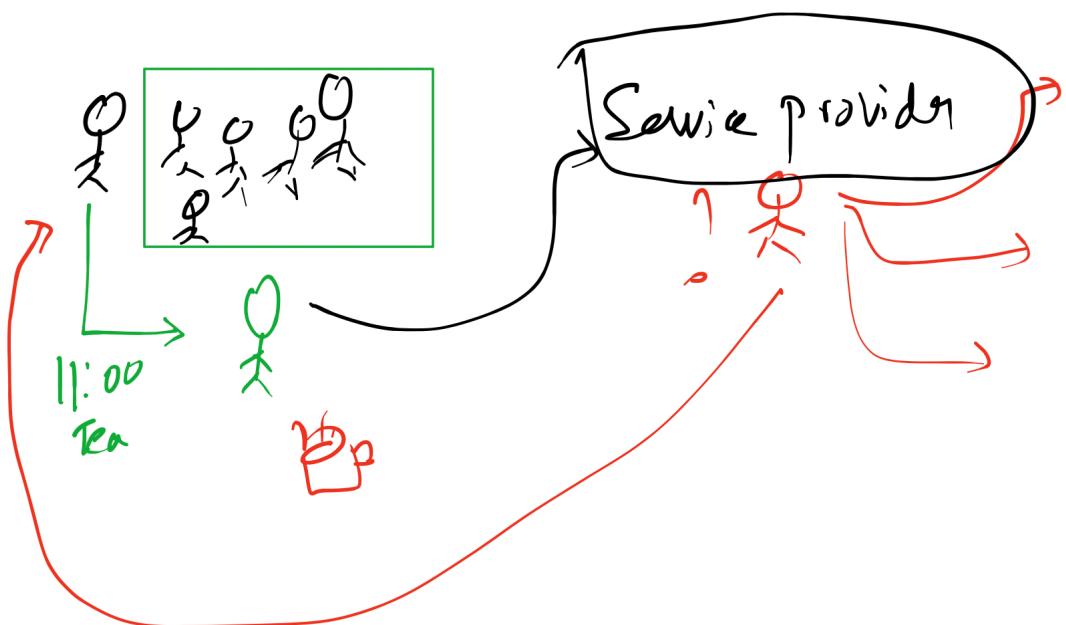
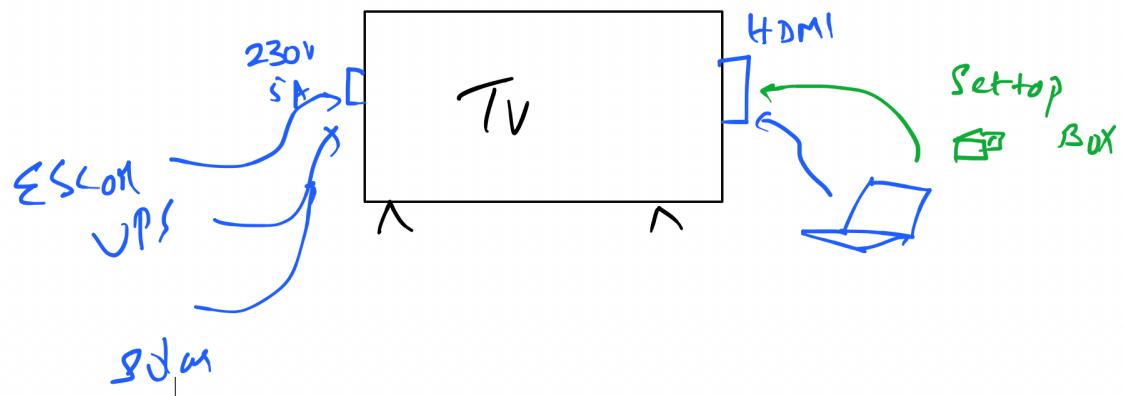
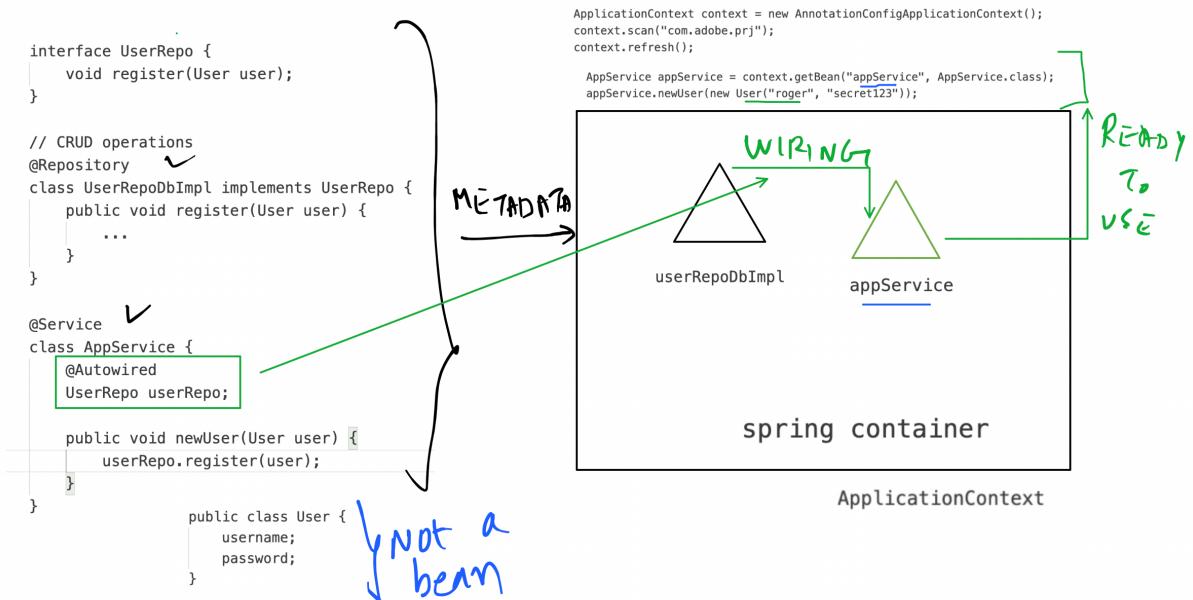


