

02. Основы WEB и Java

Лекции по информатике
для студентов второго курса Высшей школы ИТИС КФУ
2020

Ференец Александр Андреевич

старший преподаватель кафедры программной инженерии

С использованием материалов
к. т. н., доцента кафедры программной инженерии Абрамского М.М.

aferenets@it.kfu.ru

HTTP / WWW. Схема запроса (повторяем)

METHOD path Protocol

Header-name: header-value

Header-name: header-value

Body

POST /forum/topic.php?id=42 HTTP/1.1

Host: example.com

Content-type: application/x-www-form-urlencoded

Content-length: 375

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3

Referer: https://www.google.ru/

Connection: keep-alive

text=Hello%20world&title=Post%20title.....

HTTP / WWW. Схема ответа (повторяем)

Protocol CODE STATUS

Header-name: header-value

Header-name: header-value

Body

HTTP/1.1 200 OK

Date: Wed, 27 JUL 2016 11:20:59 GMT

Server: Apache

X-Powered-By: PHP/5.6.3-2ubuntu5wm1

Last-Modified: Wed, 27 JUL 2016 11:20:59 GMT

Content-Language: ru

Content-Type: text/html; charset=utf-8

Content-Length: 2437

Connection: keep-alive

<!DOCTYPE html>

<html>

...

User-Agent. Название браузера, его движка, ОС и проч. В самих браузерах можно изменить с помощью расширений.

Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0

Accept-Language. Какие языки/локаль ожидаются браузером (см. ISO/IEC 15897) + доп.инфа

ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3

Referer. Кто побудил браузер сделать запрос (напр., адрес страницы со ссылкой)

Referer: <https://www.google.ru/>

DNT. Просьба не отслеживать активность браузера (см.настройки некоторых браузеров)

DNT: 1

Date. Время генерации ответа веб-сервера.

Wed, 27 JUL 2016 11:20:59 GMT

Content-Type и Content-Length. Тип в формате MiMe и размер в байтах.

text/html; charset=utf-8

X-Powered-By и другие X-HEADER. Как правило нестандартные заголовки. Можно придумать свои для добавления своей тех.информации веб-приложения. Впрочем согласно **RFC6648** (2012 г.) не рекомендуется придумывать названия без префикс **X**.

X-Powered-By: PHP/5.6.3-2ubuntu5wm1
x-cache: cp3058 miss, cp3052 hit/6 (wikipedia.org)

Last-Modified. Время последнего изменения

Wed, 27 JUL 2016 11:20:59 GMT

ETag. Указание спец. идентификатора/хэша для описания версии кэшируемого документа.

675af34563dc-tr34

Cache-Control. Заголовок с широкими возможностями описания опций кэширования.

max-age=<seconds>

Must-revalidate

public

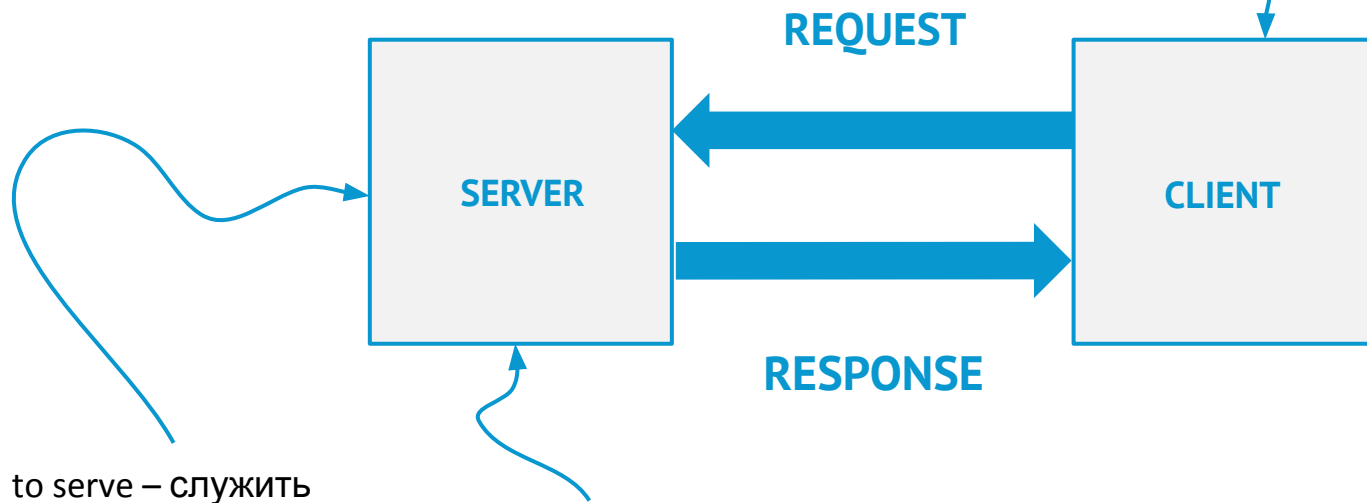
public, max-age=432000



Демонстрация заголовков в инструментах разработчика в браузере для запросов-ответов при клике по ссылке, отправке форм и AJAX-запросах.

