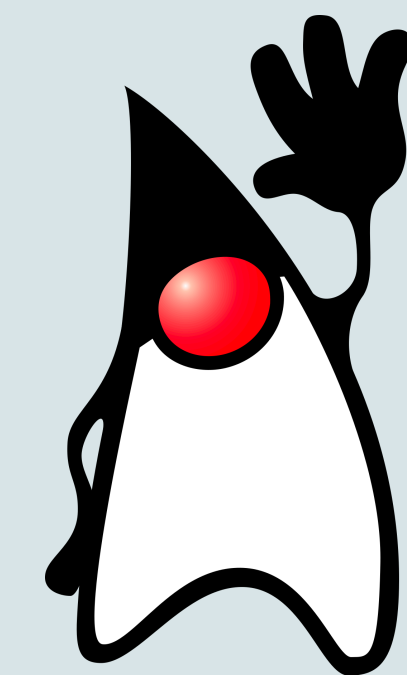




СБЕРБАНК

Корпоративный
университет



Spring MVC

Занятие №16



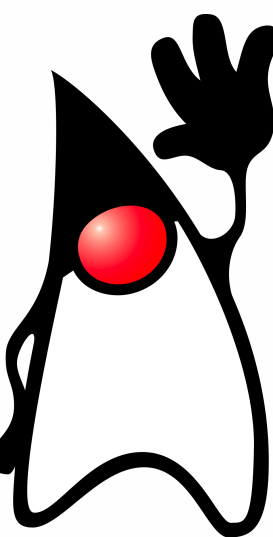
СБЕРБАНК

Корпоративный
университет

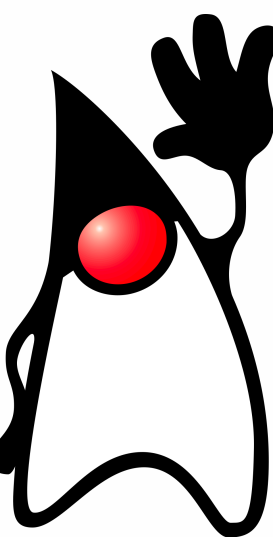


Что сможем после занятия

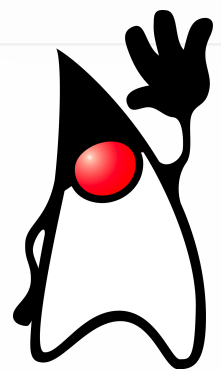
- Вспоминаем http в Java.
- WAR-file
- Spring MVC
- Безопасность приложений
- Spring Security. Начало



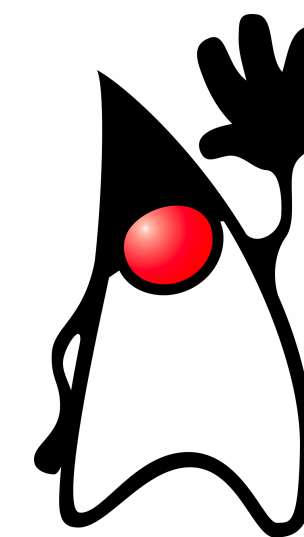
- Активно участвуем. Не стесняйтесь задавать вопрос.
- Но off-topic обсуждаем в Telegram @sb_ku_java_2019_10
- Не стесняйтесь просто спрашивать в telegram.
- ДЗ - работаем над библиотекой



**Договорились?
Поехали!**



- **Вспоминаем http в Java.**
- WAR-file
- Spring MVC
- Spring Security. Начало

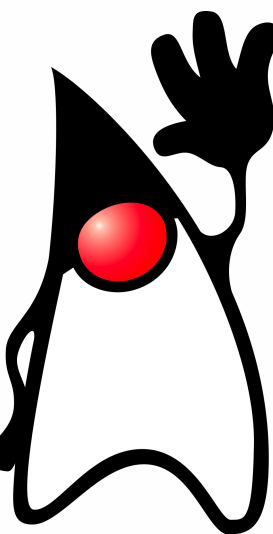


01

HTTP в Java.

Веб-сервер — сервер, принимающий **HTTP**-запросы от клиентов, обычно **веб-браузеров**, и выдающий им **HTTP**-ответы, как правило, вместе с **HTML**-страницей, изображением, **файлом**, медиа-потокom или другими данными.

Веб-сервером называют как **программное обеспечение**, выполняющее функции веб-сервера, так и непосредственно **компьютер** (см.: **Сервер (аппаратное обеспечение)**), на котором это программное обеспечение работает.

