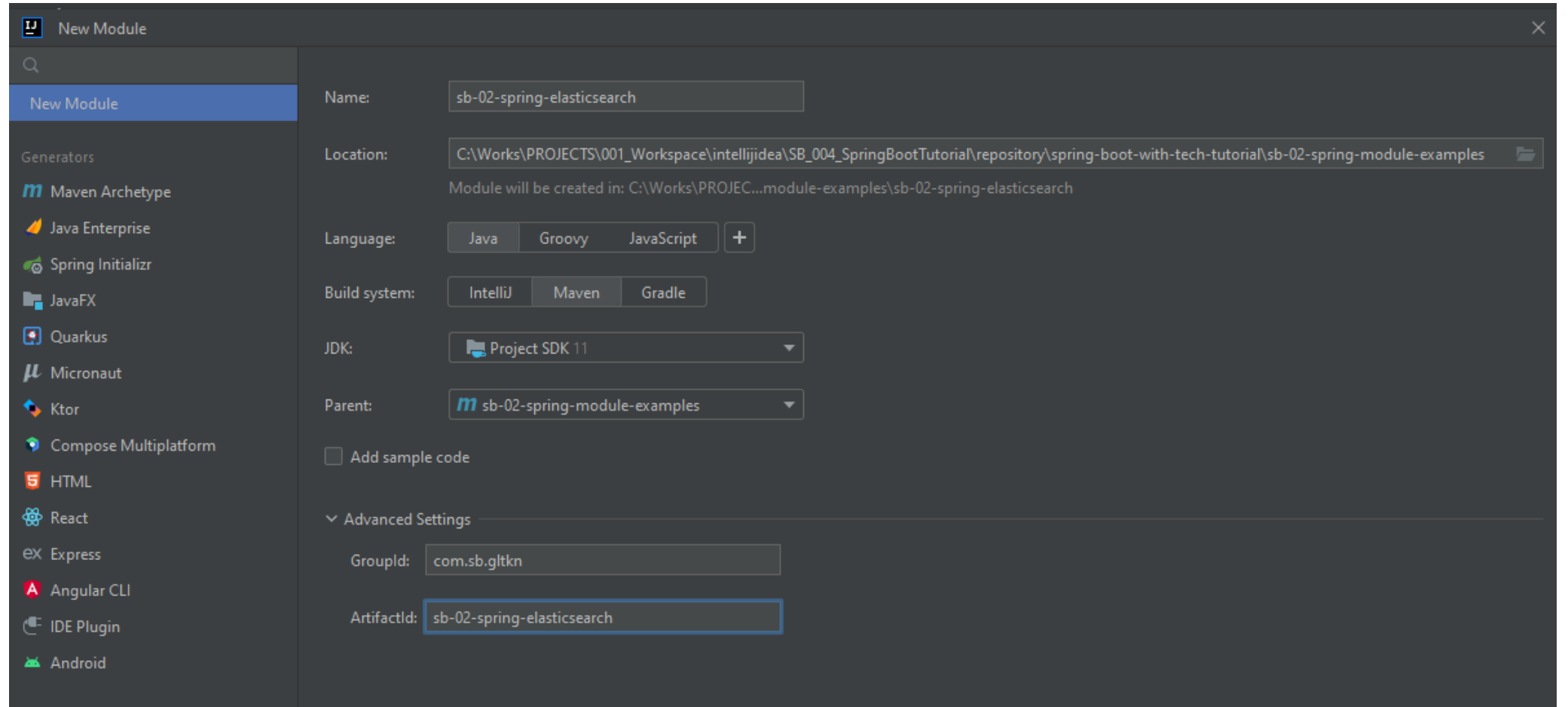