HTTP / WWW. Ещё раз об архитектуре

- Браузер
- Мобильное приложение
- Программа Java, PHP, ...
- ...



WEB-Server, но бывают другие для других протоколов

ПК, мобильные, IoT-устройства
(чайники)

браузер

git-client

ПОЧТОВЫЙ
КЛИЕНТ

CS 1.6

Skype

Серверная машина / Хост

Операционная
система

?

веб-сервер

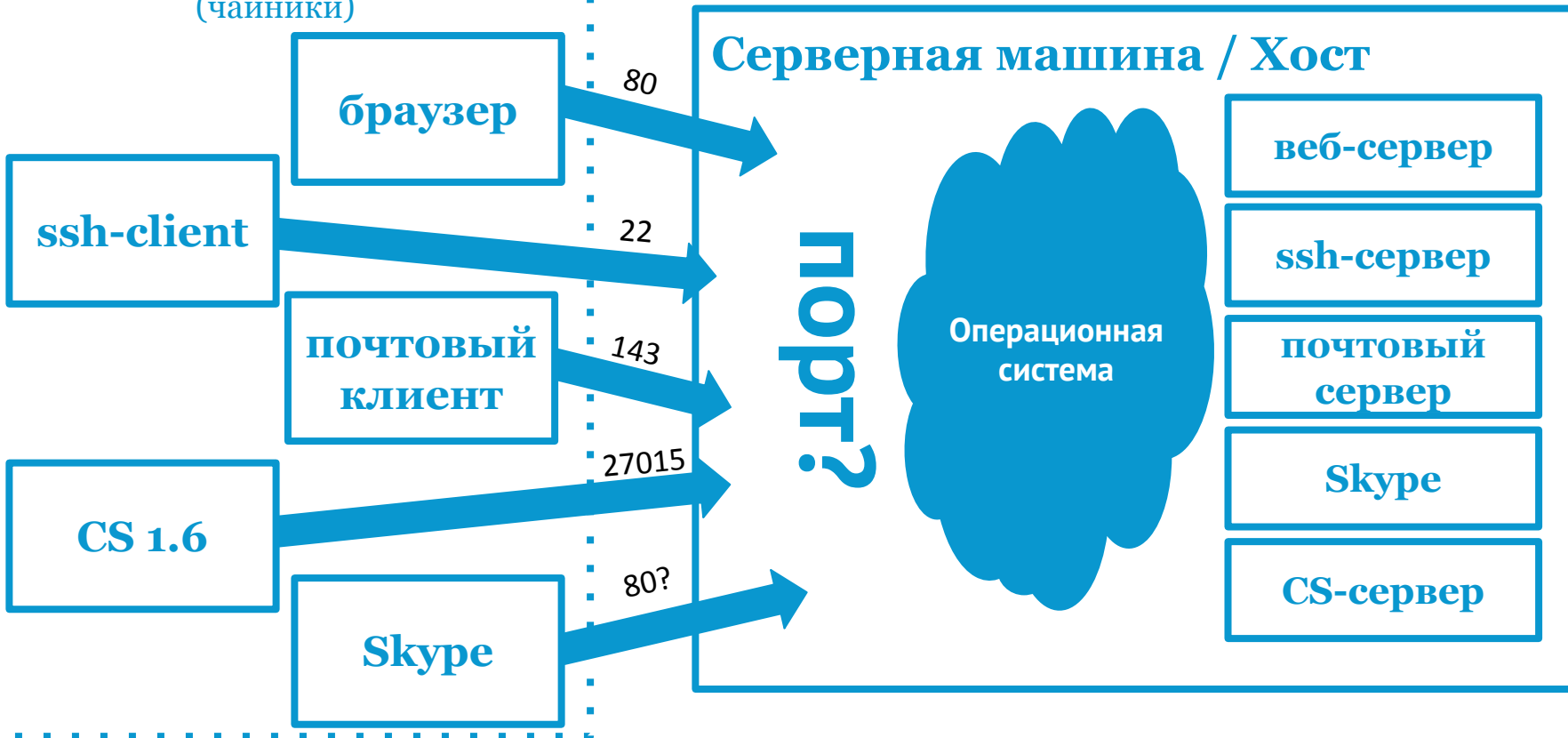
git-сервер

ПОЧТОВЫЙ
сервер

Skype

CS-сервер

ПК, мобильные, IoT-устройства
(чайники)