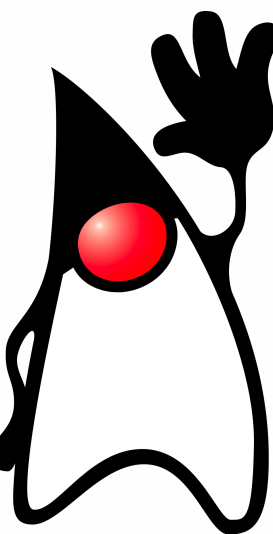


Request message

- A request line (GET logo.png HTTP/1.1)
- Request header
- An empty line
- An optional message body

Response message

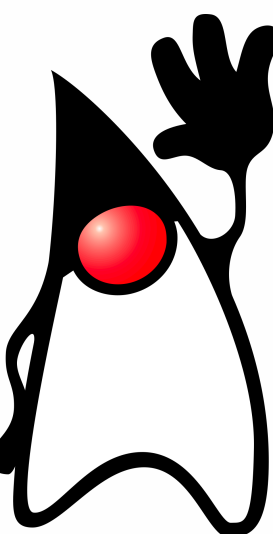
- A status line which includes the status code and reason message (HTTP/1.1 200 OK)
- Response header
- An empty line
- An optional message body



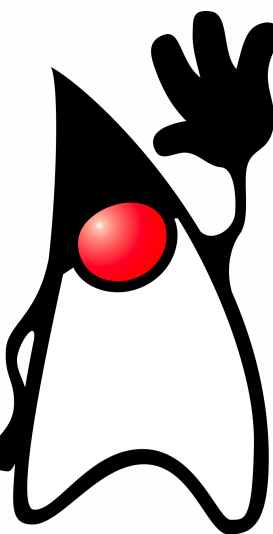
[JSR-000369 Java Servlet 4.0 Specification Final Release](#)

A servlet is a Java™ technology-based Web component, managed by a container, that generates dynamic content.

Т.е. простыми словами, это класс, который динамически создает некий контент.



1. Клиент (например Web-браузер) отправляет запрос Web-серверу
2. Web-сервер получает запрос и передает его контейнеру сервлетов.
3. Контейнер сервлетов по запросу определяет нужный сервлет и передает ему запрос.
4. Сервлет обрабатывает запрос и формирует ответ.
5. Контейнер сервлетов передает ответ серверу, а тот - клиенту



Сервлет – это абстрактный класс HttpServlet из пакета javax.servlet.http.

Основные методы.

init

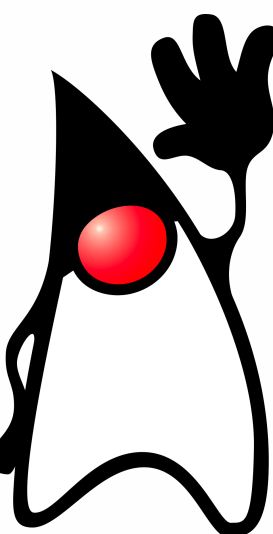
doGet

doPost

doPut

doDelete

destroy

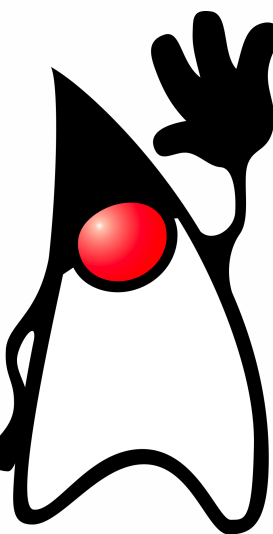


interface HttpServletRequest

Сервлет получает запрос в виде реализации этого интерфейса.

Примеры методов:

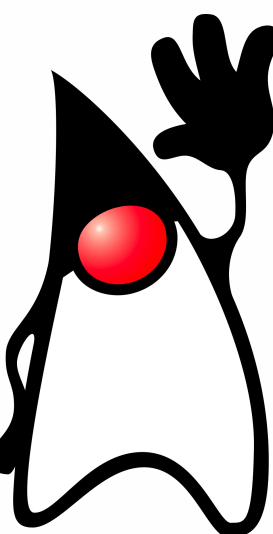
```
getCookies()  
getPathInfo();  
getUserPrincipal()  
getRequestURI();  
getSession(boolean create);  
authenticate(HttpServletRequest response)
```



ServletContext – это инфраструктурная часть, которая содержит сервлеты и прочие компоненты для обработки запросов.

ServletConext привязывается к определенному адресу и обрабатывает все запросы, которые на этот адрес приходят.

Например, ServletContext может быть привязан к адресу <http://host/dataSource>. Контейнер сервлетов все запросы, приходящие на /dataSource будет передавать в этот ServletContext.



Ваши вопросы?