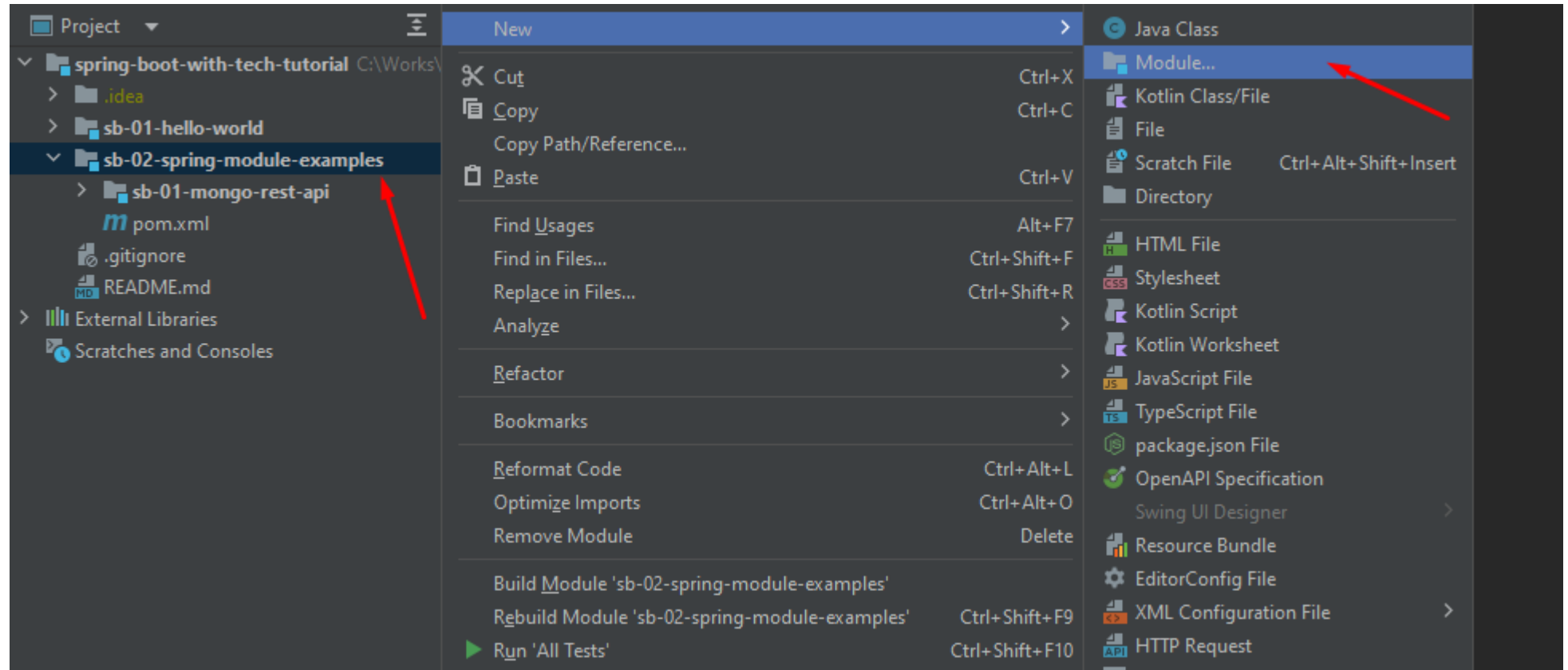
