

Bu bölümde reactive stream kullanarak non-blocking webAPI geliştireceğiz.

<https://www.reactive-streams.org/>

Reactive Stream'lerin amacı, non-blocking geri basınç ile asenkron stream işleme için bir standart sağlamaktır.

<https://github.com/reactive-streams/reactive-streams-jvm>

API, Reactive Stream implementation'ları tarafından sağlanması gereken aşağıdaki bileşenlerden oluşur:

1. Publisher

```
public interface Publisher<T> {  
    public void subscribe(Subscriber<? super T> s);  
}
```

2. Subscriber

```
public interface Subscriber<T> {  
    public void onSubscribe(Subscription s);  
    public void onNext(T t);  
    public void onError(Throwable t);  
    public void onComplete();  
}
```

3. Subscription

```
public interface Subscription {  
    public void request(long n);  
    public void cancel();  
}
```

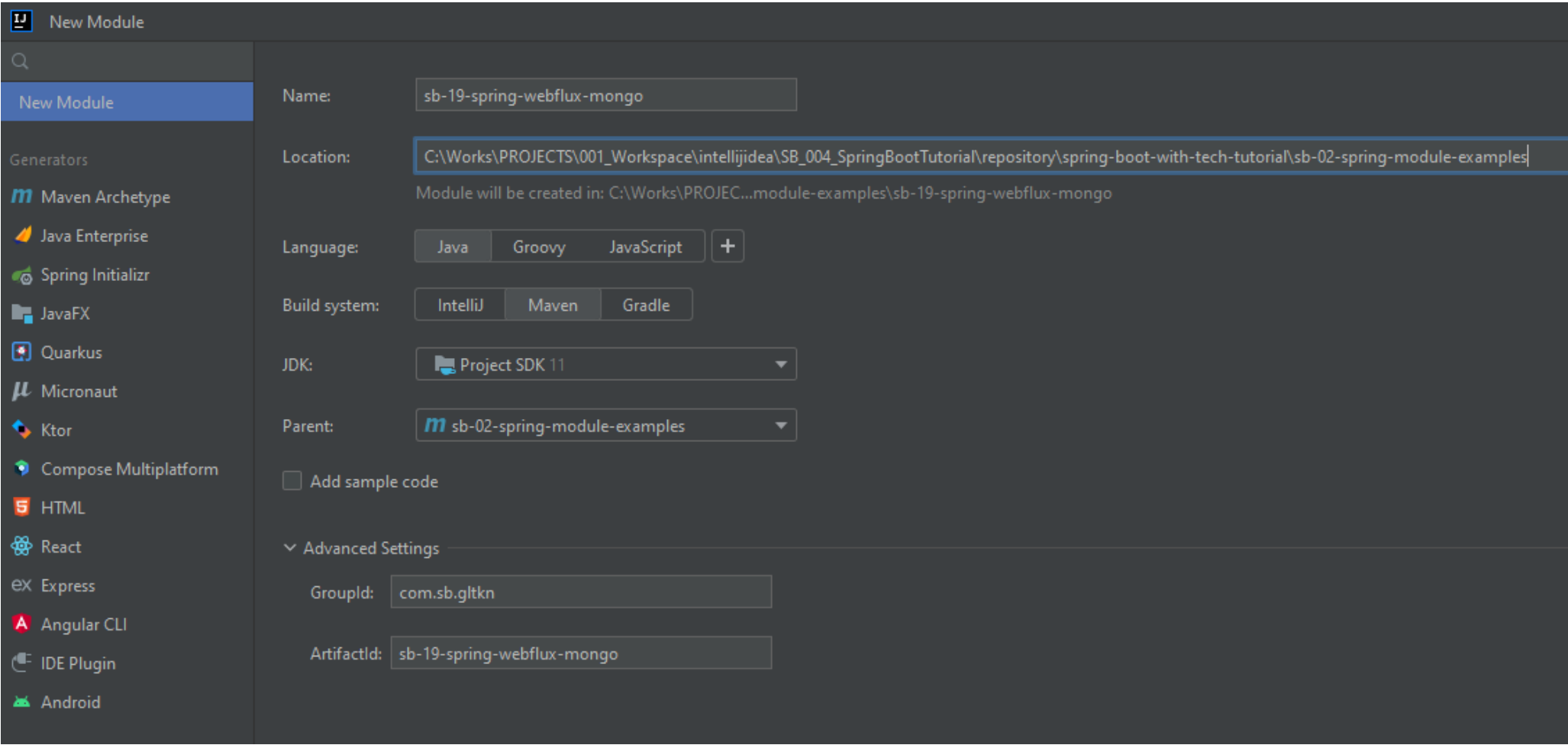
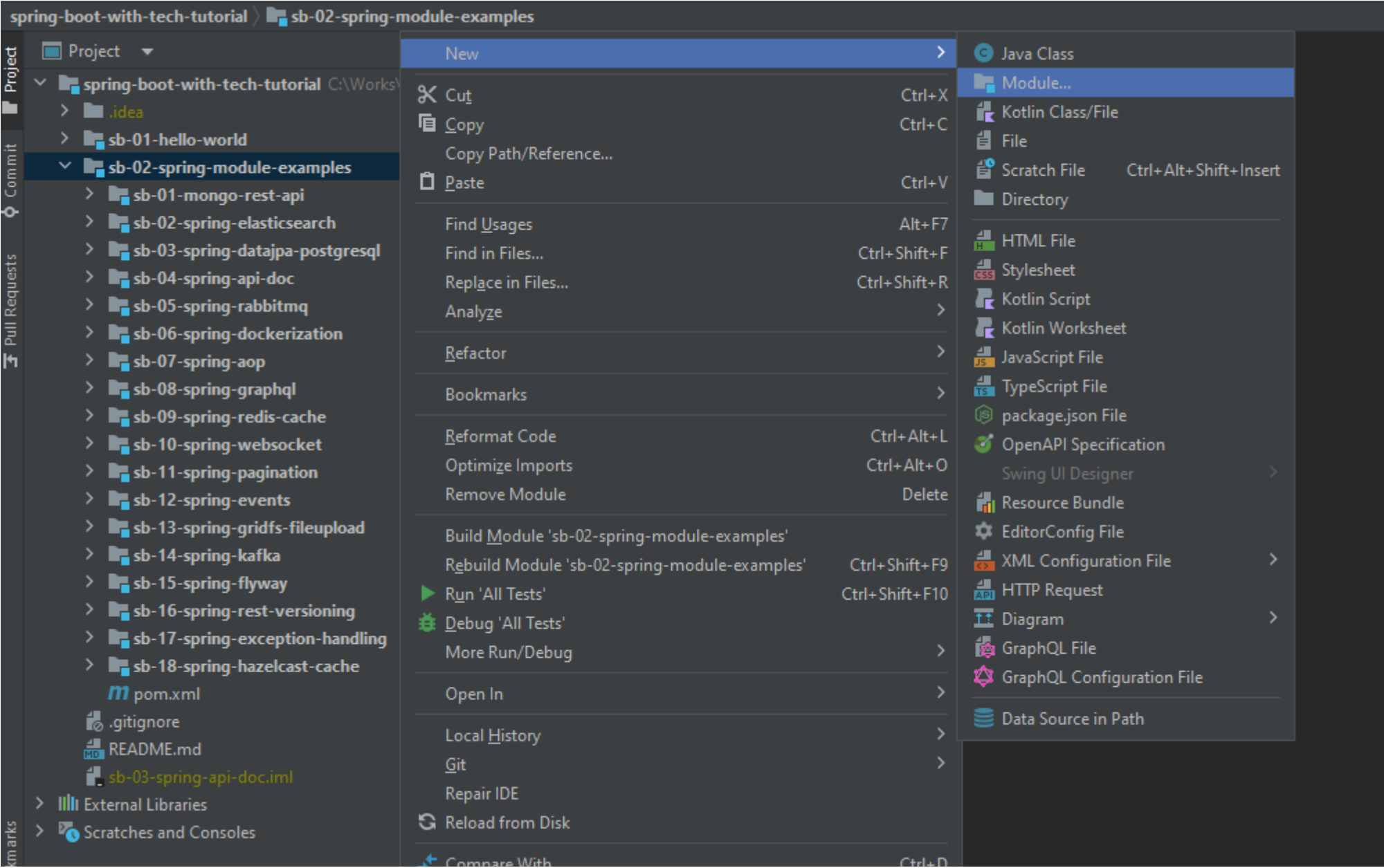
4. Processor

```
public interface Processor<T, R> extends Subscriber<T>, Publisher<R> {  
}
```

