

# Работа с базами данных

## JPA. Hibernate

Валерий Алексеевич Овчинников  
[valery.ovchinnikov@phystech.edu](mailto:valery.ovchinnikov@phystech.edu)

JPA

# Java Persistence API

1. JPA
2. JPAQL
3. ORM

# JPA: Entity

Сущность, объект этого типа будет представлен строками в БД, а класс будет описывать таблицу

@Entity  
@Table  
@Id  
@Column  
@Transient

# JPA: Entity

```
@Entity
class JavaBean {
    @Id
    private int id;
    public JavaBean() {}

    public int getId() {
        return this.id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```

# JPA: Configuration

Хранится в resources/META-INF/persistence.xml

```
<persistence-unit name="com.sirius.taxi"
    transaction-type="RESOURCE_LOCAL">
  <provider>org.hibernate.jpa.HibernatePersistenceProvider
</provider>
  <properties>
    <property name="javax.persistence.jdbc.url"
      value="jdbc:postgresql://localhost/taxi"/>
    <property name="javax.persistence.jdbc.user"
      value="test"/>
    <property name="javax.persistence.jdbc.password"
      value="pass"/>
    <property name="javax.persistence.jdbc.driver"
      value="org.postgresql.Driver"/>
  </properties>
</persistence-unit>
```

# javax.persistence: Persistence

## Инициализация JPA

```
EntityManagerFactory factory =  
    Persistence.createEntityManagerFactory(PU);
```

# javax.persistence: EntityManagerFactory

Сессия

```
EntityManager manager = factory.createEntityManager();
```



# javax.persistence: EntityManager

Транзакция

```
manager.getTransaction().begin();  
manager.persist(entity);  
manager.getTransaction().commit();
```

# Отношения: Relations

@OneToOne  
@OneToMany  
@ManyToOne  
@ManyToMany  
@JoinColumn

Как SQL, только JPQL

```
SELECT p from Person;
```

