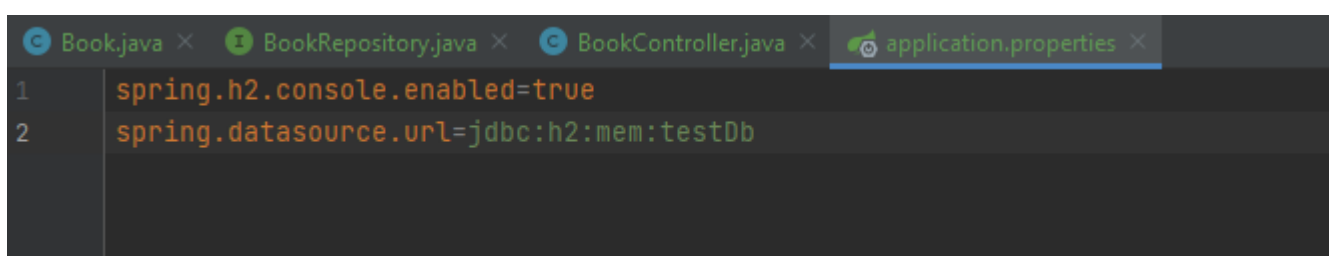
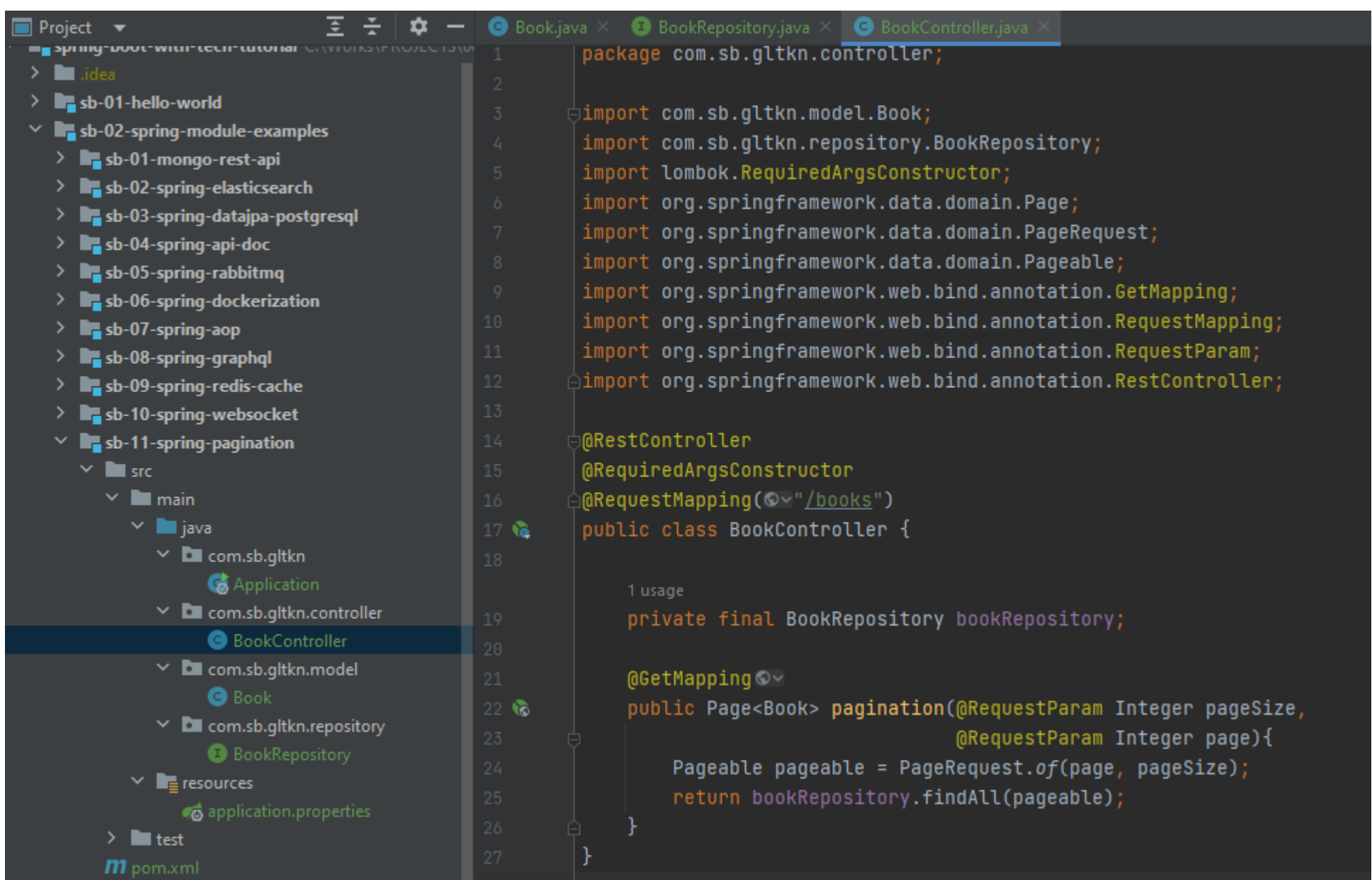
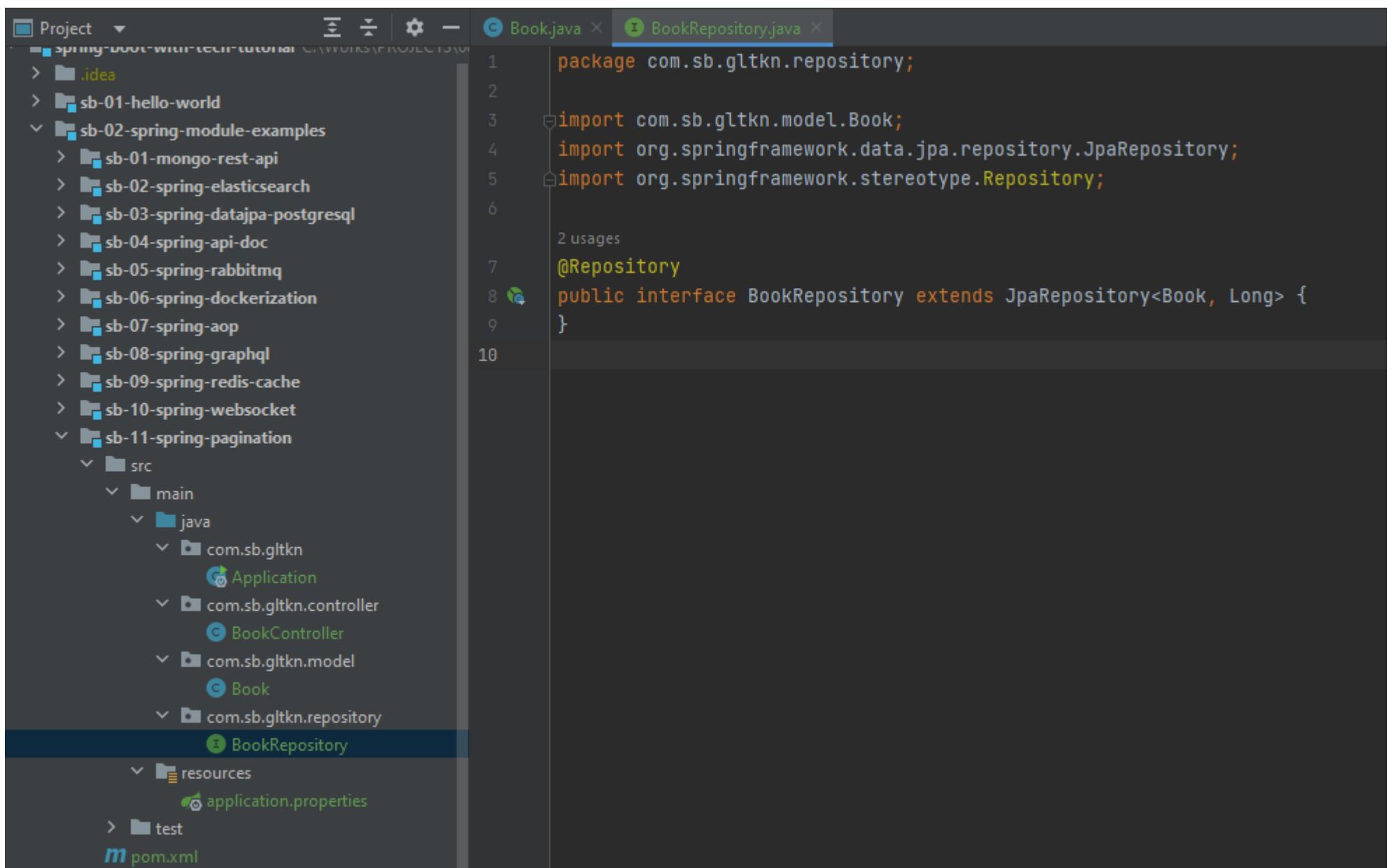



```
8      <version>2.7.0</version>
9      <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <modelVersion>4.0.0</modelVersion>
12
13    <artifactId>sb-11-spring-pagination</artifactId>
14
15    <properties>
16      <maven.compiler.source>11</maven.compiler.source>
17      <maven.compiler.target>11</maven.compiler.target>
18    </properties>
19    <dependencies>
20      <dependency>
21        <groupId>org.springframework.boot</groupId>
22        <artifactId>spring-boot-starter-web</artifactId>
23      </dependency>
24      <dependency>
25        <groupId>org.springframework.boot</groupId>
26        <artifactId>spring-boot-starter-data-jpa</artifactId>
27      </dependency>
28      <dependency>
29        <groupId>org.projectlombok</groupId>
30        <artifactId>lombok</artifactId>
31        <version>1.18.24</version>
32      </dependency>
33      <dependency>
34        <groupId>com.h2database</groupId>
35        <artifactId>h2</artifactId>
36        <version>1.4.197</version>
37        <scope>runtime</scope>
38      </dependency>
39    </dependencies>
```

```
1  package com.sb.gltkn.model;
2
3  import lombok.Data;
4  import lombok.EqualsAndHashCode;
5
6  import javax.persistence.Entity;
7  import javax.persistence.Id;
8  import javax.persistence.Table;
9  import java.time.LocalDate;
10
11  @Entity
12  @Data
13  @EqualsAndHashCode(of = "id")
14  @Table(name = "books")
15  public class Book {
16
17      @Id
18      private Long id;
19      private String isbn;
20      private String title;
21      private String author;
22      private LocalDate publishDate;
23  }
24
```