Spring WebFlux:

<https://docs.spring.io/spring-framework/docs/current/reference/html/web-reactive.html>

Projemizi oluřturalım.



```

6   <groupId>org.springframework.boot</groupId>
7   <artifactId>spring-boot-starter-parent</artifactId>
8   <version>2.7.0</version>
9   <relativePath/> <!-- lookup parent from repository -->
10  </parent>
11  <modelVersion>4.0.0</modelVersion>
12
13  <artifactId>sb-19-spring-webflux-mongo</artifactId>
14
15  <properties>
16    <maven.compiler.source>11</maven.compiler.source>
17    <maven.compiler.target>11</maven.compiler.target>
18  </properties>
19
20  <dependencies>
21    <dependency>
22      <groupId>org.springframework.boot</groupId>
23      <artifactId>spring-boot-starter-webflux</artifactId>
24    </dependency>
25    <dependency>
26      <groupId>org.springframework.boot</groupId>
27      <artifactId>spring-boot-starter-data-mongodb-reactive</artifactId>
28    </dependency>
29    <dependency>
30      <groupId>org.mongodb</groupId>
31      <artifactId>mongodb-driver-reactivestreams</artifactId>
32    </dependency>
33    <dependency>
34      <groupId>org.projectlombok</groupId>
35      <artifactId>lombok</artifactId>
36      <optional>true</optional>
37    </dependency>
38  </dependencies>

```

spring-boot-with-tech-tutorial > sb-02-spring-module-examples > sb-19-spring-webflux-mongo > src > main > resources > docker-compose.yml

Project

- spring-boot-with-tech-tutorial
  - .idea
  - sb-01-hello-world
  - sb-02-spring-module-examples
    - sb-01-mongo-rest-api
    - sb-02-spring-elasticsearch
    - sb-03-spring-datajpa-postgresql
    - sb-04-spring-api-doc
    - sb-05-spring-rabbitmq
    - sb-06-spring-dockerization
    - sb-07-spring-aop
    - sb-08-spring-graphql
    - sb-09-spring-redis-cache
    - sb-10-spring-websocket
    - sb-11-spring-pagination
    - sb-12-spring-events
    - sb-13-spring-gridfs-fileupload
    - sb-14-spring-kafka
    - sb-15-spring-flyway
    - sb-16-spring-rest-versioning
    - sb-17-spring-exception-handling
    - sb-18-spring-hazelcast-cache
    - sb-19-spring-webflux-mongo
      - src
        - main
          - resources
            - docker-compose.yml
        - test

Commit

Pull Requests

Bookmarks

Structure

```

1  version: '3.1'
2  services:
3    mongo:
4      image: mongo:4.4.6
5      ports:
6        - '27017:27017'
7      command: --serviceExecutor adaptive