Dependency Injection





Problem:

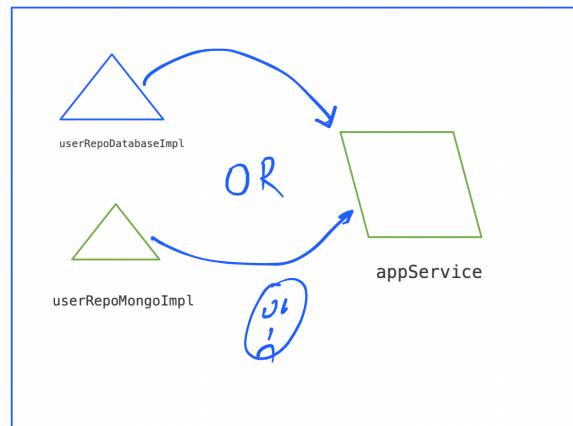
```

@Repository
public class UserRepositoryDatabaseImpl implements UserRepo{

@Repository
public class UserRepositoryMongoImpl implements UserRepo{

@Service
public class AppService {
    @Autowired
    private UserRepository userRepository;
}

```



```

Field userRepository AppService required a single bean, but 2 were found:
- userRepositoryDatabaseImpl:
- userRepositoryMongoImpl:
*****
APPLICATION FAILED TO START
*****

```

Solution 1:

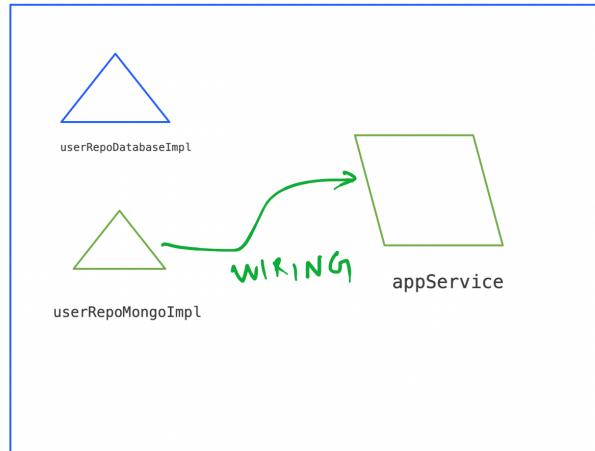
```

@Repository
public class UserRepoDatabaseImpl implements UserRepo{
    ...
}

@Primary
@Repository
public class UserRepoMongoImpl implements UserRepo{
    ...
}

@Service
public class AppService {
    @Autowired
    private UserRepo userRepo;
}