<http://localhost:8080/books?pageSize=5&page=1>

```
← → ↺ ⬆ ⓘ localhost:8080/books?pageSize=5&page=1
1 // 20220628001220
2 // http://localhost:8080/books?pageSize=5&page=1
3
4 {
5   "content": [
6
7   ],
8   "pageable": {
9     "sort": {
10      "sorted": false,
11      "unsorted": true,
12      "empty": true
13    },
14    "pageNumber": 1,
15    "pageSize": 5,
16    "offset": 5,
17    "unpaged": false,
18    "paged": true
19  },
20  "last": true,
21  "totalPages": 0,
22  "totalElements": 0,
23  "first": false,
24  "numberOfElements": 0,
25  "size": 5,
26  "number": 1,
27  "sort": {
28    "sorted": false,
29    "unsorted": true,
30    "empty": true
31  },
32  "empty": true
33 }
```

```
Console Actuator
main] com.sb.gltkn.Application : Starting Application using Java 11.0.15 on DESKTOP-DN9PH1A with PID 4896 (C:\Works\PROJECTS\001_Workspac
main] com.sb.gltkn.Application : No active profile set, falling back to 1 default profile: "default"
main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 45 ms. Found 1 JPA repository interfaces.
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.63]
main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1501 ms
main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
main] o.s.b.a.h2.H2ConsoleAutoConfiguration : H2 console available at '/h2-console'. Database available at 'jdbc:h2:mem:testDb'
main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.9.Final
main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.No
main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view
```


