

# Конспект спринта №14

## JPA, Транзакции, etc.

Тема JPA настолько обширная, что по ней пишут целые книги.

Для того чтобы ориентироваться в этой теме и понимать, что искать (в поисковых системах), мы решили сделать шпаргалку по теме.

### JPA

В целом JPA это стандарт отвечающий за хранение (персистентность) данных. Его зоны ответственности это маппинг, операции чтения/записи в хранилище и управление сущностями в памяти.

### Маппинг

Маппинг это сопоставление понятий БД (таблицы, строки) и понятий из ООП (классы, свойства, объекты). Маппинг производится с помощью аннотаций, декларативно. Обеспечивается поддержка разных типов и сложных схем отношений сущностей (связи). Так же поддерживаются некоторые возможности ООП (наследование сущностей).

SQL	Java	JPA
таблица	класс	@Entity + @Table
столбец	свойство класса	@Column, @Id
строка	объект	сущность
ячейка	значение свойства	@Temporal, @Enumerated
первичный ключ	свойство	@Id, @GeneratedValue
связи (join'ы)	структура из классов	@JoinColumn, @MappedBy, @OneToMany, @ManyToOne

Ключевые слова: java, jpa, mapping, columns, entity, type, property, class, table, superclass, join.

### Чтение/запись

В процессы чтения и записи мы так же включаем формирование запросов с условиями и конвертацию данных в соответствии с маппингом. Поддерживается поиск по сложным критериям, выгрузка результатов в виде списка сущностей, в виде пользовательских проекций. По критериям возможно производить так же обновление (update) сущностей. Поддерживаются сложные запросы.

За это отвечают ORM-фреймворки (например, Hibernate) и EntityManager.

Ключевые слова: hibernate, jpa, query, criteria, fetch list, fetch one, lazy, eager.

## Сущности

Сущности это не просто объекты, десериализованные из БД. За ними следит менеджер сущностей. Сущность живёт в памяти по определенным правилам, это называется жизненный цикл, по этим же правилам принято сохранять сущность, удалять её и фиксировать результаты изменений в хранилище.

Ключевые слова: persist, merge, detach, flush, session, entity manager, state.

## JPA + Spring

При совмещении двух фреймворков, JPA и Spring Framework (Spring Boot), возникает ряд особенностей. Это готовые подключения через DataSource, встроенные транзакции и репозитории.

## Подключение

Подключение происходит с помощью настройки параметров в файле properties. По ним Spring сам понимает, к какой базе и по какому адресу подключить Data Source. К нему подключаются Hibernate (как имплементация), а уже к нему присоединяется EntityManager, как верхнеуровневый клиент для разработчика.

Ключевые слова: datasource, driver, spring boot, properties, additional parameters

## Транзакции

Внутри Spring реализован механизм транзакционного исполнения запросов через заданный DataSource и всё что его использует, то есть, всей транзакцией управляет Spring. Более того, Spring вводит понятие логических транзакций, то есть вложенность исполнения разных транзакций тоже контролирует Spring.

Ключевые слова: transactional, isolation level, transaction rollback, transaction exceptions

## Репозитории

Для обобщения работы с хранилищами разной природы, такими как СУБД (SQL, NoSQL), Кэши, Объектные хранилища, в Spring вводится понятие Репозиторий. Так же внутри репозитория вводится упрощенный способ запроса данных и механизм построения REST API для сущностей.

Ключевые слова: repository, query method, query projection, rest repository, search method parameters