```



Solution 2:

```

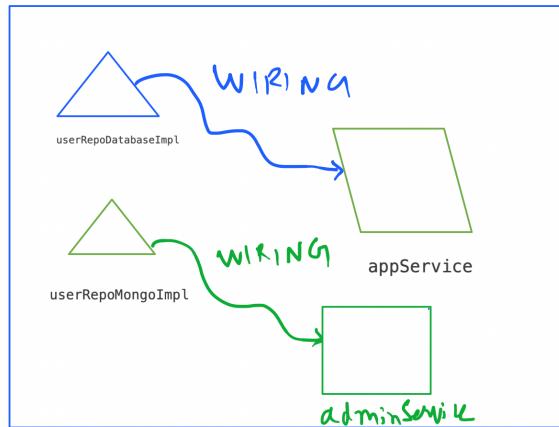
@Repository
public class UserRepoDatabaseImpl implements UserRepo{
    ...
}

@Repository
public class UserRepoMongoImpl implements UserRepo{
    ...
}

@Service
public class AdminService {
    @Qualifier("userRepoMongoImpl")
    @Autowired
    private UserRepo userRepo; // wiring is done by Spring Container
}

@Service
public class AppService {
    @Qualifier("userRepoDatabaseImpl")
    @Autowired
    private UserRepo userRepo; // wiring
}

```



Solution 3: @Profile

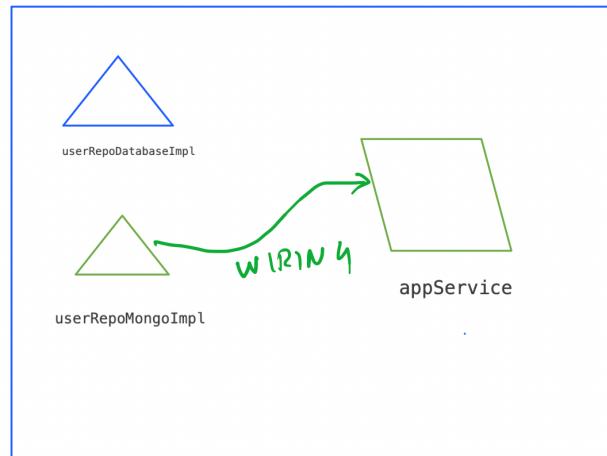
```

@Profile("dev")
@Repository
public class UserRepoDatabaseImpl implements UserRepo{

    @Profile("prod")
    @Repository
    public class UserRepoMongoImpl implements UserRepo{

        @Service
        public class AppService {
            @Autowired
            private UserRepo userRepo;
        }
    }
}

```

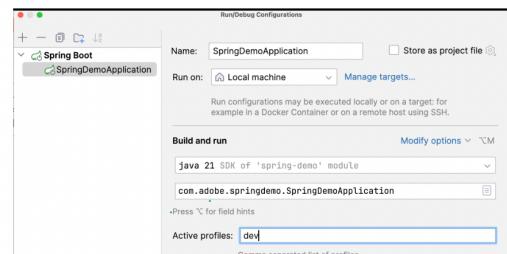


`application.properties`

```

1 spring.application.name=spring-demo
2
3
4 spring.profiles.active=prod

```



`AppConfig`

```

@Configuration
public class AppConfig {

    // factory method
    @Bean
    public DataSource getDataSource() throws Exception{
        ComboPooledDataSource cpds = new ComboPooledDataSource();
        cpds.setDriverClass( "org.h2.Driver" ); //loads the jdbc driver
        cpds.setJdbcUrl( "jdbc:h2:mem:testdb" );
        cpds.setUser("sa");
        cpds.setPassword("password");
        cpds.setMinPoolSize(5);
        cpds.setAcquireIncrement(5);
        cpds.setMaxPoolSize(20);
        return cpds;
    }
}