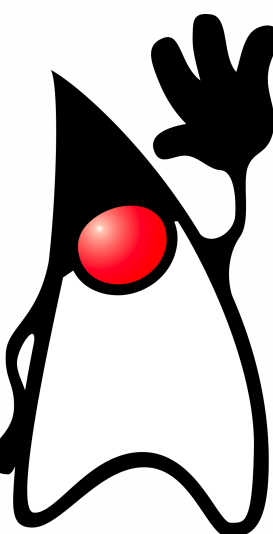
02

WAR-файл

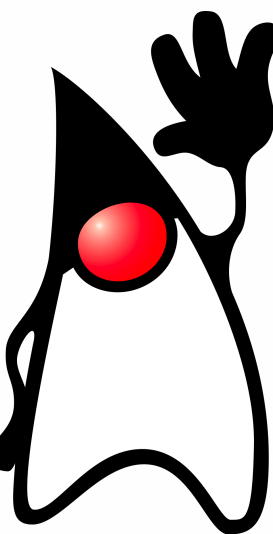
Web Archive или **Web Application Resource**^[1] — формат файла, описывающий, как полное веб-приложение упаковывается в соответствии со спецификацией **Java-сервлетов** в файл в формате **JAR** или **ZIP**.^[2] Такие файлы имеют расширение «.war» и поэтому называются ещё «WAR-файлами».

Преимущества WAR-файлов:

- лёгкость разработки, тестирования и развёртывания
- версию развёрнутого приложения легко идентифицировать
- Все J2EE-контейнеры поддерживают WAR-файлы



```
/index.html  
/guestbook.jsp  
/images/logo.png  
/WEB-INF/web.xml  
/WEB-INF/classes/org/wikipedia/Util.class  
/WEB-INF/classes/org/wikipedia/MainServlet.class  
/WEB-INF/lib/util.jar  
/META-INF/MANIFEST.MF
```

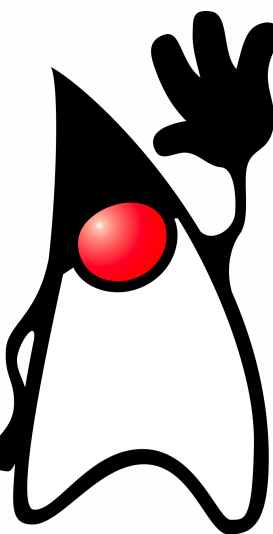


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
    "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">

<web-app>
    <servlet>
        <servlet-name>HelloServlet</servlet-name>
        <servlet-class>mypackage.HelloServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
        <url-pattern>/HelloServlet</url-pattern>
    </servlet-mapping>

