Лекция 15

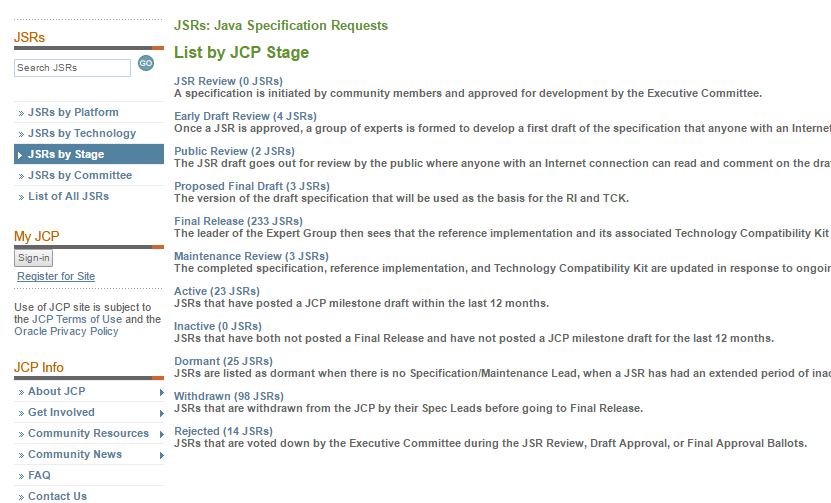
РИС, III курс, ИСиТ+ПОИТ+ПОИБМС

**Java Platform Enterprise Edition (Java EE)**

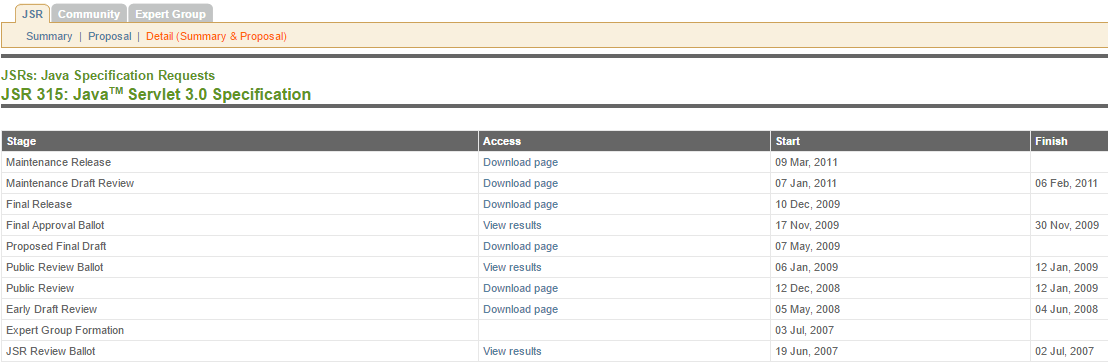
1. **Java Community Process (JCP):** организация, созданная Sun Microsystems в 1998 г.; состоит из Java-разработчиков и владельцев патентов; занимается разработкой стандартов (JSR) технологий Java; два исполнительных комитета: SE/EE и ME.

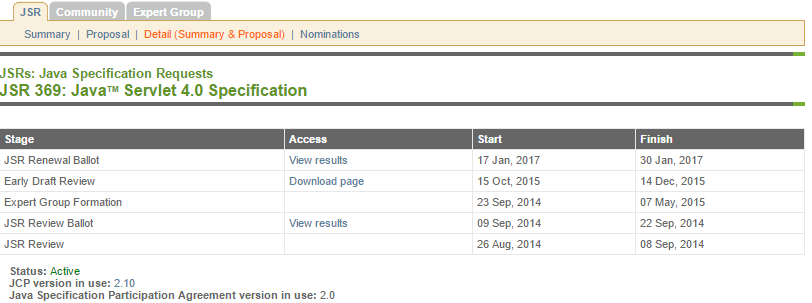
****

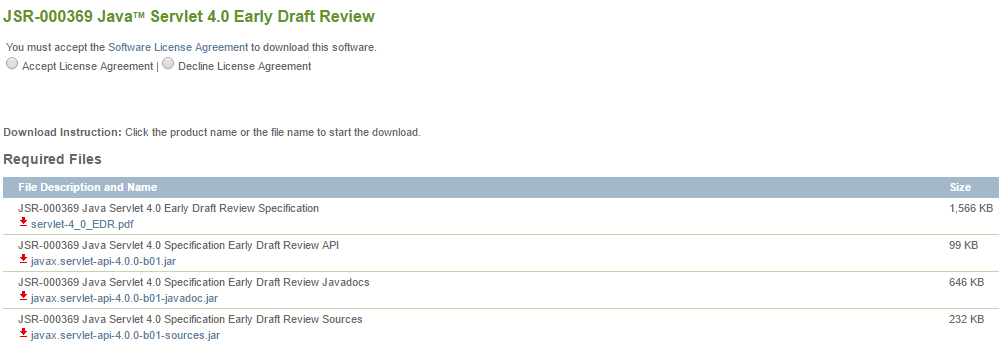
1. **Java Specification Request (JSR):** запрос на спецификацию Java; формальный документ описывающий спецификацию, которую предлагается добавить к Java-платформе;
2. **Стадии JSR:**