```

Şimdi docker'ı çalıştıralım.  
Öncelikle VirtualBox üzerinden **default** isimli makineyi durdurdum.

```
EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker-machine start default
Starting "default"...
(default) Check network to re-create if needed...
(default) Windows might ask for the permission to configure a dhcp server. Sometimes, such confirmation window is minimized in the taskbar.
(default) Waiting for an IP...
Machine "default" was started.
Waiting for SSH to be available...
Detecting the provisioner...
Started machines may have new IP addresses. You may need to re-run the `docker-machine env` command.

EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker-machine env default
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.102:2376"
export DOCKER_CERT_PATH="C:\Users\EmreGltkn\.docker\machine\machines\default"
export DOCKER_MACHINE_NAME="default"
export COMPOSE_CONVERT_WINDOWS_PATHS="true"
# Run this command to configure your shell:
# eval "$(C:\Users\EmreGltkn\bin\docker-machine.exe env default)"
```

```
EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ eval "$(docker-machine env default)"

EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
6ecb6c40a477     postgres  "docker-entrypoint.s..."  2 months ago  Up 39 seconds  0.0.0.0:5432->5432/tcp  resources_db_1
```

```
EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker stop $(docker ps -qa)
6ecb6c40a477
7f8cab4b2d81
bfa4eb0bff7a
3fdffa2824a1
34d999770799
1fb2b9cc021a
289bbaf1f275
160d3d0ccef6
dbcbe2232246
173e825e0359

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

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

docker-compose -f /c/Works/PROJECTS/001\_Workspace/intellijidea/SB\_004\_SpringBootTutorial/repository/spring-boot-with-tech-tutorial/sb-02-spring-module-examples/sb-19-spring-webflux-mongo/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-19-spring-webflux-mongo/src/main/resources/docker-compose.yml up -d
Found orphan containers (resources_db_1) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.
Creating resources_mongo_1 ...
Creating resources_mongo_1 ... done
```

```
EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
9ebc6806b37a     mongo:4.4.6  "docker-entrypoint.s..."  30 seconds ago  Up 33 seconds  0.0.0.0:27017->27017/tcp  resources_mongo_1
```





```
1 package com.sb.gltkn.repository;
2
3 import com.sb.gltkn.entity.Employee;
4 import org.springframework.data.mongodb.repository.ReactiveMongoRepository;
5
6 4 usages
7 public interface EmployeeRepository extends ReactiveMongoRepository<Employee, String> {
8 }
```

```
3 import com.sb.gltkn.entity.Employee;
4 import com.sb.gltkn.repository.EmployeeRepository;
5 import lombok.RequiredArgsConstructor;
6 import org.springframework.web.bind.annotation.GetMapping;
7 import org.springframework.web.bind.annotation.PathVariable;
8 import org.springframework.web.bind.annotation.RequestMapping;
9 import org.springframework.web.bind.annotation.RestController;
10 import reactor.core.publisher.Flux;
11 import reactor.core.publisher.Mono;
12
13 @RestController
14 @RequestMapping("/employees")
15 @RequiredArgsConstructor
16 public class EmployeeController {
17
18     2 usages
19     private final EmployeeRepository employeeRepository;
20
21     @GetMapping("/{id}")
22     public Mono<Employee> getById(@PathVariable String id){
23         return employeeRepository.findById(id);
24     }
25
26     @GetMapping
27     public Flux<Employee> getAll(){
28         return employeeRepository.findAll();
29     }
30 }
```

Uygulama çalıştırıldığı esnada random oluşturulan 100 employee veritabanına kaydedilecektir.

```
package com.sb.gltkn;

import ...

@SpringBootApplication
public class Application {

    2 usages
    @Autowired
    private EmployeeRepository employeeRepository;

    public static void main(String[] args) { SpringApplication.run(Application.class, args); }

    @EventListener(ApplicationStartedEvent.class)
    public void appStart() {
        1
        if (employeeRepository.count().block() == 0) {
            IntStream.range(0, 100) IntStream
                .mapToObj(this::generate) Stream<Employee>
                .map(employeeRepository::save) Stream<Mono<Employee>>
                .collect(Collectors.toList()) List<Mono<Employee>>
                .forEach(item -> item.subscribe(System.out::println));
        }
    }

    1 usage
    private Employee generate(int i) {
        return Employee.builder()
            .name("Name"+ i)
            .lastname("Lastname"+ i)
            .birthDate(LocalDate.now())
            .build();
    }
}
```

**Data Sources and Drivers**

**Data Sources** **Drivers**

**Project Data Sources**

- mydb@192.168.99.102
- postgres@192.168.99.102

**Problems** 1

**General** Options SSH/SSL Schemas Advanced

Name: mydb@192.168.99.102 [Create DDL Mapping](#)

Comment:

Connection type: default Driver: MongoDB [More Options](#)

Host: 192.168.99.102 Port: 27017

Authentication: No auth

Database: mydb

URL: mongodb://192.168.99.102:27017/mydb  
Overrides settings above

**Succeeded** [Copy](#)

DBMS: Mongo DB (ver. 4.4.6)  
Case sensitivity: plain=mixed, delimited=mixed  
Driver: MongoDB JDBC Driver (ver. 1.16, JDBC4.2)  
Ping: 851 ms