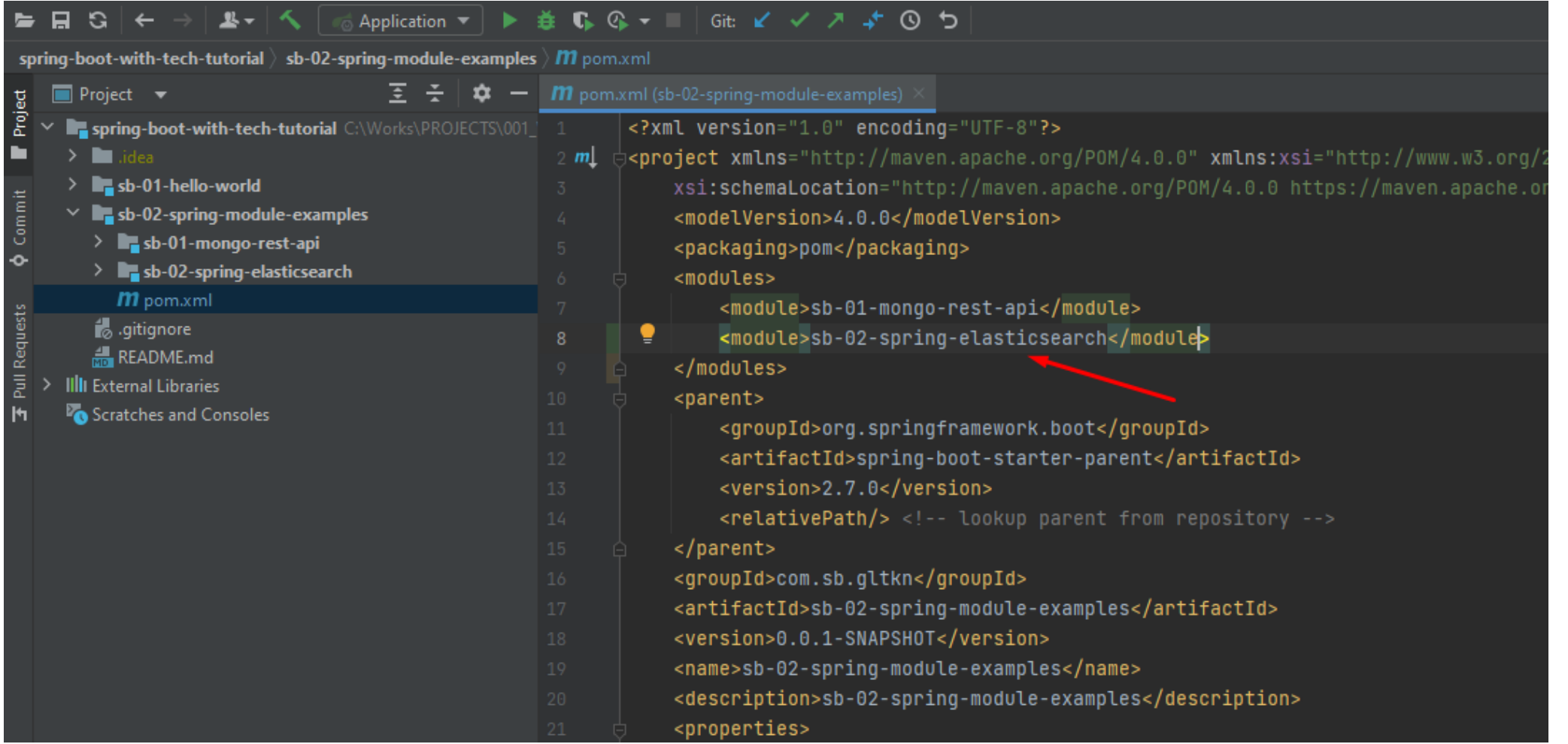