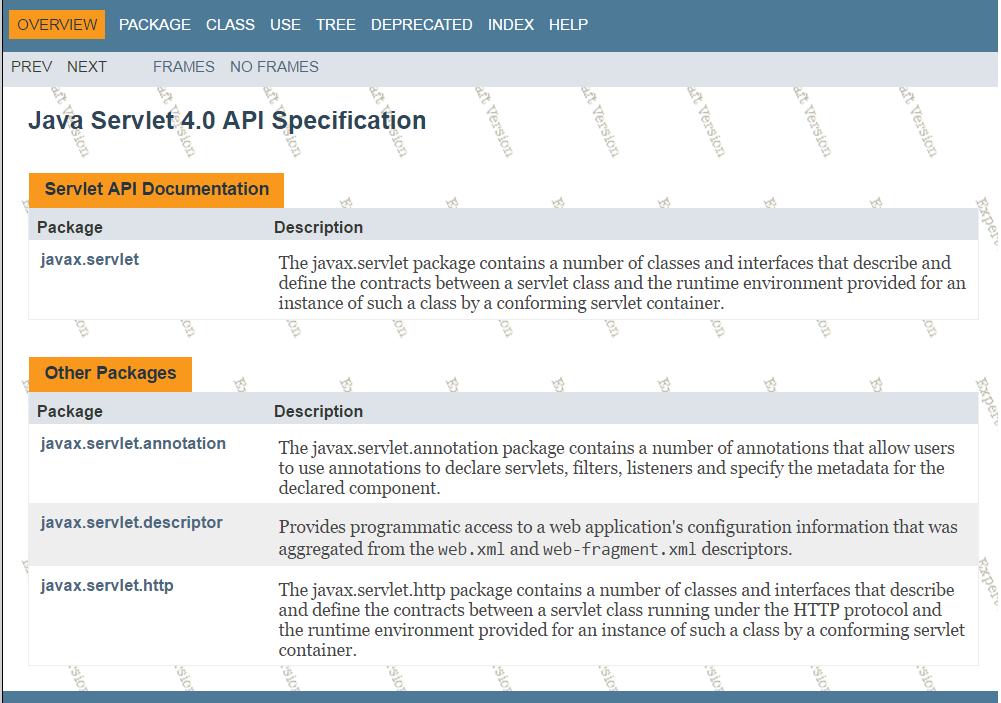


1. **Пример JSR 315, 369**

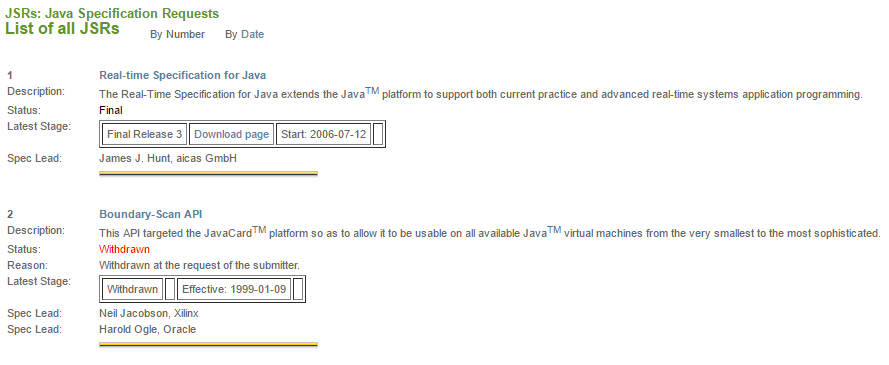
****

****

****

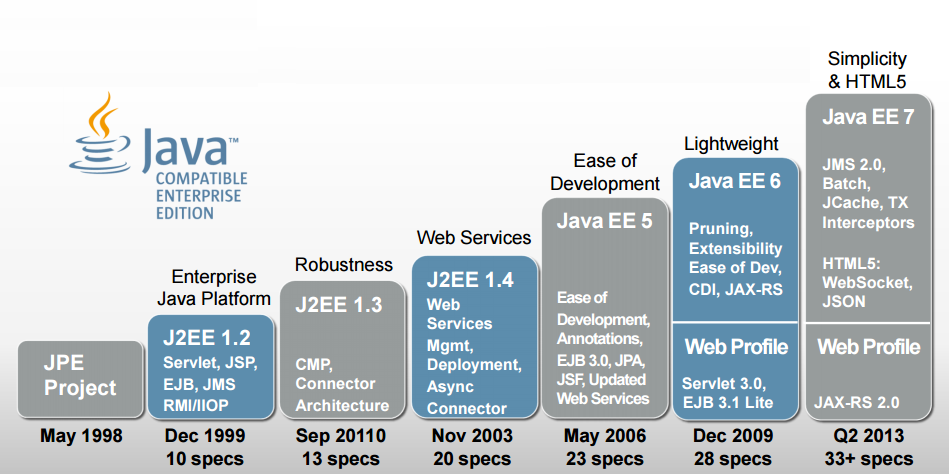
****

1. **JSR:** все на 22.03.2017



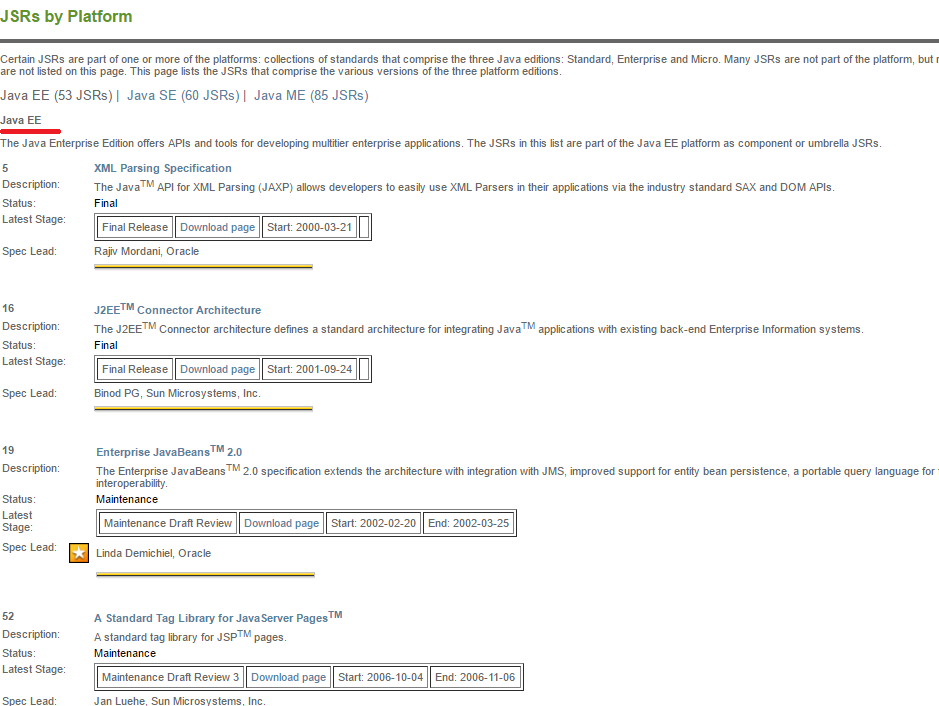
****

1. **Java EE 7**