[Test Connection](#) MongoDB 4.4.6

[OK](#) [Cancel](#) [Apply](#)

Code editor showing `Application.java` with `@SpringBootApplication` and `@Autowired` annotations. The database console shows the `employee` collection. The console output displays log messages for MongoDB connections and employee data.

```
package com.sb.gltn;

import ...

@SpringBootApplication
public class Application {

    2 usages
    @Autowired
    private EmployeeRepository employeeRepository;

    public static void main(String[] args) { SpringApplication.run(Application.class, args); }

    @EventListener(ApplicationStartedEvent.class)
```

Database console shows the `employee` collection.

Console output shows log messages for MongoDB connections and employee data:

```
2022-10-19 00:42:25.150 INFO 13552 --- [Thread-7] org.mongodb.driver.connection : opened connection [connectionId{localValue:11, serve
2022-10-19 00:42:25.168 INFO 13552 --- [Thread-8] org.mongodb.driver.connection : Opened connection [connectionId{localValue:12, serve
2022-10-19 00:42:25.266 INFO 13552 --- [Thread-7] org.mongodb.driver.connection : Opened connection [connectionId{localValue:13, serve
2022-10-19 00:42:25.284 INFO 13552 --- [Thread-7] org.mongodb.driver.connection : Opened connection [connectionId{localValue:14, serve
2022-10-19 00:42:25.336 INFO 13552 --- [Thread-8] org.mongodb.driver.connection : Opened connection [connectionId{localValue:16, serve
2022-10-19 00:42:25.336 INFO 13552 --- [Thread-7] org.mongodb.driver.connection : Opened connection [connectionId{localValue:15, serve
2022-10-19 00:42:25.483 INFO 13552 --- [Thread-9] org.mongodb.driver.connection : Opened connection [connectionId{localValue:17, serve
Employee(id=634f1dc0cbd2bd24ca50407e, name=Name13, lastname=Lastname13, birthDate=2022-10-19)
Employee(id=634f1dc0cbd2bd24ca504072, name=Name1, lastname=Lastname1, birthDate=2022-10-19)
2022-10-19 00:42:25.495 INFO 13552 --- [Thread-8] org.mongodb.driver.connection : Opened connection [connectionId{localValue:18, serve
Employee(id=634f1dc0cbd2bd24ca504081, name=Name16, lastname=Lastname16, birthDate=2022-10-19)
Employee(id=634f1dc0cbd2bd24ca504082, name=Name17, lastname=Lastname17, birthDate=2022-10-19)
Employee(id=634f1dc0cbd2bd24ca50407f, name=Name14, lastname=Lastname14, birthDate=2022-10-19)
Employee(id=634f1dc0cbd2bd24ca504080, name=Name15, lastname=Lastname15, birthDate=2022-10-19)
Employee(id=634f1dc0cbd2bd24ca504085, name=Name20, lastname=Lastname20, birthDate=2022-10-19)
Employee(id=634f1dc0cbd2bd24ca504086, name=Name21, lastname=Lastname21, birthDate=2022-10-19)
```

Code editor showing the `select * from employee;` query. The database console shows the `employee` collection. The console output displays the query result in JSON format.

```
select * from employee;
```