Projemizi aşağıdaki gibi oluşturalım.

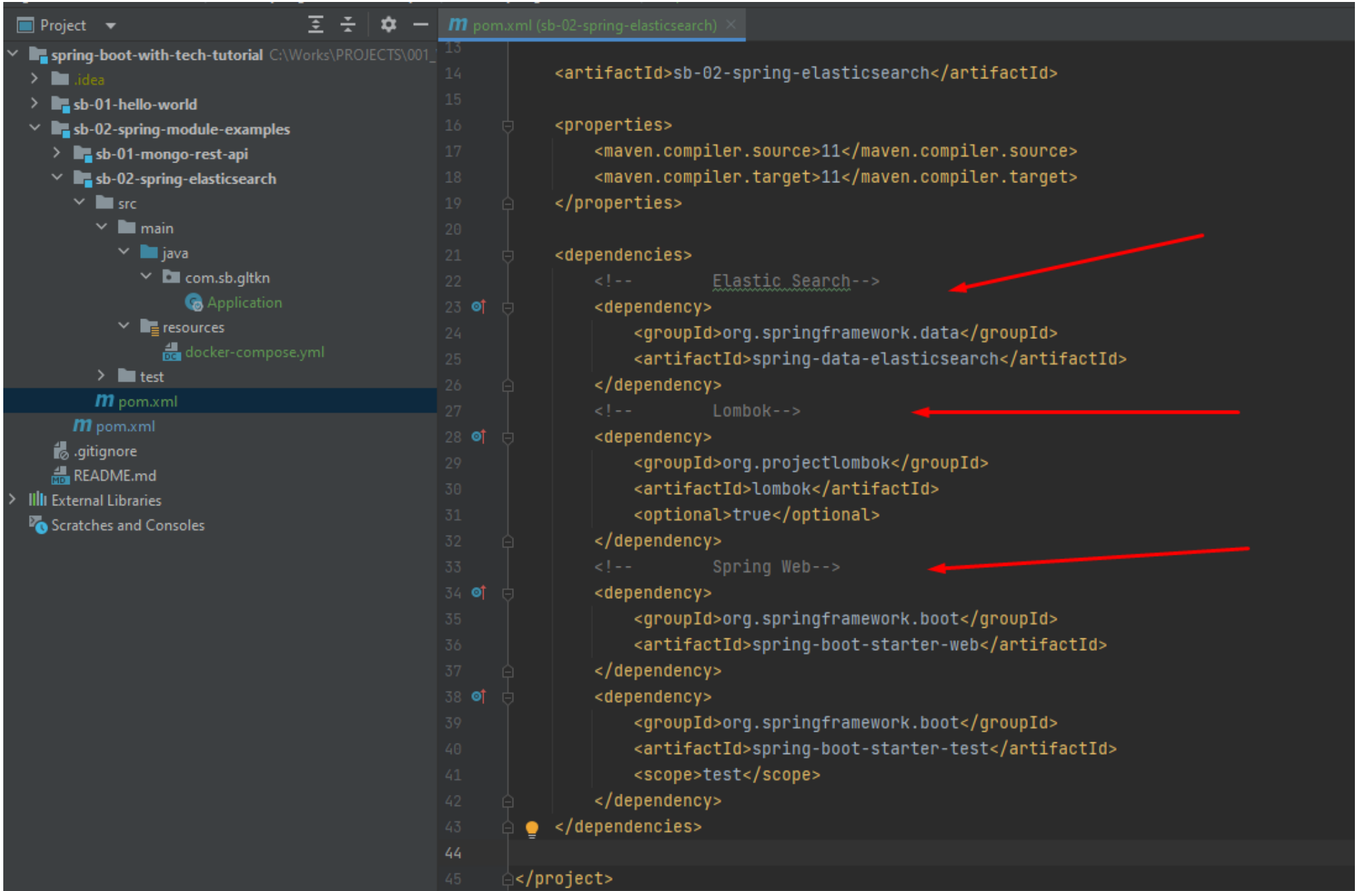


sb-02-spring-module-examples projesi'nin pom.xml'ine yeni eklediğimiz bu modül alttaki gibi otomatik eklenmiştir.