← → ↺ 🏠 ⓘ localhost:8080/h2-console/login.jsp?jsessionId=0a078874695acfaf1eea19e6ae777f13

English ▾ Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded) ▾

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testDb

User Name: sa

Password:

Connect Test Connection

← → ↺ 🏠 ⓘ localhost:8080/h2-console/login.do?jsessionId=0a078874695acfaf1eea19e6ae777f13

🔗 🧩 ☒ Auto commit 🔄 📄 | Max rows: 1000 ▾ ⏮ ⏭ ⏮ | 📄 | Auto complete Off ▾ Auto select On ▾ ?

📁 jdbc:h2:mem:testDb

📄 BOOKS ←

⊕ ID

⊕ AUTHOR

⊕ ISBN

⊕ PUBLISH_DATE

⊕ TITLE

⊕ Indexes

📁 INFORMATION_SCHEMA

👤 Users

📄 H2 1.4.197 (2018-03-18)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM BOOKS

SELECT * FROM BOOKS;
ID AUTHOR ISBN PUBLISH_DATE TITLE
(no rows, 8 ms)

Edit

Veri ekleyelim.

← → ↺ 🏠 ⓘ localhost:8080/h2-console/login.do?jsessionId=0a078874695acfaf1eea19e6ae777f13

🔗 🧩 ☒ Auto commit 🔄 📄 | Max rows: 1000 ▾ ⏮ ⏭ ⏮ | 📄 | Auto complete Off ▾ Auto select On ▾ ?

📁 jdbc:h2:mem:testDb

📄 BOOKS

⊕ ID

⊕ AUTHOR

⊕ ISBN

⊕ PUBLISH_DATE

⊕ TITLE

⊕ Indexes

📁 INFORMATION_SCHEMA

👤 Users

📄 H2 1.4.197 (2018-03-18)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM BOOKS

SELECT * FROM BOOKS;
ID AUTHOR ISBN PUBLISH_DATE TITLE
1 user1 isbn1 2001-01-01 title1
2 user2 isbn2 2001-01-02 title2
3 user3 isbn3 2001-01-03 title3
(3 rows, 3 ms)

Edit

```
1 // 20220628003051
2 // http://localhost:8080/books?pageSize=5&page=0
3
4 {
5   "content": [
6     {
7       "id": 1,
8       "isbn": "isbn1",
9       "title": "title1",
10      "author": "user1",
11      "publishDate": "2001-01-01"
12    },
13    {
14      "id": 2,
15      "isbn": "isbn2",
16      "title": "title2",
17      "author": "user2",
18      "publishDate": "2001-01-02"
19    },
20    {
21      "id": 3,
22      "isbn": "isbn3",
23      "title": "title3",
24      "author": "user3",
25      "publishDate": "2001-01-03"
26    }
27  ],
28  "pageable": {
29    "sort": {
```

```
28    "pageable": {
29      "sort": {
30        "empty": true,
31        "sorted": false,
32        "unsorted": true
33      },
34      "offset": 0,
35      "pageNumber": 0,
36      "pageSize": 5,
37      "paged": true,
38      "unpaged": false
39    },
40    "last": true,
41    "totalElements": 3,
42    "totalPages": 1,
43    "size": 5,
44    "sort": {
45      "empty": true,
46      "sorted": false,
47      "unsorted": true
48    },
49    "first": true,
50    "numberOfElements": 3,
51    "number": 0,
52    "empty": false
53  }
```