****

****

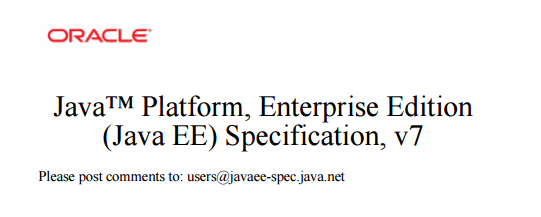
1. **JRS Java EE (53 JSRs):**

****

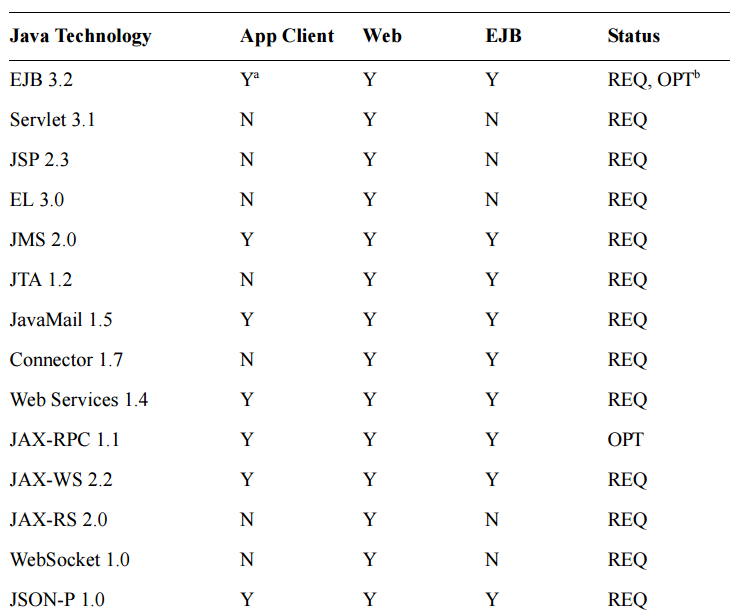
1. **Java EE версии:**

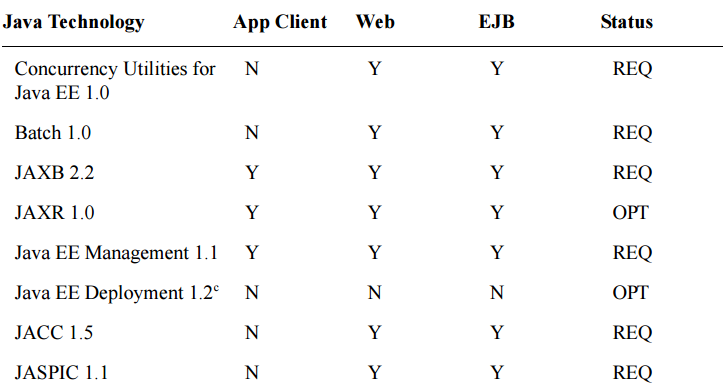
****

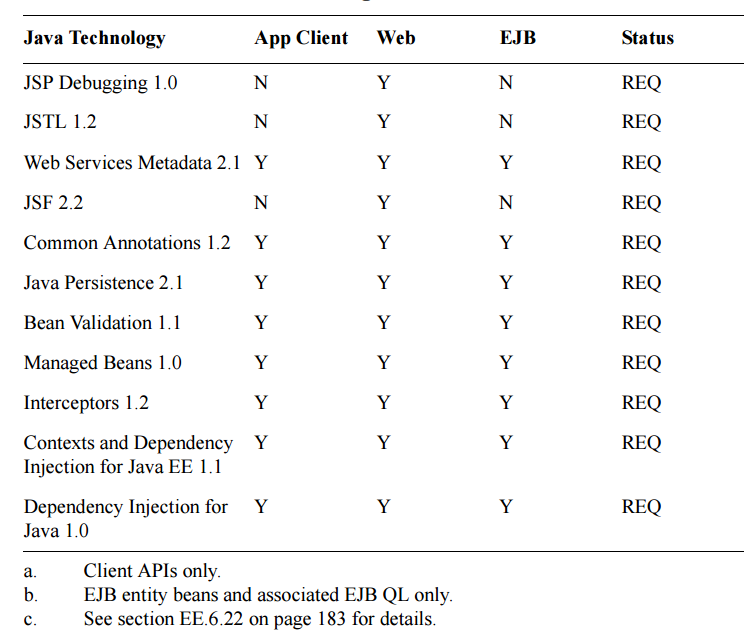
1. **Java EE спецификация:**