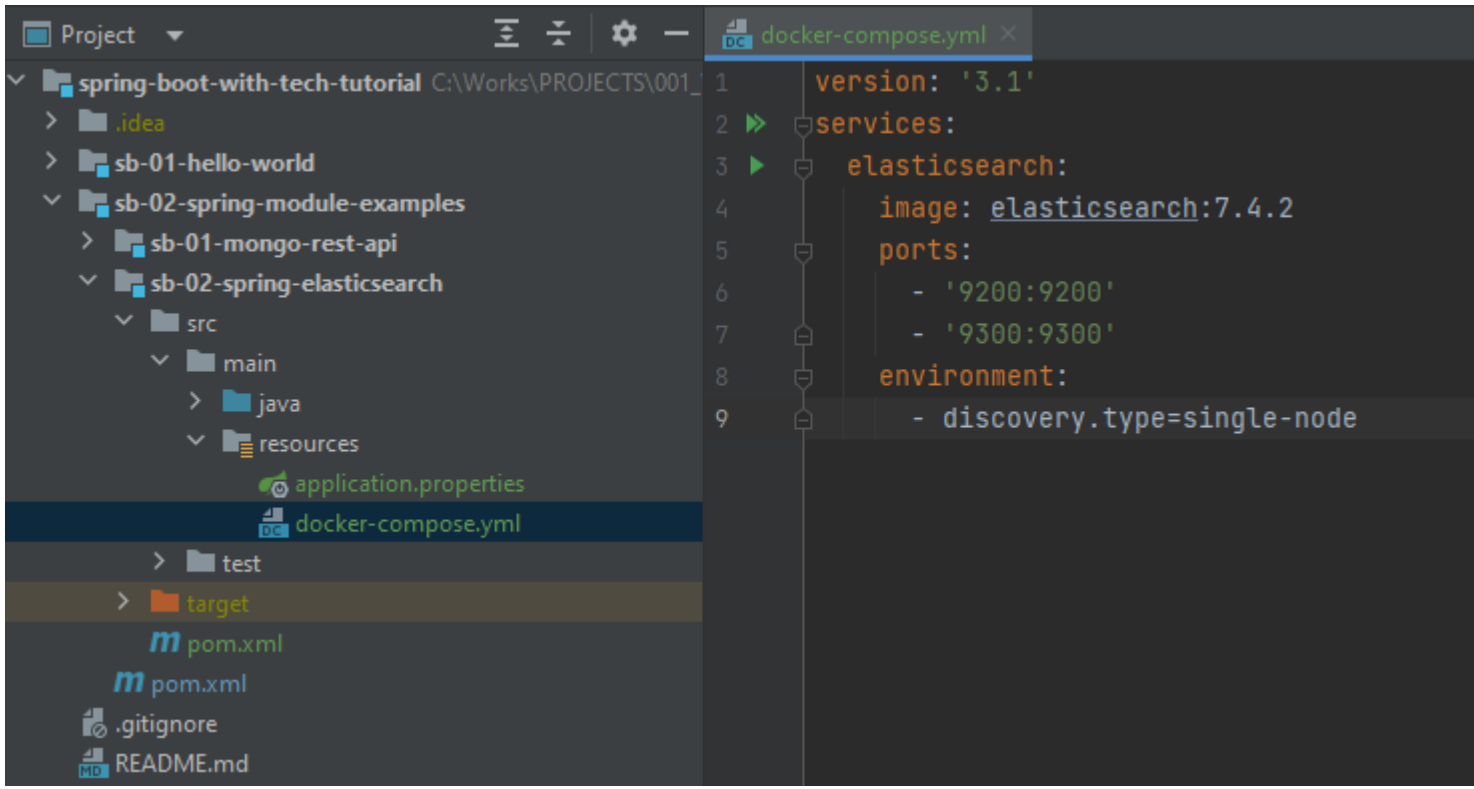


```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/maven-v4_0_0.xsd">
<modelVersion>4.0.0</modelVersion>
<packaging>pom</packaging>
<modules>
<module>sb-01-mongo-rest-api</module>
<module>sb-02-spring-elasticsearch</module>
</modules>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.7.0</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.sb.gltkn</groupId>
<artifactId>sb-02-spring-module-examples</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>sb-02-spring-module-examples</name>
<description>sb-02-spring-module-examples</description>
<properties>
```

Parent bilgisini güncelledik ve aşağıdaki 3 dependency'i ekledik.



```
<artifactId>sb-02-spring-elasticsearch</artifactId>
<properties>
<maven.compiler.source>11</maven.compiler.source>
<maven.compiler.target>11</maven.compiler.target>
</properties>
<dependencies>
<!-- Elastic Search-->
<dependency>
<groupId>org.springframework.data</groupId>
<artifactId>spring-data-elasticsearch</artifactId>
</dependency>
<!-- Lombok-->
<dependency>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
<optional>true</optional>
</dependency>
<!-- Spring Web-->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>
</project>
```