<http://localhost:8080/books?pageSize=2&page=0>

```
localhost:8080/books?pageSize=2&page=0

1 // 20220628003242
2 // http://localhost:8080/books?pageSize=2&page=0
3
4 {
5   "content": [
6     {
7       "id": 1,
8       "isbn": "isbn1",
9       "title": "title1",
10      "author": "user1",
11      "publishDate": "2001-01-01"
12    },
13    {
14      "id": 2,
15      "isbn": "isbn2",
16      "title": "title2",
17      "author": "user2",
18      "publishDate": "2001-01-02"
19    }
20  ],
21  "pageable": {
22    "sort": {
23      "empty": true,
24      "sorted": false,
25      "unsorted": true
26    },
27    "offset": 0,
28    "pageNumber": 0,
29    "pageSize": 2,
30    "paged": true,
31    "unpaged": false
32  },
33  "last": false,
34  "totalElements": 3,
35  "totalPages": 2,
36  "size": 2,
37  "sort": {
38    "empty": true,
39    "sorted": false,
40    "unsorted": true
41  },
42  "first": true,
43  "numberOfElements": 2,
44  "number": 0,
45  "empty": false
46 }
```

<http://localhost:8080/books?pageSize=2&page=1>

← → ↻ 🏠 ⓘ localhost:8080/books?pageSize=2&page=1

```
1 // 20220628003416
2 // http://localhost:8080/books?pageSize=2&page=1
3
4 {
5   "content": [
6     {
7       "id": 3,
8       "isbn": "isbn3",
9       "title": "title3",
10      "author": "user3",
11      "publishDate": "2001-01-03"
12    }
13  ],
14  "pageable": {
15    "sort": {
16      "empty": true,
17      "sorted": false,
18      "unsorted": true
19    },
20    "offset": 2,
21    "pageNumber": 1,
22    "pageSize": 2,
23    "paged": true,
24    "unpaged": false
25  },
26  "last": true,
27  "totalElements": 3,
28  "totalPages": 2,
29  "size": 2,
30  "sort": {
31    "empty": true,
32    "sorted": false,
33    "unsorted": true
```


Spring'in kendi Pageable interface'i üzerinden yapalım.

```
Book.java x BookRepository.java x BookController.java x application.properties x
6 import org.springframework.data.domain.Page;
7 import org.springframework.data.domain.PageRequest;
8 import org.springframework.data.domain.Pageable;
9 import org.springframework.web.bind.annotation.GetMapping;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RequestParam;
12 import org.springframework.web.bind.annotation.RestController;
13
14 @RestController
15 @RequiredArgsConstructor
16 @RequestMapping("/books")
17 public class BookController {
18
19     2 usages
20     private final BookRepository bookRepository;
21
22     @GetMapping
23     public Page<Book> pagination(@RequestParam Integer pageSize,
24                                @RequestParam Integer page){
25         Pageable pageable = PageRequest.of(page, pageSize);
26         return bookRepository.findAll(pageable);
27     }
28
29     @GetMapping("/pageable")
30     public Page<Book> pagination(Pageable pageable){
31         return bookRepository.findAll(pageable);
32     }
33 }
```

Attaki Insert scripti resources klasörü altında bulunmaktadır.

← → ↺ 🏠 ⓘ localhost:8080/h2-console/login.do?jsessionId=e4c0e2879b1d64e580c716fbabdba70b

🔊 💰 ☒ Auto commit 🔗 🔒 | Max rows: 1000 🏃 🏃 🛑 | 📋 | Auto complete Off 📄 Auto select On ?