```

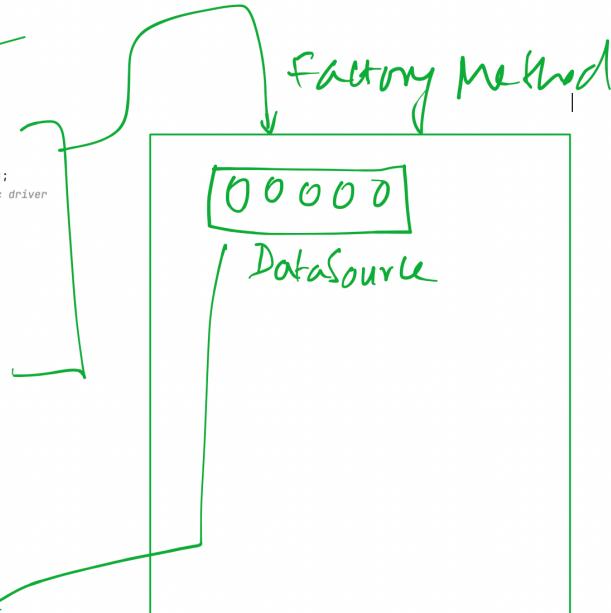
`AppService`

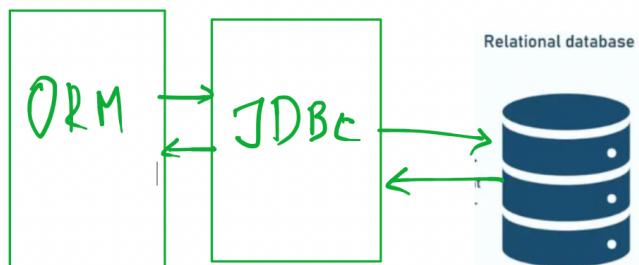
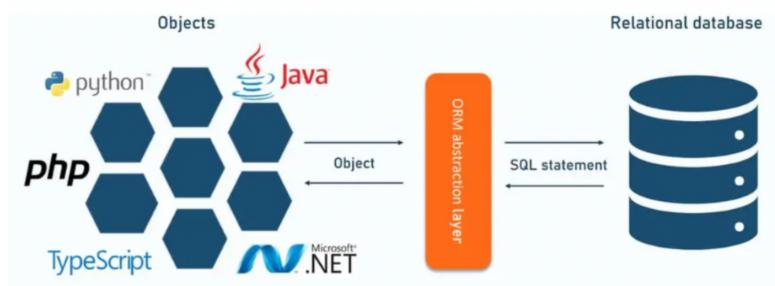
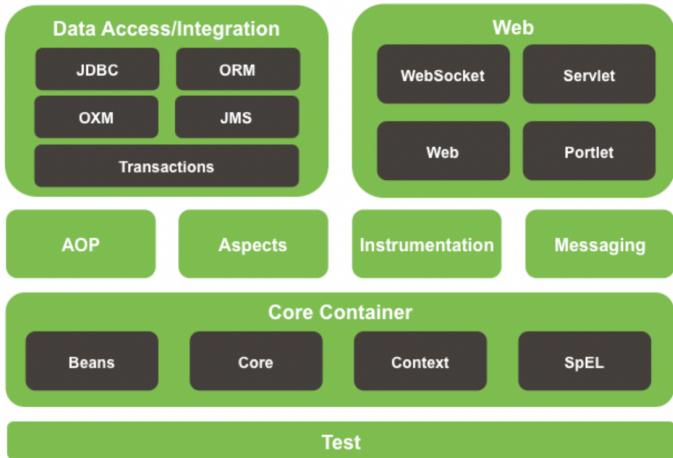
```

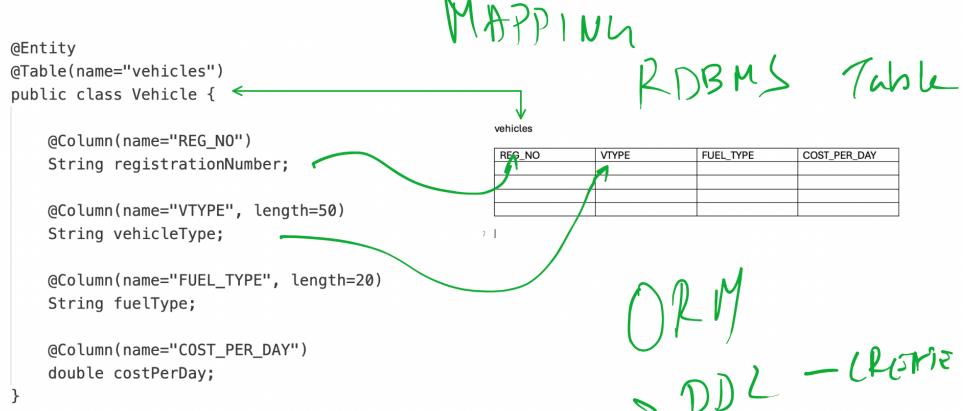
@Service
public class AppService {
    @Autowired
    private UserRepo userRepo;

    @Autowired
    DataSource dataSource;
}

```







ORM
 → DDL - CREATE, DROP, ALTER
 → DML - INSERT
 DELETE
 UPDATE
 SELECT

JDBC

```

@Override
public void addProduct(Product product) throws PersistenceException {
    Connection con = null;
    String SQL = "INSERT INTO products (id, name, price) VALUES (0, ?, ?)";
    try {
        con = DBUtil.getConnection();
        PreparedStatement ps = con.prepareStatement(SQL);
        ps.setString(1, product.getName());
        ps.setDouble(2, product.getPrice());
        ps.executeUpdate();
    } catch (SQLException e) {
        throw new PersistenceException("unable to add product!!!", e);
    } finally {
        DBUtil.closeConnection(con);
    }
}

```

ORM

```

@Override
public void addProduct(Product product) {
    em.persist(product);
}

```

