**План курса**

* 1

[**Настройка окружения и обзор архитектуры будущего приложения**](http://devstudy.net/course/jee-resume#lesson1)

Длительность урока: 57:55

* 1. Настройка окружения:
     1. FireFox - HTML браузер ([**https://www.mozilla.org/en-US/firefox/new/**](https://www.mozilla.org/en-US/firefox/new/));
     2. Firebug - плагин для FireFox, который позволяет просматривать структуру страницы и параметров HTTP запросов (**[http://getfirebug.com/](http://getfirebug.com/" \t "_blank)**);
     3. JDK 1.7 и настройка JAVA\_HOME (**[http://www.oracle.com/technetwork/java/javase/downloads/index.html](http://www.oracle.com/technetwork/java/javase/downloads/index.html" \t "_blank)**);
     4. Tomcat 8.0 и запуск в debug режиме ([**https://tomcat.apache.org/**](https://tomcat.apache.org/));
     5. Postgresql 9.5.0 - сервер базы данных (**[http://www.postgresql.org/download/](http://www.postgresql.org/download/" \t "_blank)**);
     6. PgAdmin - админпанель для работы с Postgresql;
     7. SQL Manager Postgres - SQL клиент для работы с сервером Postgresql (**[http://www.sqlmanager.net/en/products/postgresql/manager](http://www.sqlmanager.net/en/products/postgresql/manager" \t "_blank)**);
     8. Eclipse Mars - IDE для разработки на Java ([**http://www.eclipse.org/downloads**](http://www.eclipse.org/downloads));
     9. Git for windows - инструмент системы контроля версий Git for windows (**[https://git-scm.com/download/win](https://git-scm.com/download/win" \t "_blank)**).
  2. Постановка задачи - разработка web приложения по составлению резюме
  3. Обзор архитектуры - использование шаблона MVC для проекта и определение правил взаимодействия между компонентами
  4. Создание Maven проекта в Eclipse для данной системы, определение структуры проекта и загрузка кода на github
  5. Напоминание возможностей Servlet, Listener, Filter, JSP компонентов для данного проекта
  6. Запуск Tomcat в отладочном режиме и отладка web приложений
  7. Основные правила передачи управления между Servlet, Filter и JSP при использование MVC шаблона

В jsp параметер trimDirectiveWhitespaces=*"true"* урезает пробелы  
<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"* trimDirectiveWhitespaces=*"true"*%>

# sitemesh

1)добавить депенденси в пом

<https://mvnrepository.com/artifact/org.sitemesh/sitemesh/3.0.1>

<http://wiki.sitemesh.org/wiki/display/sitemesh3/Getting+Started+with+SiteMesh+3>

2)добавить фильтр в web.xml

<filter>

<filter-name>sitemesh</filter-name>

<filter-class>org.sitemesh.config.ConfigurableSiteMeshFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>sitemesh</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

3)Создать template/page\_template.jsp и добавить необходимый минимум на jsp

<html>

<head>

<title><sitemesh:write property='title'/></title>

<sitemesh:write property='head'/>

</head>

<body>

<sitemesh:write property='body'/>

</body>

</html>

The configuration file should be created at **/WEB-INF/sitemesh3.xml**:

<sitemesh>

<mapping path="/\*" decorator="/decorator.html"/>

</sitemesh>

Или в моем проекте   
<sitemesh>

<mapping path=*"/\*"* decorator=*"/WEB-INF/template/page-template.jsp"*/>

</sitemesh>

4)

Для page\_template.jsp вырезаем отдельно Header.jsp

footer.jsp

nav.jsp

css.jsp

js.jsp

и так далее , подключаем с помощью

<jsp:include page=*"../section/header.jsp"*></jsp:include>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"* trimDirectiveWhitespaces=*"true"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title><sitemesh:write property=*'title'* /></title>

<sitemesh:write property=*'head'* />

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<title>Test Bootstrap</title>

<jsp:include page=*"../section/css.jsp"*></jsp:include>

</head>

<body>

<jsp:include page=*"../section/header.jsp"*></jsp:include>

<jsp:include page=*"../section/nav.jsp"*></jsp:include>

<section class=*"main"*> <sitemesh:write property=*'body'* /> </section>

<jsp:include page=*"../section/footer.jsp"*></jsp:include>

<jsp:include page=*"../section/js.jsp"*></jsp:include>

</body>

</html>

5)Добавить profile.jsp - и сюда весь див тела главной страницы <div class=”container”>

добавить контороллер ProfilerController.java @WebServlet("/profile")

**protected** **void** doGet(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {

req.getRequestDispatcher("/WEB-INF/JSP/profile.jsp").forward(req, resp);

}

Проверяем работает ли

6) режим тело на отдельные дивы и размещаем их по тегам

WEBINF/tag – здесь им место по дефолту

Подключаем теги в profile.jsp

Сначала подключаем библиотеку тегов profile.jsp

<%@ taglib prefix="resume" tagdir="/WEB-INF/tags" %>

Потом располагаем теги по колонкам на странице  
<div class=*"col-md-4 col-sm-6 col-xs-6 .hidden-sm"*>

<resume:profile-main></resume:profile-main>

<resume:profile-languages></resume:profile-languages>

<resume:profile-hobbies></resume:profile-hobbies>

<resume:profile-information></resume:profile-information>

</div>