📁 jdbc:h2:mem:testDb

📁 BOOKS

📄 ID

📄 AUTHOR

📄 ISBN

📄 PUBLISH_DATE

📄 TITLE

📄 Indexes

📁 INFORMATION_SCHEMA

👤 Users

📄 H2 1.4.197 (2018-03-18)

Run Run Selected Auto complete Clear SQL statement:

INSERT INTO BOOKS (ID, AUTHOR, ISBN, PUBLISH_DATE, TITLE) VALUES (1, 'TEST AUTHOR', 'TESTISBN', '2001-01-01', 'TEST KITAP 1');
INSERT INTO BOOKS (ID, AUTHOR, ISBN, PUBLISH_DATE, TITLE) VALUES (2, 'TEST AUTHOR1', 'TESTISBN2', '2001-01-01', 'TEST KITAP 2');
INSERT INTO BOOKS (ID, AUTHOR, ISBN, PUBLISH_DATE, TITLE) VALUES (3, 'TEST AUTHOR2', 'TESTISBN3', '2001-01-01', 'TEST KITAP 3');

INSERT INTO BOOKS (ID, AUTHOR, ISBN, PUBLISH_DATE, TITLE) VALUES (1, 'TEST AUTHOR', 'TESTISBN', '2001-01-01', 'TEST KITAP 1');
Update count: 1
(3 ms)

INSERT INTO BOOKS (ID, AUTHOR, ISBN, PUBLISH_DATE, TITLE) VALUES (2, 'TEST AUTHOR1', 'TESTISBN2', '2001-01-01', 'TEST KITAP 2');
Update count: 1
(0 ms)

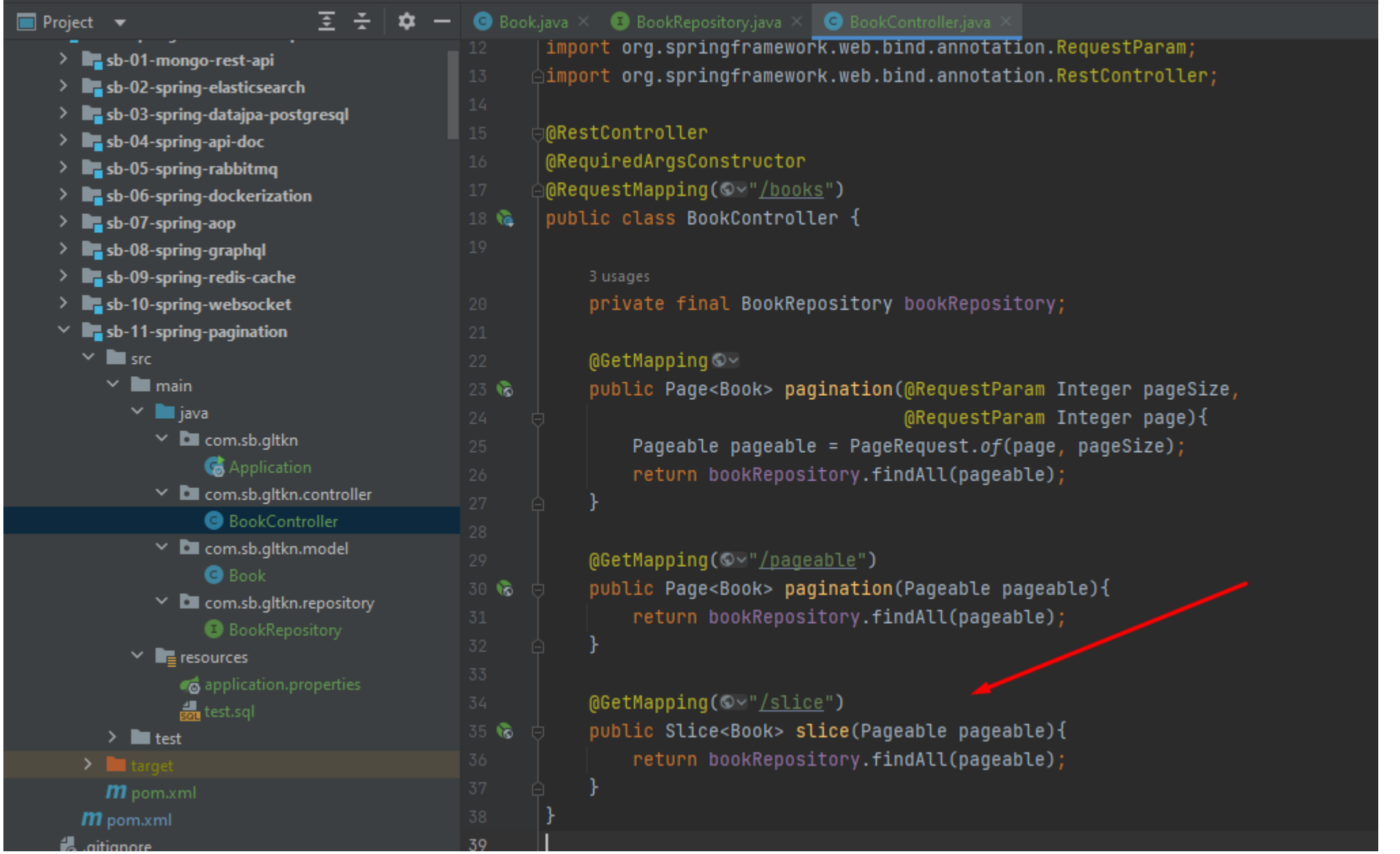
INSERT INTO BOOKS (ID, AUTHOR, ISBN, PUBLISH_DATE, TITLE) VALUES (3, 'TEST AUTHOR2', 'TESTISBN3', '2001-01-01', 'TEST KITAP 3');
Update count: 1
(1 ms)