****

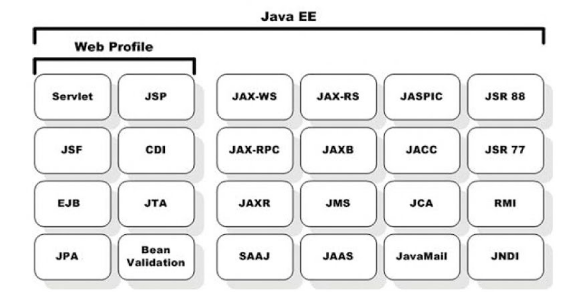
1. **Java EE состав технологий:**

****

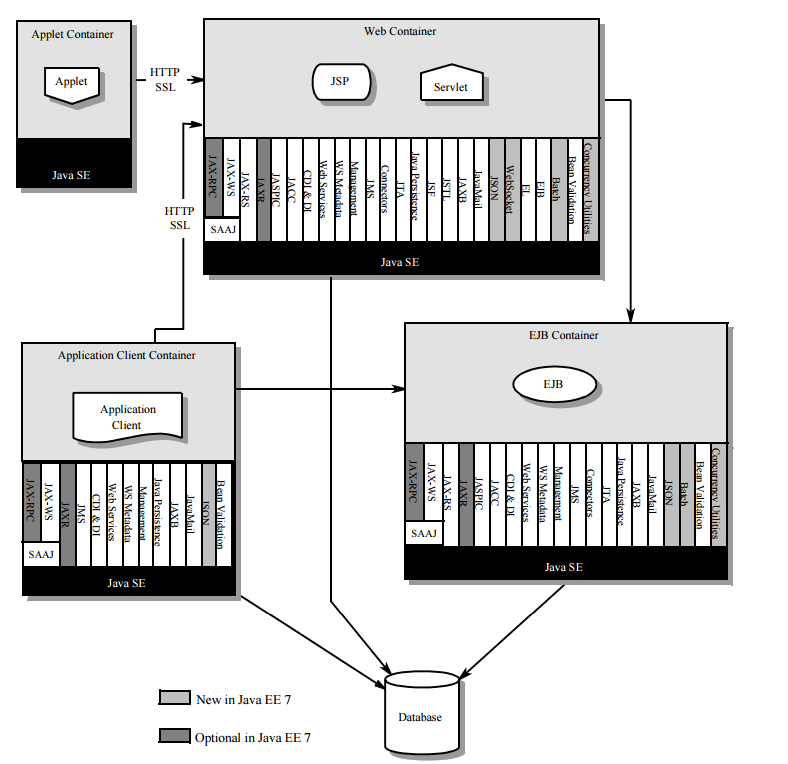
****

****

1. **Java EE профиль:** поименованное множество технологий Java EE; в Java EE 7.0 определен один профиль – web-профиль(Servlet, JSP, JSTL, JSF, WebSocket, EL).

