

Семинар №7

Задачи семинара

1. Вопросы и обсуждение семинара №7
2. Tomcat — сервер Web приложений
3. Web приложение
4. Servlet — компонент для обработка HTTP запросов
5. JSP — средство динамической генерации HTML

Материалы

1. Tomcat 7.0 <http://tomcat.apache.org/download-70.cgi>
2. J2EE Tutorial Web tier - <http://docs.oracle.com/javaee/5/tutorial/doc/geysj.html>
3. JSP <http://docs.oracle.com/javaee/5/tutorial/doc/bnagx.html>
4. JSTL <http://docs.oracle.com/javaee/5/tutorial/doc/bnakc.html>

Java — задания

1. Скачать и распаковать Tomcat 7.0
2. В папке bin находится скрипт запуска catalina.sh или catalina.bat Запустить tomcat
3. Убедится что открывается <http://localhost:8080>
4. Посмотреть примеры
 1. <http://localhost:8080/examples/servlets/>
 2. <http://localhost:8080/examples/jsp/>
5. Создать мавен проект на основе **maven-archetype-webapp**
6. Заменить заголовок web.xml на тот что в webapps/examples/WEB-INF/web.xml и удалить metadata-complete="true"
7. Проверить наличие файла index.jsp
8. Разместить приложение путем сборки *mvn package*
9. Скопировать файл *.war в папку webapps и проверить доступность приложения в браузере. Используйте logs/catalina.out для мониторинга
10. Добавить директорию java внутри src/main и использовать ее для хранения Java кода
11. В maven зависимости (pom.xml) добавить
 1. javax.servlet-api (<scope>provided</scope>)
12. Изменить уровень поддержки Java в pom.xml

```
<build>
<finalName>.....</finalName>
<plugins>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.0</version>
```

```

    <configuration>
      <source>1.7</source>
      <target>1.7</target>
    </configuration>
  </plugin>
</plugins>
</build>

```

13. Добавить простейший servlet:

```

@WebServlet( name = "SimpleServlet", urlPatterns = {"/simple"} )
public class Hello extends HttpServlet{
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        resp.getOutputStream().println("Hello");
    }
}

```

14. Проверить его доступность в браузере

15. Создайте JSP страницу в папки /WEB-INF/

16. Используйте

```
req.getRequestDispatcher("/WEB-INF/test.jsp").forward(req, resp);
```

для передачи управления от сервлета на страницу

17. Используйте JSTL & EL для вывода данных

18. Используйте req.setAttribute для передачи данных между servlet и JSP
для передачи управления от сервлета на страницу

19. Скопируйте БД H2 jar файл (например h2-1.3.170.jar) в Tomcat/lib директорию и перезапустите Tomcat

20. Добавьте \$TOMCAT_HOME/conf/context.xml

```

<Context>

<Resource name="jdbc/testDS" auth="Container"
    type="javax.sql.DataSource"
    driverClassName="org.h2.Driver"
    url="jdbc:h2:mem:test"
    username="sa" password=""
    maxActive="20" maxIdle="10" maxWait="-1"
    description="Datasource to database" />

```

21. Добавьте ссылку на ресурс в WEB-INF/web.xml

```

<resource-ref>
  <res-ref-name>jdbc/testDS</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>

```

22. Используйте в сервлете аннотацию @Resource для получения DataSource. Убедитесь удаётся получить соединение до базы данных.

```

@Resource(name = "jdbc/testDS")
private DataSource ds;

```

Написать WEB приложение для ведения гостевой книги

1. Написать страницу со списком сообщений (отсортированных от новых к старым) и форму добавления нового сообщения