Бесплатная электронная книга

# учусь spring-data

Free unaffiliated eBook created from
**Stack Overflow contributors.**

#spring-data

# Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: spring-data

It is an unofficial and free spring-data ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official spring-data.



Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

# глава 1: Начало работы с пружинными данными

## замечания

Современные программные приложения могут хранить данные в хранилищах более одного типа. В то время как традиционные хранилища данных, такие как реляционные базы данных, остаются популярными, базы данных NoSQL и облачное хранилище также стали обычным явлением. Каждый из этих типов хранилищ данных имеет свои сильные стороны и поэтому подходит для различных типов бизнес-приложений. Поэтому сложные бизнес-приложения используют более одного типа хранилища данных, чтобы сделать операции хранения, извлечения и представления данных более эффективными. Это представляет проблему, с которой программисты приложений должны иметь дело со сложностью понимания API, предоставляемой несколькими хранилищами данных, и соответствующим образом использовать эти API в своих бизнес-приложениях.

Spring Data - это проект, целью которого является обеспечение последовательного, простого в использовании API для программистов приложений, независимо от используемого хранилища данных. Он сочетает в себе мощь Spring Framework с концепциями из проверенных парадигм доступа к данным, таких как дизайн, основанный на доменах, для обеспечения знакомой и последовательной основы для прикладных программистов для доступа к различным типам хранилищ данных, при этом сохраняя специфику базового хранилища данных, где это необходимо.

Проект Spring Data состоит из нескольких подпроектов, которые могут использоваться в качестве библиотек для доступа к определенным типам хранилищ данных. Полный набор хранилищ данных, поддерживаемых Spring Data и его подпроектами, можно получить на главной странице проекта.

## Examples

### Установка или настройка

Spring Data - это проект, состоящий из нескольких подпроектов. Наиболее распространенными являются Spring Data JPA , Spring Data MongoDB , Spring Data Elasticsearch , Spring Data Neo4J , Spring Data Cassandra и Spring Data Redis .

Если вы не разрабатываете собственный подпроект, основанный на данных Spring, маловероятно, что вам нужно будет использовать его непосредственно в своем приложении. Подробные сведения об их установке и настройке см. В отдельных подпроектах. Если, однако, у вас есть необходимость использовать Spring Data в вашем

приложении напрямую, следующие инструкции будут полезны.

### Использование Maven

```
<dependencies>
  <dependency>
    <groupId>org.springframework.data</groupId>
    <artifactId>spring-data-commons</artifactId>
    <version>[version-number]</version>
  </dependency>
</dependencies>
```

### Использование Gradle

```
dependencies {
  compile 'org.springframework.data:spring-data-commons:[version-number]'
}
```

Замените *[номер версии]* на версию Spring Data, которую вы хотите использовать.

Прочитайте Начало работы с пружинными данными онлайн: https://riptutorial.com/ru/spring-data/topic/5440/начало-работы-с-пружинными-данными

# глава 2: Разбиение на страницы с данными весны

## Вступление

Разбиение на страницы путем передачи парметер с пользовательским запросом в весенних данных JPA

## Examples

### Разбиение на страницы путем передачи парметер с пользовательским запросом в весенних данных JPA

Я использую Spring Boot 1.4.4.RELEASE, с MySQL как абстракцию базы данных и Spring Data JPA для работы с MySQL. В самом деле, это модуль Spring Data JPA, который так упрощает настройку разбиения на страницы в приложении загрузки Spring.

Сценарий выставляет конечную точку / учеников / класс / {id}. Он вернет список студентов и другую информацию подкачки (которую мы увидим через минуту) на основе параметров страницы и размера и classroomId, которые были переданы вместе с ним.

Для начала создаем домен Студент

```
@Entity
@Table(name = "student")
public class Student {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "name")
    private String name;

    @NotNull
    @Column(name = "rollnumber", nullable = false)
    private Integer rollnumber;

    @Column(name = "date_of_birth")
    private LocalDate dateOfBirth;

    @Column(name = "address")
    private String address;

    @ManyToOne(optional = false)
    @NotNull
    private Classroom classroom;
```

```
//getter and setter

}
```

Студент связан с классом

```
@Entity
@Table(name = "classroom")
public class Classroom  {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "standard")
    private String standard;

    @Column(name = "section")
    private String section;

    @Column(name = "year")
    private String year;

//getter && setter

}
```

у нас есть RestController

```
@RestController
@RequestMapping("/api")
public class StudentResource {

private final StudentService studentService;

    public StudentResource(StudentService studentService) {
        this.studentService = studentService;
    }

 @GetMapping("/students/classroom/{id}")
    public ResponseEntity<Page<StudentDTO>> getAllStudentsBasedOnClassroom(@ApiParam Pageable
pageable,@PathVariable Long id)
        throws URISyntaxException {
        Page<StudentDTO> page = studentService.findByClassroomId(id, pageable);
        HttpHeaders headers = PaginationUtil.generatePaginationHttpHeaders(page,
"/api/students/classroom");
        return new ResponseEntity<Page<StudentDTO>>(page, headers, HttpStatus.OK);
    }

}
```

Обратите внимание, что мы не передали RequestParams нашему методу обработчика. Когда удаляется конечная точка / ученики / класс / 1? Page = 0 и размер = 3, Spring автоматически разрешит параметры страницы и размера и создаст экземпляр, пригодный для просмотра. Затем мы передадим этот экземпляр страницы на уровень сервиса, который передаст его на наш уровень репозитория.

Класс обслуживания

```
public interface StudentService {

  Page<StudentDTO> findByClassroomId(Long id,Pageable pageable);

}
```

service impl (здесь я пользователь StudentMapper для преобразования класса в DTOby с помощью mapStruct или мы можем делать вручную)

```
@Service
@Transactional
public class StudentServiceImpl implements StudentService{

 private final StudentRepository studentRepository;

    private final StudentMapper studentMapper;

    public StudentServiceImpl(StudentRepository studentRepository, StudentMapper
studentMapper) {
        this.studentRepository = studentRepository;
        this.studentMapper = studentMapper;
    }
@Override
    public Page<StudentDTO> findByClassroomId(Long id, Pageable pageable) {
         log.debug("Request to get Students based on classroom : {}", id);
         Page<Student> result = studentRepository.findByClassroomId(id, pageable);
         return result.map(student -> studentMapper.studentToStudentDTO(student));
    }

}
```

это интерфейс карты

```
@Mapper(componentModel = "spring", uses = {})
public interface StudentMapper{

  StudentDTO   studentToStudentDTO(Student student);
}
```

затем в специальном методе StudentRepository i worte

```
public interface StudentRepository extends JpaRepository<Student,Long> {

    Page<Student> findByClassroomId(Long id, Pageable pageable);

}
```

то он даст нам все ниже информации с соответствующими данными

```
"last": false,
  "totalElements": 20,
  "totalPages": 7,
```

```
  "size": 3,
  "number": 0,
  "sort": null,
  "first": true,
  "numberOfElements": 3
```

Прочитайте Разбиение на страницы с данными весны онлайн: https://riptutorial.com/ru/spring-data/topic/9142/разбиение-на-страницы-с-данными-весны

# кредиты

| S. No | Главы | Contributors |
|---|---|---|
| 1 | Начало работы с пружинными данными | Community, manish, sunkuet02 |
| 2 | Разбиение на страницы с данными весны | Aman Tuladhar, VISHWANATH N P |