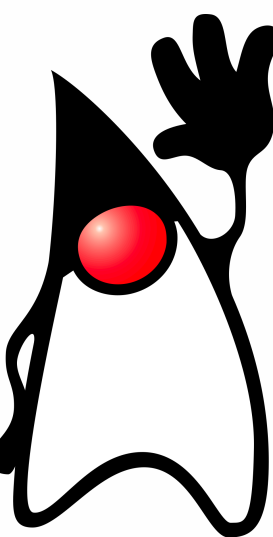
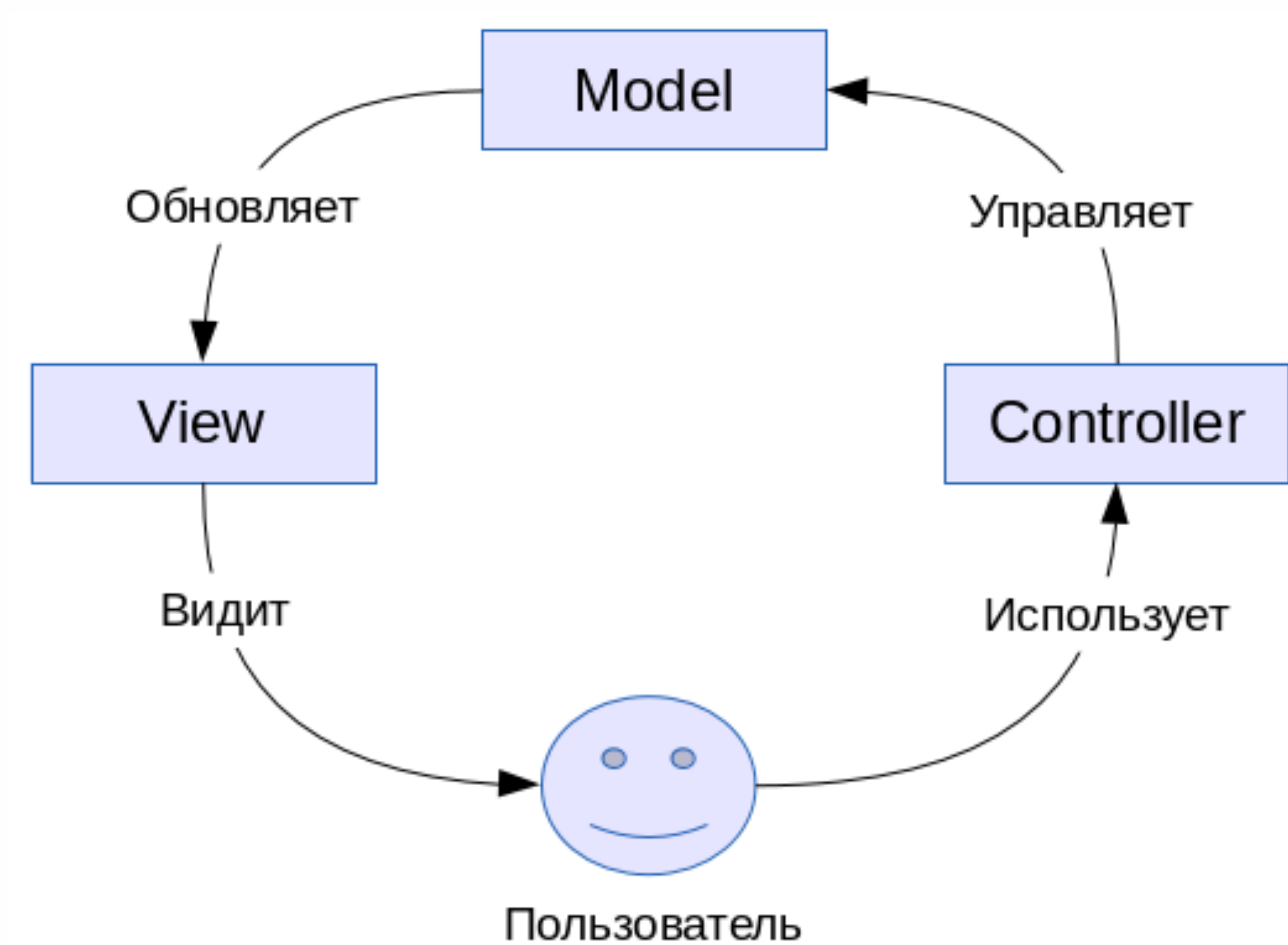
    <resource-ref>
        <description>
            Resource reference to a factory for javax.mail.Session
            instances that may be used for sending electronic mail messages,
            preconfigured to connect to the appropriate SMTP server.
        </description>
        <res-ref-name>mail/Session</res-ref-name>