Şimdi ise Slice ile pagination yapalım.

Slice, yukarıdaki örnekte gösterilen Page'e göre daha performanslıdır. Çünkü Page ile totalElements bilgisi için count sorgusu da atılır. Ancak Slice bu count sorgusunu atmaz.

Biz size'ı örneğin 3 gönderirsek Slice onu 4 ile sorguluyor ve bir sonraki datayı da çekiyor.

Eğer data var ise devamının olduğunu, yok ise data olmadığı bilgisine sahip oluyor. Bu da Slice'ı daha performanslı yapıyor.



```
12 import org.springframework.web.bind.annotation.RequestParam;
13 import org.springframework.web.bind.annotation.RestController;
14
15 @RestController
16 @RequiredArgsConstructor
17 @RequestMapping("/books")
18 public class BookController {
19
20     3 usages
21     private final BookRepository bookRepository;
22
23     @GetMapping
24     public Page<Book> pagination(@RequestParam Integer pageSize,
25                                @RequestParam Integer page){
26         Pageable pageable = PageRequest.of(page, pageSize);
27         return bookRepository.findAll(pageable);
28     }
29
30     @GetMapping("/pageable")
31     public Page<Book> pagination(Pageable pageable){
32         return bookRepository.findAll(pageable);
33     }
34
35     @GetMapping("/slice")
36     public Slice<Book> slice(Pageable pageable){
37         return bookRepository.findAll(pageable);
38     }
39 }
```

<http://localhost:8080/books/slice?size=1&page=0>

```
⬅ ➡ ↺ 🏠 ⓘ localhost:8080/books/slice?size=1&page=0
4  ▾ {
5  ▾   "content": [
6  ▾     {
7       "id": 1,
8       "isbn": "TESTISBN",
9       "title": "TEST KITAP 1",
10      "author": "TEST AUTHOR",
11      "publishDate": "2001-01-01"
12     }
13   ],
14 ▾   "pageable": {
15 ▾     "sort": {
16         "empty": true,
17         "sorted": false,
18         "unsorted": true
19     },
20     "offset": 0,
21     "pageSize": 1,
22     "pageNumber": 0,
23     "paged": true,
24     "unpaged": false
25   },
26   "last": false,
27   "totalPages": 3,
28   "totalElements": 3,
29   "number": 0,
30   "size": 1,
31 ▾   "sort": {
32       "empty": true,
33       "sorted": false,
34       "unsorted": true
35     },
36   "first": true,
```