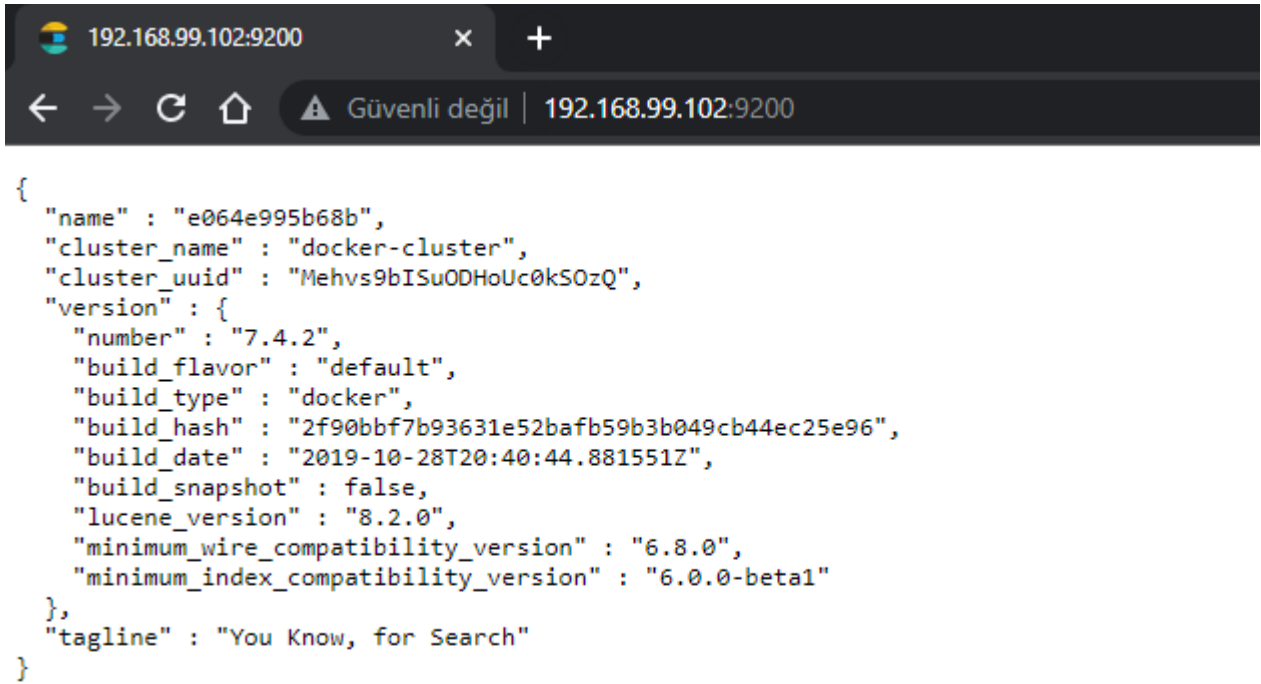
**docker-compose -f /c/Works/PROJECTS/001\_Workspace/intellijidea/SB\_004\_SpringBootTutorial/repository/spring-boot-with-tech-tutorial/sb-02-spring-module-examples/sb-02-spring-elasticsearch/src/main/resources/docker-compose.yml up -d**