        <res-type>javax.mail.Session</res-type>
        <res-auth>Container</res-auth>
    </resource-ref>
</web-app>
```

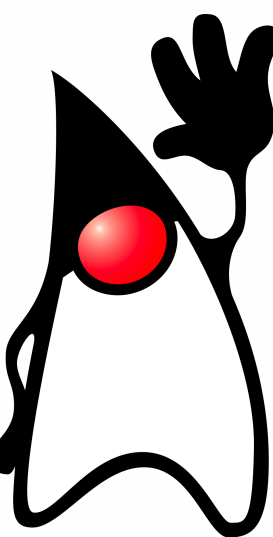
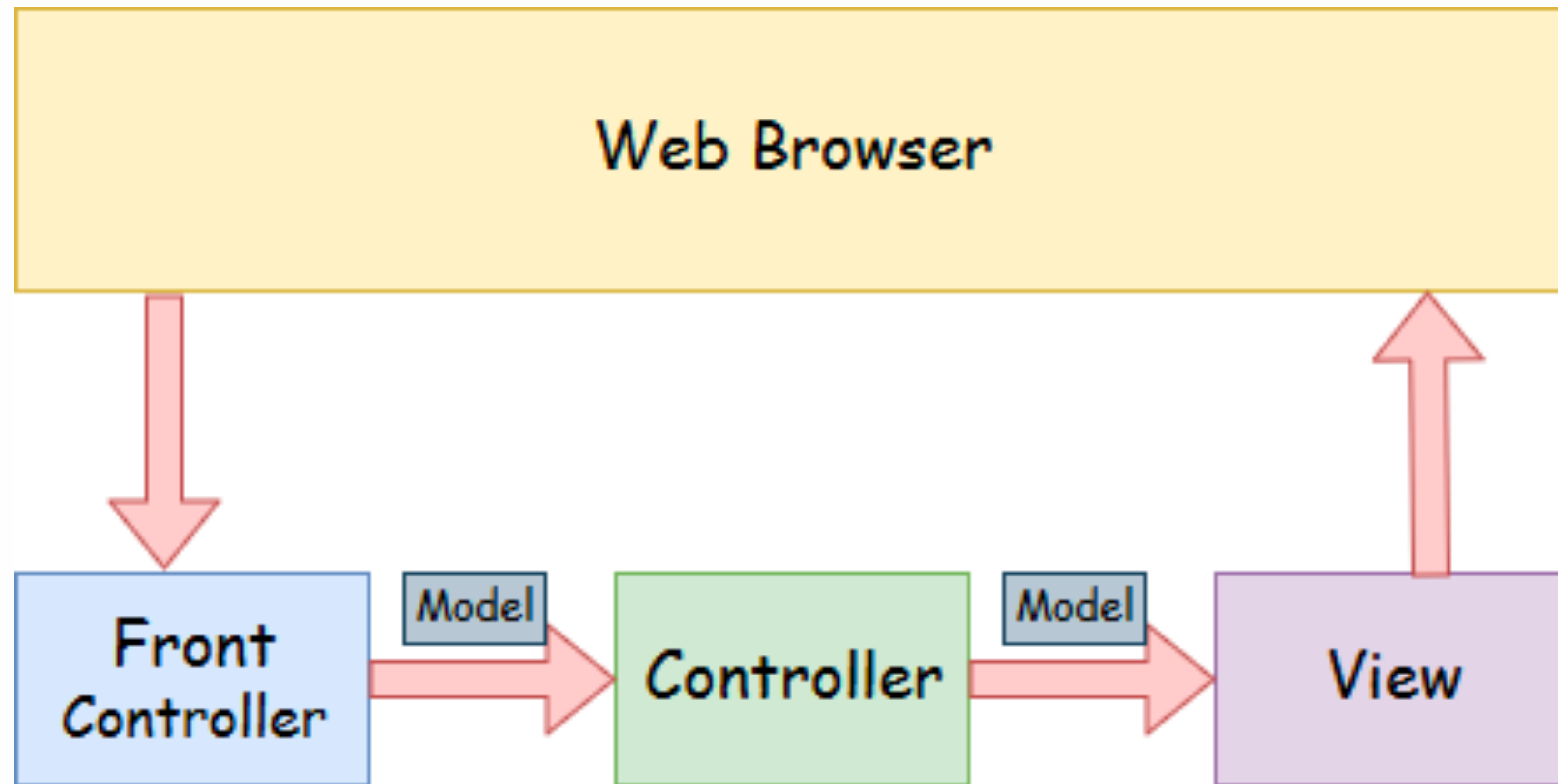


Ваши вопросы?

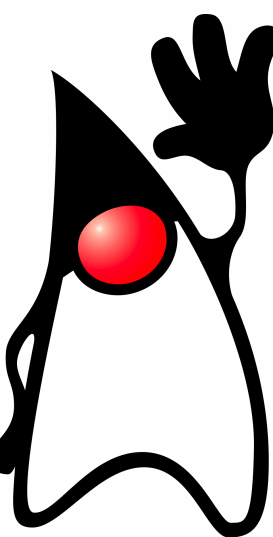
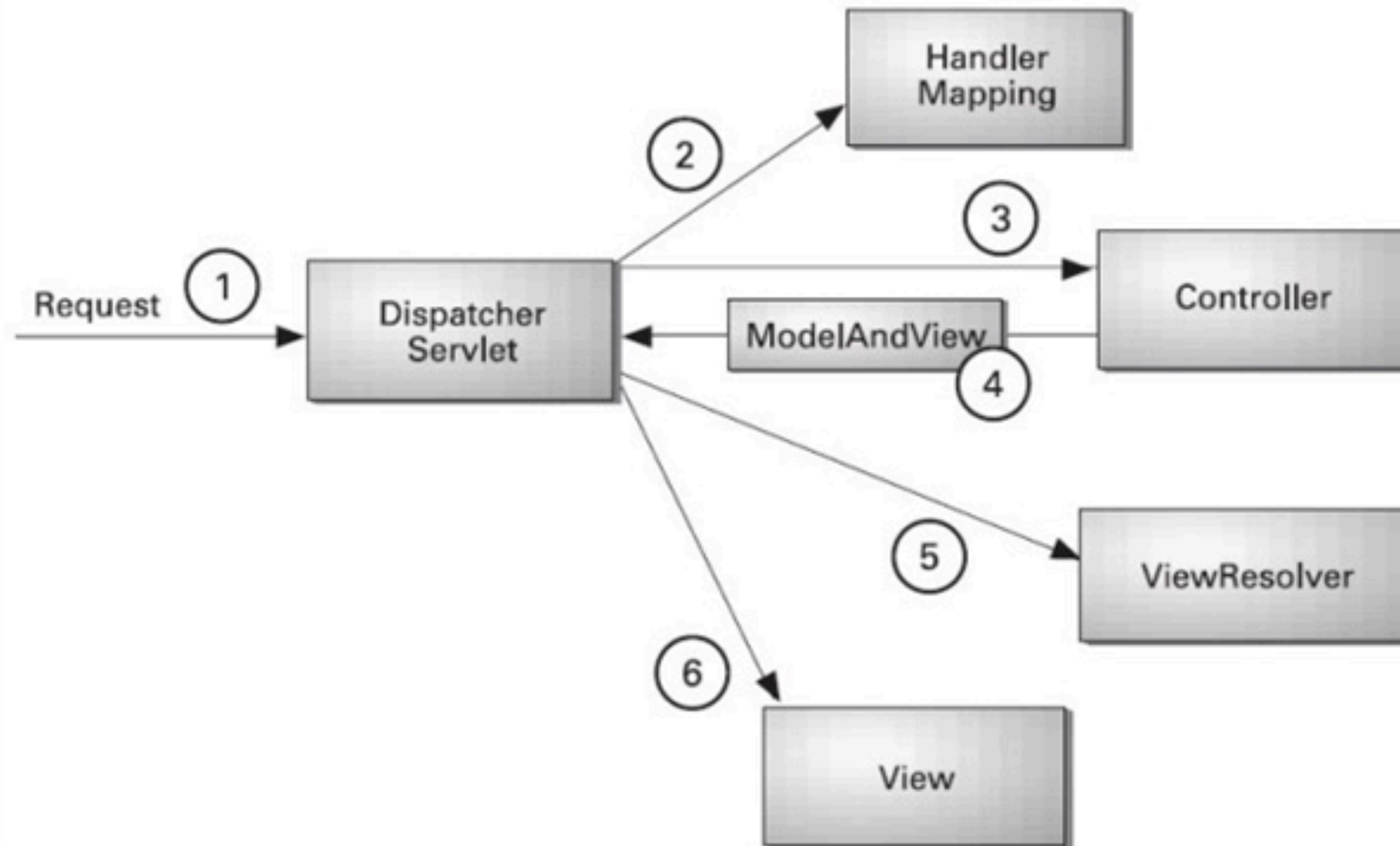
03

Spring MVC



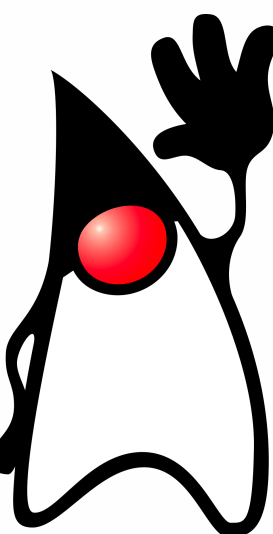


Spring MVC. Flow