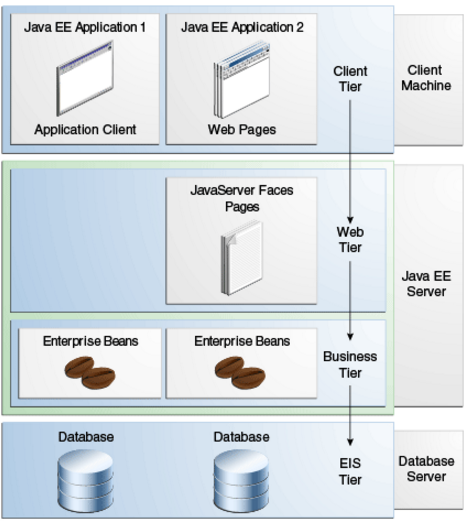
****

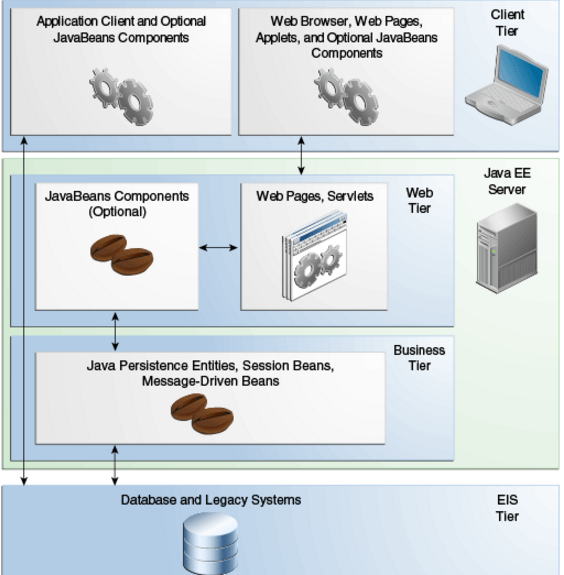
1. **Архитектура Java EE:** 4 типа контейнеров, контейнер – логическое объединение технологий с общим жизненным циклом, предназначен для сокрытия технических сложностей; каждый контейнер поддерживает свой набор интерфейсов; любое приложение разворачивается в один или несколько контейнеров.

****

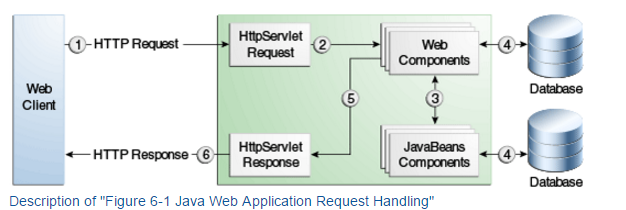
1. **Applets container:** код выгружается с сервера на клиентский компьютер, выполняется на java-машине клиента, модель «песочницы» (нет доступа к локальным компьютера ресурсам).
2. **Application client container:** Java SE – приложение, доступ к web-контейнеру по HTTP, к EJB-контейнеру по RMI-IIOP.
3. **Web Container:** службы для управления web-компонентами, все к чему можно обратиться по HTTP, (Servlet, JSP, JSF, Filters, Listeners, WS, EJB Lite).
4. **EJB Container:** службы управления компонентами EJB, все к чему можно обратиться по RMI-IIOP.
5. **Java EE SDK: Java EE Software Developer Kit** –комплект средств разработки, позволяющий разрабатывать приложения на платформе Java EE; в состав Java EE входит GlassFish Application Server.