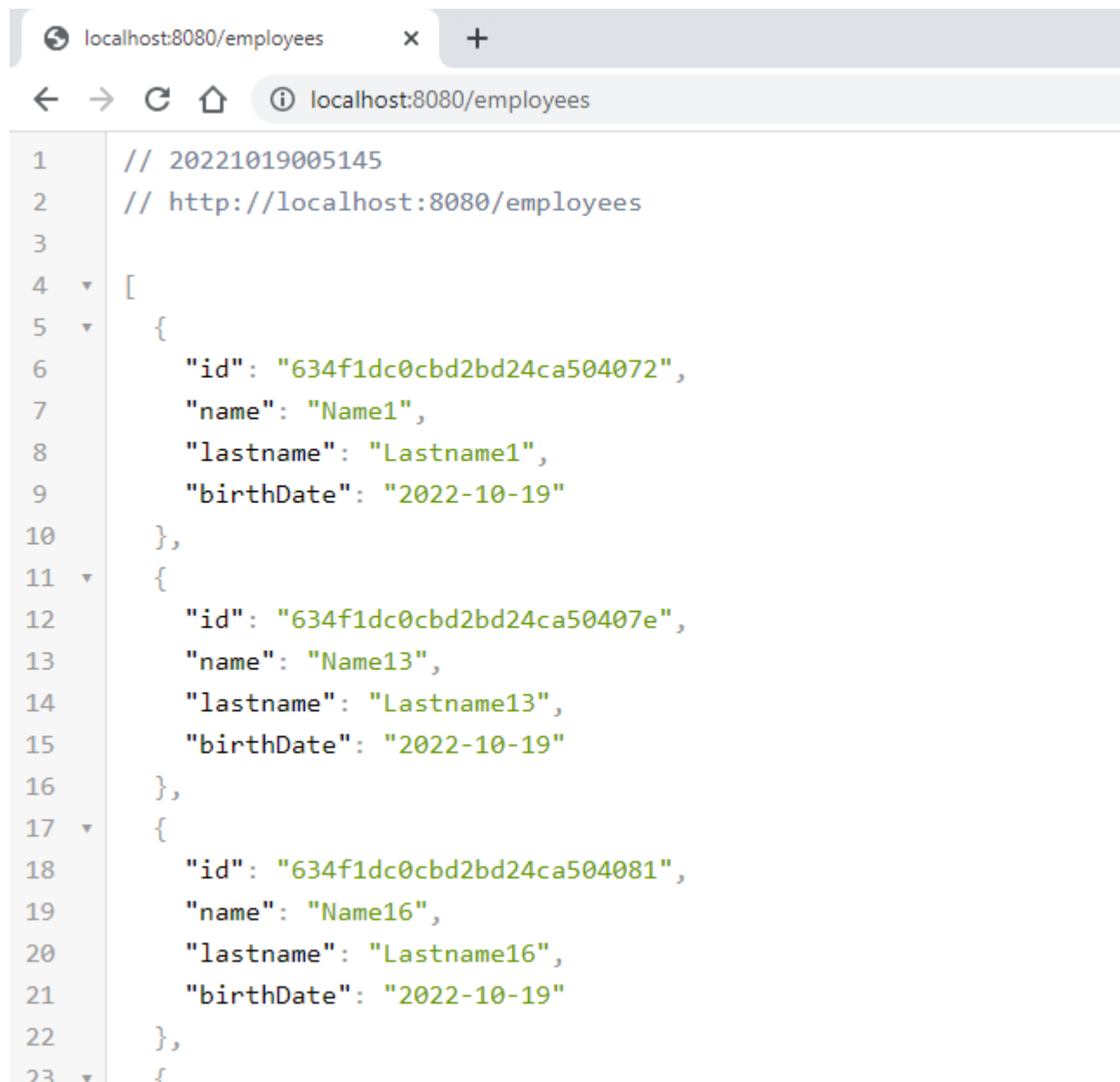
Database console shows the `employee` collection.

Console output shows the query result in JSON format:

_id	_class	birthDate	lastname	name
634f1dc0cbd2bd24ca504072	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname1	Name1
634f1dc0cbd2bd24ca50407e	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname13	Name13
634f1dc0cbd2bd24ca504081	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname16	Name16
634f1dc0cbd2bd24ca504082	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname17	Name17
634f1dc0cbd2bd24ca50407f	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname14	Name14
634f1dc0cbd2bd24ca504080	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname15	Name15
634f1dc0cbd2bd24ca504085	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname20	Name20
634f1dc0cbd2bd24ca504086	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname21	Name21
634f1dc0cbd2bd24ca504087	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname22	Name22
634f1dc0cbd2bd24ca504089	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname24	Name24
634f1dc0cbd2bd24ca504088	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname23	Name23
634f1dc0cbd2bd24ca50408a	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname25	Name25
634f1dc0cbd2bd24ca504083	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname18	Name18
634f1dc0cbd2bd24ca50408c	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname27	Name27
634f1dc0cbd2bd24ca504084	com.sb.gltn.entity.Employee	2022-10-18T21:00:00.000Z	Lastname19	Name19