```
<html>
<body>
<h1>Message : ${message}</h1>
</body>
</html>
```

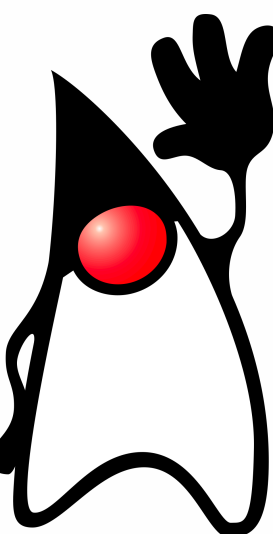
```
public String printWelcome( ModelMap modelMap ) {
    modelMap.addAttribute( "message", "hello, world" );
    return "index";
}
```



https://ru.wikipedia.org/wiki/JavaServer_Pages

Java Server Pages - технология, позволяющая веб-разработчикам создавать содержимое, которое имеет как статические, так и динамические компоненты.

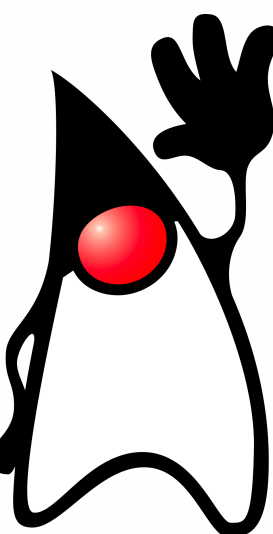
Код JSP-страницы транслируется в Java-код сервлета с помощью компилятора JSP-страниц **Jasper**, и затем компилируется в **байт-код** виртуальной машины **Java (JVM)**.