1. **Java EE сетевые протоколы:** HTTP, HTTPS, RMI(JMRP), RMI-IIOP, IIOP(CORBA).
2. **Java EE контейнеры:** логические компоненты инфраструктуры: Applet Container, Web Container, Application Client Container, EJB Container.
3. **Java EE: четырехуровневая модель приложения**

****

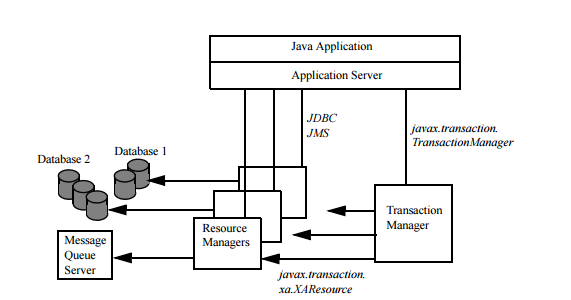
****

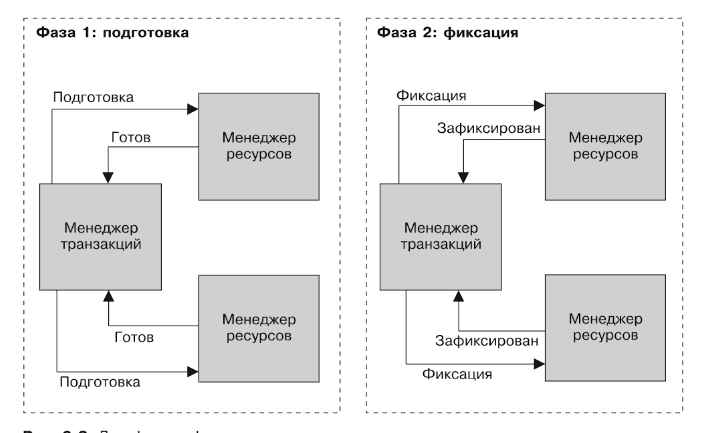
1. **Java EE: web-приложение**



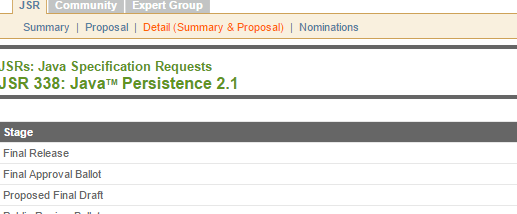
1. **Java EE: Java EE Server (Application Server)**- сервер обеспечивающий работу web и ejb –контейнеров.

1. **Java EE технологии:** Общий принцип: настройка на среду работы, применение модели провайдеров.
2. **Java EE: Базовые службы (сервисы):**
3. **Java EE CDI: Context and Dependency Injection** –внедрение зависимостей (DI – Мартин Фаулер); система аннотаций (@), с помощью которых могут описаны применяемые технологии; применяется, как правило, к POJO (Plain Old Java Object – простой Java-объект в старом стиле).
4. **Java EE JTA: Java Transaction API, JTA/XA: JTA Extended Architecture. Модели:** локальная(менеджер ресурсов), программная (возлагается на программиста), декларативная (выполняется контейнером). ACID.

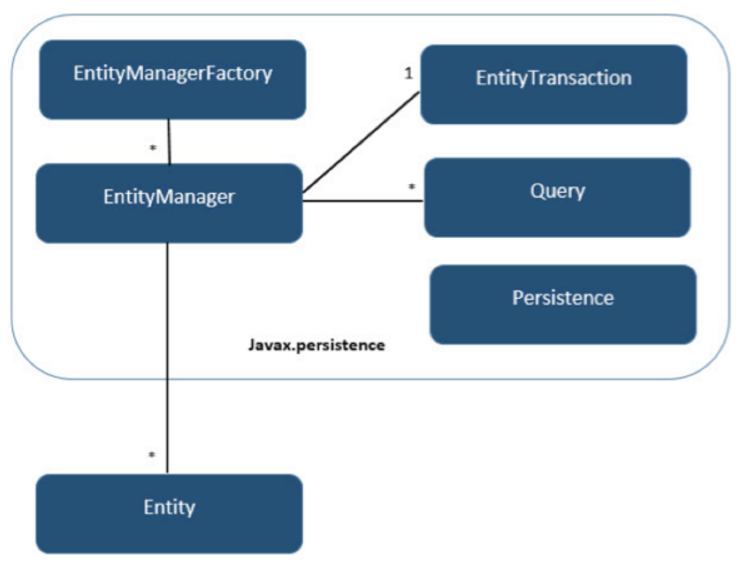
****

****

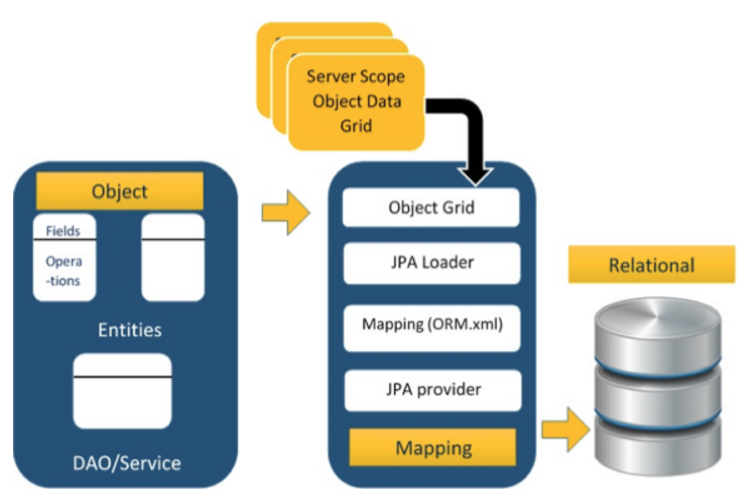
1. **Java EE JPA:** **Java Persistence API**

