

Аудит в JPA

В работе с базами данных часто нужно выяснить одни и те же вопросы:

- Когда была создана запись?
- Когда запись последний раз обновлялась?
- Кто создал запись?

Это настолько частые задачи, что их решение встроено в большинство фреймворков. В Spring Boot этот механизм называется **Auditing**. С его помощью мы можем автоматически записать в таблицы всю необходимую информацию.

Чтобы включить этот механизм, нужно добавить аннотацию `@EnableJpaAuditing` в классе `main`:

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

Затем подключаем аннотацию `@EntityListeners` к отслеживаемым сущностям:

```
import java.time.LocalDate;

import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;
// Остальные импорты

@Entity
@Table(name = "users")
@EntityListeners(AuditingEntityListener.class)
@Setter
@Getter
class User {
    // Остальные поля
```

```
@LastModifiedDate
private LocalDate updatedAt;

@CreatedDate
private LocalDate createdAt;
}
```

Создание сущности приведет к заполнению **createdAt**, а любое изменение — к обновлению **updatedAt**.

С указанием того, кто создал сущность или последний обновлял ее, все чуть сложнее. Потому что для этой задачи механизму обновления нужно иметь доступ к текущему пользователю, то придется выполнять дополнительные действия. Мы пока не проходили аутентификацию, поэтому пропустим эту часть, но если вы хотите, то можете изучить [официальную документацию](#).

Далее →