```
Имя вашего хоста: <%= request.getRemoteHost( ) %>
```

```
<%--  
    Отобразит каталог изделий  
    и актуальную корзину покупателя.  
--%>
```

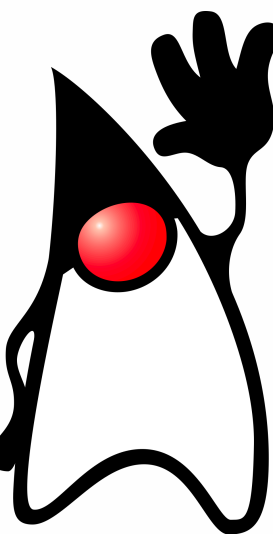
```
<% if (Math.random( ) < 0.5) { %>  
    <B>Удачного</B> вам дня!  
<% } else { %>  
    <B>Неудачного</B> вам дня!  
<% } %>
```



```
<%@ taglib uri="URI к библиотеке тегов" prefix="префикс знака" %>
```

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

```
<c:forEach items="${books}" var="book">
```



Приложение библиотеки

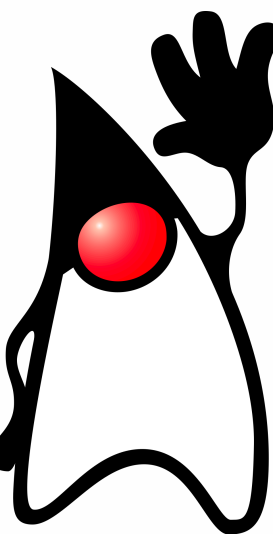
0. Настройка tomcat на локале

1. Делаем приветственное сообщение, с текущей датой.

2. Список книг (ой)

3. Создание заполненной базы

4. Редактирование книги

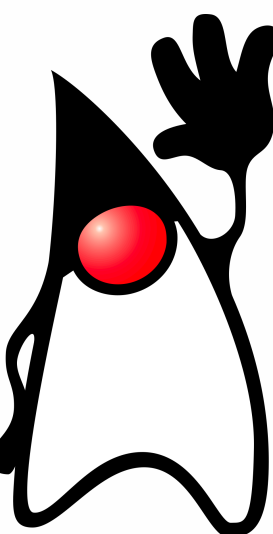


Ваши вопросы?

<https://www.javatpoint.com/spring-mvc-tutorial>

Сделать форму редактирования

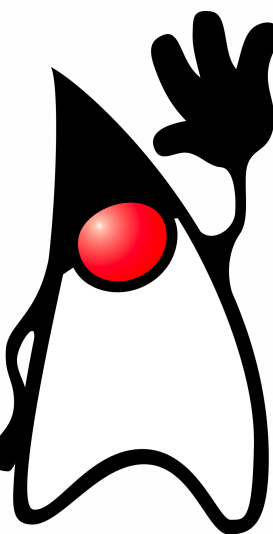
<https://www.javatpoint.com/spring-mvc-form-tag-library>



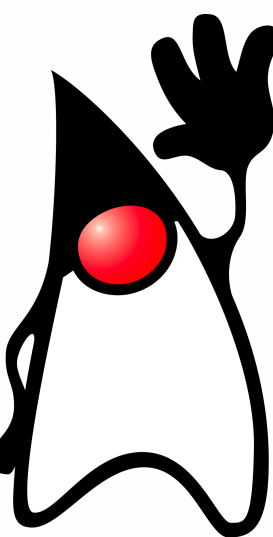
04

Безопасность приложений

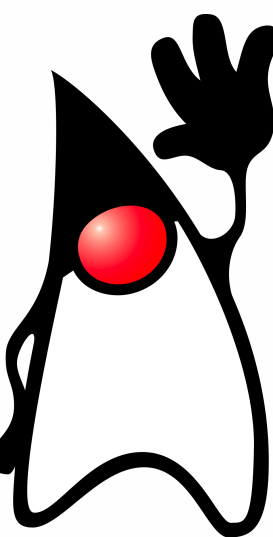
Какие задачи безопасности возникают в Enterprise приложении?



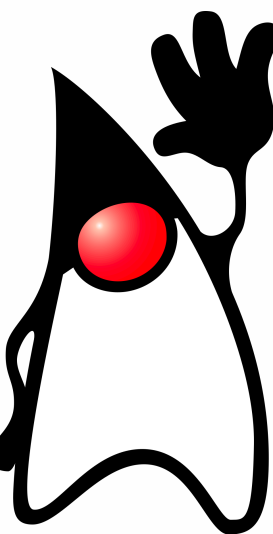
- Пользователь тот, за кого он себя выдает
- Пользователю предоставляет доступ только к тому функционалу, к которому он имеет доступ



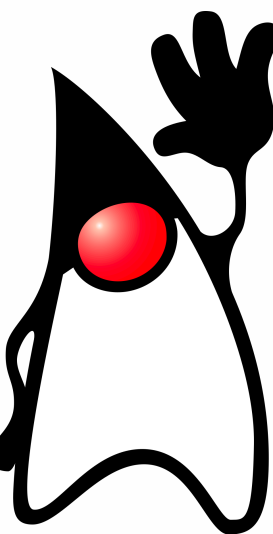
- Какие механизмы необходимы для решения этих задач?



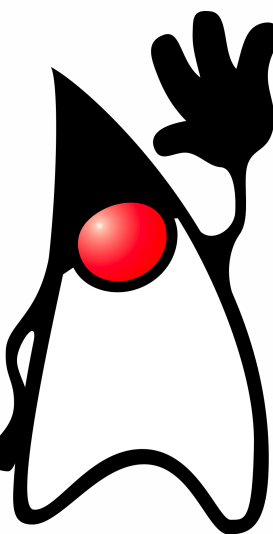
- Механизм подтверждения подлинности пользователя
- Механизм предоставления/запрещения доступа пользователям
- Механизм хранения прав доступа
- Механизм проверки прав доступа



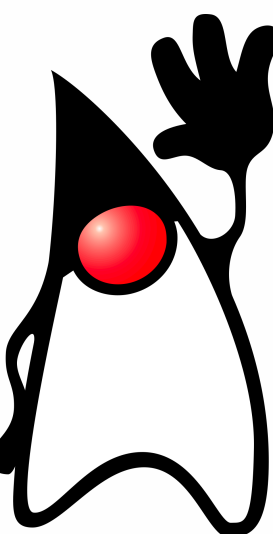
- Реализация механизмов самостоятельное - сложная задача