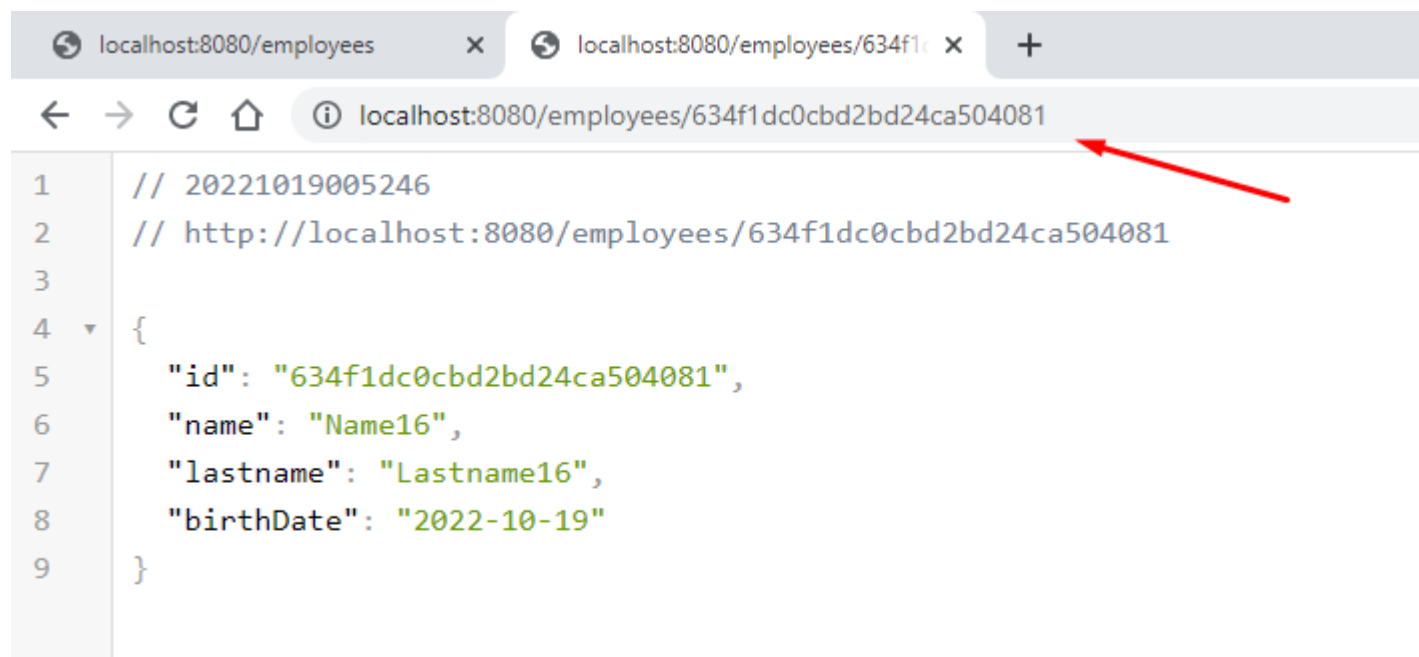


<http://localhost:8080/employees>



```
1 // 20221019005145
2 // http://localhost:8080/employees
3
4 [
5   {
6     "id": "634f1dc0cbd2bd24ca504072",
7     "name": "Name1",
8     "lastname": "Lastname1",
9     "birthDate": "2022-10-19"
10  },
11  {
12    "id": "634f1dc0cbd2bd24ca50407e",
13    "name": "Name13",
14    "lastname": "Lastname13",
15    "birthDate": "2022-10-19"
16  },
17  {
18    "id": "634f1dc0cbd2bd24ca504081",
19    "name": "Name16",
20    "lastname": "Lastname16",
21    "birthDate": "2022-10-19"
22  },
23  {
```

<http://localhost:8080/employees/634f1dc0cbd2bd24ca504081>



```
1 // 20221019005246
2 // http://localhost:8080/employees/634f1dc0cbd2bd24ca504081
3
4 {
5   "id": "634f1dc0cbd2bd24ca504081",
6   "name": "Name16",
7   "lastname": "Lastname16",
8   "birthDate": "2022-10-19"
9 }
```