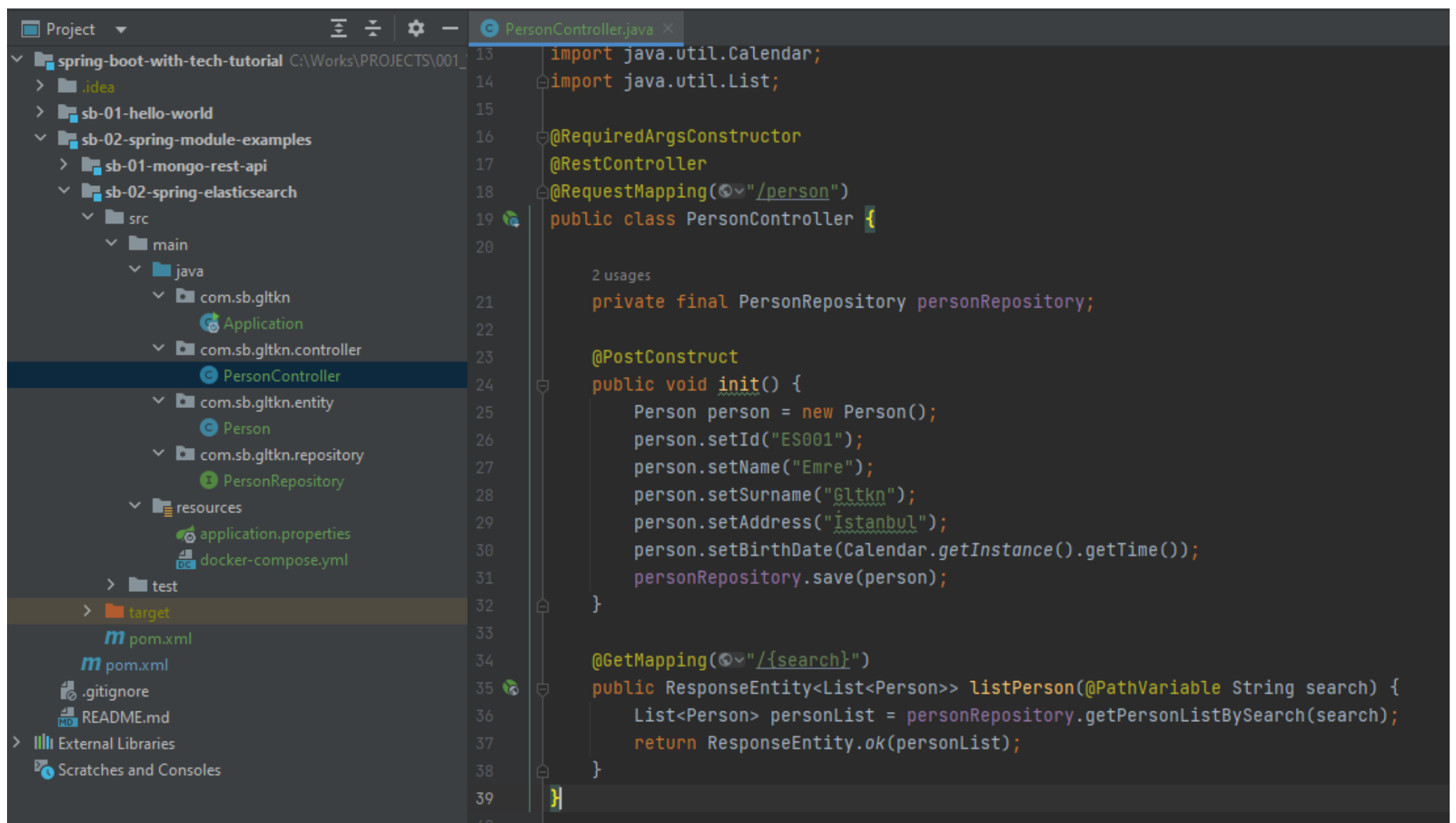
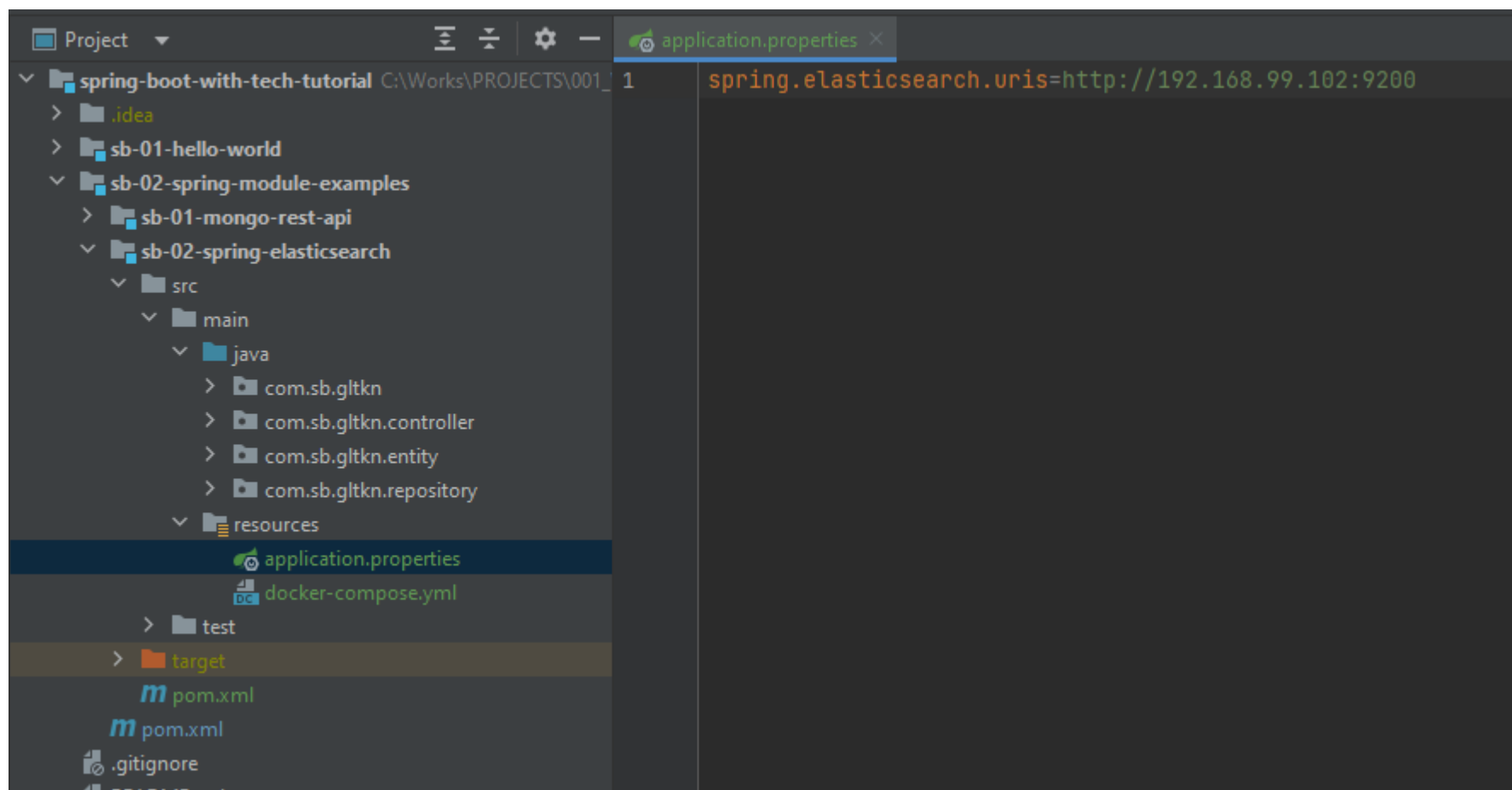
```
EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker-compose -f /c/Works/PROJECTS/001_Workspace/intellijidea/SB_004_SpringBootTutorial/repository/spring-boot-with-tech-tutorial/sb-02-spring-module-examples/sb-02-spring-elasticsearch/src/main/resources/docker-compose.yml up -d
Starting resources_elasticsearch_1 ...
Starting resources_elasticsearch_1 ... done
```

```
EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                               NAMES
059b151b50e0   elasticsearch:8.2.3   "/bin/tini -- /usr/l..." 13 minutes ago Up 24 seconds 0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp resources_elasticsearch_1
```

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

<http://192.168.99.102:9200/> linki üzerinden aşağıdaki gibi elasticsearch'ün çalıştığını görebiliriz.





GET /posts

POST User - post

GET Person-ElasticSearch

+

...

N

▶ Person-ElasticSearch

GET

▼

http://localhost:8080/person/emre

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettings

Query Params

	KEY	VALUE	DESCRIPTION
	Key	Value	Description

BodyCookiesHeaders (5)Test Results

PrettyRawPreviewVisualizeJSON

1

[

2

{

3

"id": "ES001",

4

"name": "Emre",

5

"surname": "Gltkn",

6

"address": "İstanbul",

7

"birthDate": "2022-06-19T19:10:16.100+00:00"

8

}

9

]

GET Person-ElasticSearch

+

...

N

▶ Person-ElasticSearch

GET

▼

http://localhost:8080/person/re

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettings

Query Params

	KEY	VALUE	DESCRIPTION
	Key	Value	Description

BodyCookiesHeaders (5)Test Results

PrettyRawPreviewVisualizeJSON

1

[