****

1. **Java EE JPA:** Entity, POJO (Plain Old Java Object – простой Java-объект в старом стиле).
2. **Java EE JPA:** **java.persistence,** EntityManagerFactory – фабрика классов для EntityManager; EntityManager – интерфейс для управления операциями сохранения; Entity-сущность, которая может быть сохранена как база данных, EntityTransaction – класс для поддержки транзакций, Persistence – класс со статическими методами для создания экземпляров EntityManagerFactory, Query – интерфейс реализуется вендором для доступа к реляционным объектов.

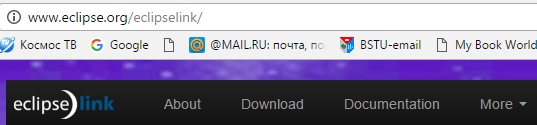


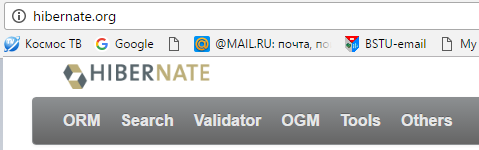
1. **Java EE JPA:** **ORM** – Object Relation Mapping, **DAO** – Data Access Object (объект для доступа к данным), **JPA Provider**: например: EclipseLink 2.5 (образцовая реализация JPA 2.1), Hibernate, TopLink; Mapping file (OXM - Object XML Mapping); **JPA Loader**: механизм кэш памяти (Object Grid) хранящей POJO-объекты, соответствующие реляционным объектам; **Object Grid:** данные в памяти

****

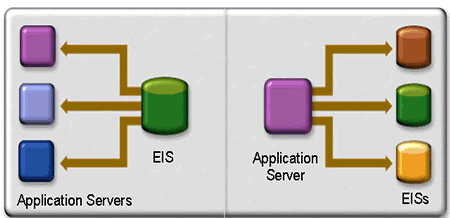
1. **Java EE JPA: JPQL –** Java Persistence Query Language, создан на основе НQL (Hibernate Query Language).

1. **Java EE JPA:** реализацииEclipseLink 2.5 (образцовая реализация JPA 2.1), Hibernate, TopLink.

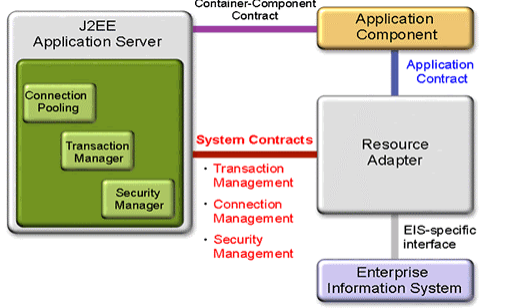




1. **Java EE JCA: Java Connector Architecture** – стандартная архитектура для доступа к ресурсам EIS (Enterprise Information System – информационная система предприятия)

****

1. **Java EE JCA:** архитектура коннектора.

****

1. **Java EE JAAS: Java Authentication and Authorization Service.**

**Реализация в Tomcat**

* 1. ***JAAS (Java Authentication and Authorization Service)*** – Java-сервис по аутентификации и авторизации.
  2. Для хранения информации о ролях пользователей, об именах пользователей и их паролях в сервере Apache Tomcat используется специальный xml-файл: ***tomcat-users.xml***
  3. Каждый пользователь характеризуется: именем, паролем и списком ролей, приписываемых ему.
  4. Защищенный ресурс (html –страница, jsp-страница, сервлет) привязывается к ролям. Все пользователи, содержащие в своем списке роль, обеспечивающую доступ (привязанную роль) к некоторому ресурсу, имеют возможность доступа к данному ресурсу.
  5. Технология JAAS предполагает две формы аутентификации пользователей: ***BASIC*** и ***FORM*** .
  6. BASIC-аутентификация подразумевает использование стандартной формы ввода данных для аутентификации (имени и пароля).
  7. FORM-аутентификация подразумевает использование собственной формы (разработанной пользователем, но по определенным правилам jsp-страницы) для аутентификации.

Применение JAAS-технологии сводится к xml-описанию необходимой информации в дескрипторе приложения web.xml и к разработке форм для ввода имени и пароля пользователя и сообщения об ошибке (для FORM-аутентификации).