- Механизмы повторяются от приложения к приложению
- Если микросервисы - нужно в каждый добавлять это решение
- А есть Spring Security



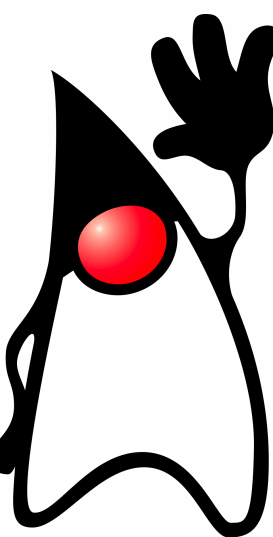
- А вот Spring Security и содержит реализацию механизмов безопасности



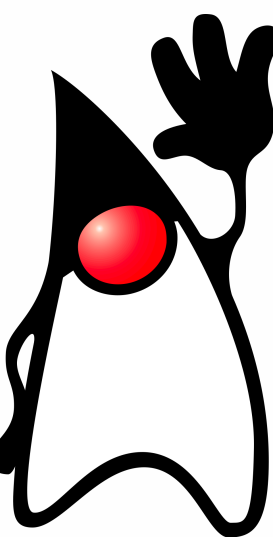
- Аутентификация - проверка что пользователь тот, за кого себя выдает
- Авторизация - проверка/предоставление прав доступа к объекту (объекты бывают разные)



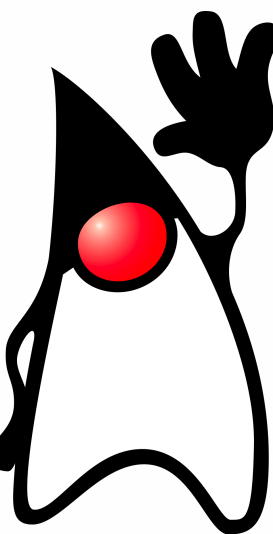
- HTTP Basic authentication
- HTTP Digest authentication
- HTTP X.509
- Form-based authentication



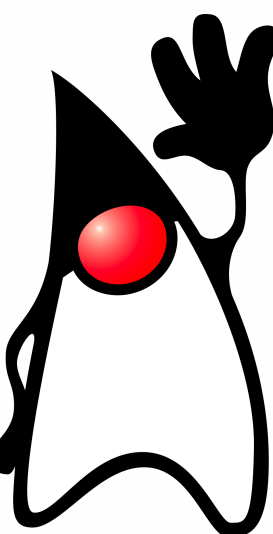
- OpenID
- LDAP
- JDBC/In-Memory хранение пользователей
- Можно сделать свою реализацию интерфейсов



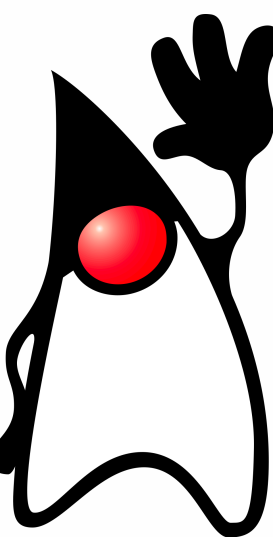
- По URL-ам
- Методы в сервисах
- По объектам



- Не является Firewall
- Не является антивирусом
- Не спасает приложение от различных атак - SQL Injection, XSS и т. п.
- Хотя есть защита от некоторых
- Не шифрует/обфусцирует поток



- Раньше был совсем другим фреймворком
- Он, кстати, имеет архитектуру совсем отличную от Spring Framework
- Версионировался отдельно от Spring Framework
- Сейчас - 5.*.*.RELEASE - ну т.е. со Spring
- Java 6+ для 4-й, Java 8+ для 5-й



Сайт проекта

Spring Security

[https://projects.spring.io/
spring-security/](https://projects.spring.io/spring-security/)



Документация

Spring Security Reference

<https://docs.spring.io/spring-security/site/docs/current/reference/htmlsingle/>



Документация

[http://
www.mkyong.com/
tutorials/spring-
security-tutorials/](http://www.mkyong.com/tutorials/spring-security-tutorials/)



Ваши вопросы?

05

Безопасность приложений

Спасибо за внимание!