**<!-- ----------------------------начало web.xml ---------------------------------------- --->**

**<security-constraint>**

**<web-resource-collection>**

**<web-resource-name>JSP-Servlet Sss</web-resource-name>**

**<url-pattern>/index.jsp</url-pattern>**

**<http-method>GET</http-method>**

**<http-method>POST</http-method>**

**</web-resource-collection>**

**<auth-constraint>**

**<role-name>Rolexxy</role-name>**

**</auth-constraint>**

**</security-constraint>**

**<login-config>**

**<auth-method>BASIC</auth-method>**

**</login-config>**

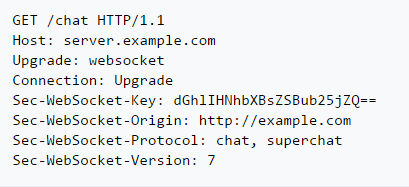
**<security-role>**

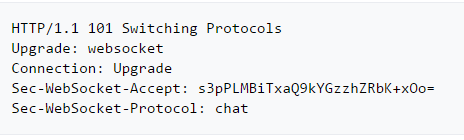
**<role-name>Rolexxy</role-name>**

**</security-role>**

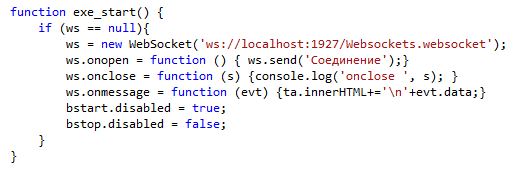
**<!-- ---------------------------продолжение web.xml ---------------------------------------- --->**

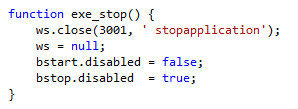
1. **Java EE WebSocket: WebSocket -** протокол дуплексной связи (RFC 6455, Спецификация W3C). Первоначально соединение по HTTP, а потом переход websocket-протокол.
2. **Java EE WebSocket: соединение**



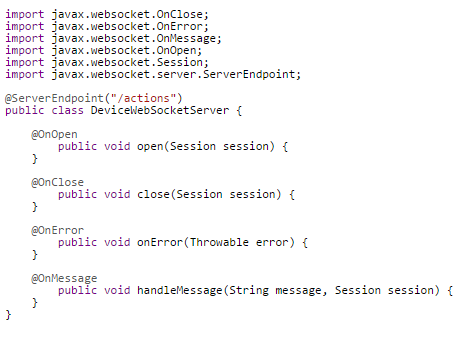
****

1. **Java EE WebSocket: клиент**



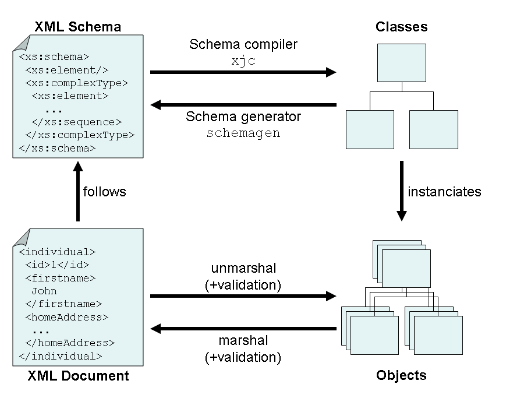


1. **Java EE WebSocket: сервер**

****

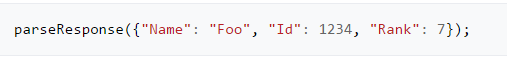
**Java EE JAX-RPC:** версия 1.1, SOAP 1.1

1. **Java EE JAX-WS:** JSR 224**,** версия 2.1, SOAP 1.1, SOAP 1.2, XML/HTTP – пересылка XML-сообщений (без SOAP), использует JAXB
2. **Java EE JAX-RS:** JSR 311, REST
3. **SAAJ:** API низкого уровня для работы с SOAP Web Service.
4. **Java EE JAXR: Java API for XML Registries.**  UDDI – частный случай. Регистрация в хранилище (базе данных) XML-информации о Web-сервисах. Полное API для провайдеров: Registry provider, JAXR-провайдер.
5. **Java EE JAXB: Java Architecture for XML Binding.** JSR 222.Специальный инструмент для пересылки (сериализация/десериализация в XML) объектов.



1. **Java EE JSON-P: JSON with padding** – способ запросить данные, которые находятся в другом домене (страница полученная с сервера, не может связаться с другим сервером – политика ограничения домена), <script> - исключение; можно запросить js-скрипт, начиненный json-данными, только GET-метод

****



1. **Java EE JSON-P:** альтернатива CORS, должен поддерживать браузер, применение заголовка (от сервера) Access-Control-Allow-Origin c указанием доверенных серверов.

1. **Java EE JASPIC: Java Authentication Service Provider Interface for Container** – контракт между сервером и разработчиком (для )**.**
2. **оо**
3. **оо**
4. **оо**
5. **оо**
6. **о**
7. **оо**
8. **оо**