Число 0-65535. IANA выдаёт. TCP и UDP

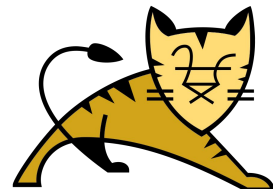
- **80** HTTP
- **8080** HTTP (альтернативный)
 - **20/21** FTP
 - **22** SSH
 - **143** IMAP
- **443** HTTPS (SSL/TLS)
 - **3306** MySQL
 - **5432** PostgreSQL

Application Server – это полная реализация всех спецификаций стандарта JavaEE: распределенные транзакции, EJB, пулинг соединений к БД и т.д.. Используется для хостинга полноценных enterprise-приложений (.ear). Примеры – JBoss, TomEE, Glassfish, WebSphere.

Servlet Container – это реализация минимальной необходимой части JavaEE – Servlet API и JSP. Умеет хостить статический контент, jsp-страницы и выполнять Java код сервлетов. Остальные спецификации JavaEE при необходимости добавляются сторонними библиотеками-реализациями. Примеры – Tomcat, Jetty.

Каждый сервер приложений содержит в себе контейнер сервлетов. Spring Framework – альтернатива JEE, поэтому для запуска приложения на основе Spring обычно достаточно веб-контейнера. Если же приложение использует всю мощь JEE, необходим сервер приложений.

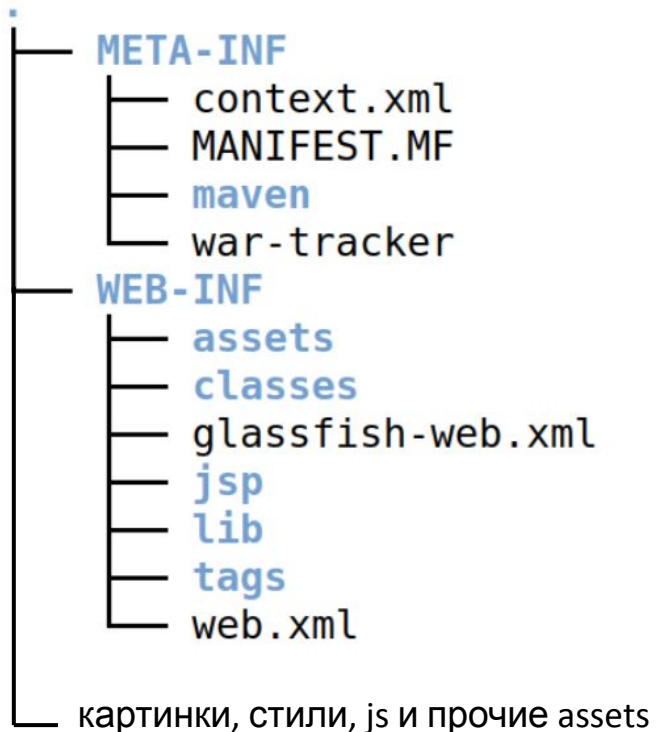
<https://itsobes.ru/JavaSobes/chem-otlichaetsia-servlet-container-ot-application-server/>



APACHE. Не только сервер



```
ist@ist-notebook:~/tomcat9/webapps/spring-bookcatalogue-0.1$ tree -L 2
```



```
package javax.servlet;
```

```
...
```

```
public interface Servlet {
```

вызывается при загрузке сервлета в контейнере сервлетов

```
    void init(ServletConfig var1) throws ServletException;
```

```
    ...
```

```
    void service(ServletRequest var1, ServletResponse var2) throws ServletException, IOException;
```

```
    ...
```

```
    void destroy();
```

вызывается при удалении сервлета из контейнера сервлетов

вызывается при необходимости выполнить именно данный сервлет для обработки запроса

- Данные запроса берем из request,
- Результат пишем в response.

```
}
```

1. До начала обслуживания запросов (если сервлет еще ни разу не загружался):
 - i. Tomcat загружает класс сервлета,
 - ii. Tomcat создает экземпляр класса сервлета,
 - iii. Tomcat вызывает метод `init()` сервлета.
2. Обслуживание запросов:
 - i. Tomcat получает запрос, определяет, какой сервлет использовать и вызывает у него `service(...)`.
 - ii. Каждый запрос обслуживается в отдельном потоке.
3. Изъятие сервлета из эксплуатации:
 - i. Tomcat вызывает метод `destroy()`

Servlets. Mapping – web.xml

```
<web-app metadata-complete="true" version="4.0" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee">
  <description> Servlet and JSP Examples. </description>
  <display-name>Servlet and JSP Examples</display-name>
  <request-character-encoding>UTF-8</request-character-encoding>
```

```
<servlet>
  <servlet-name>simpleimagepush</servlet-name>
  <servlet-class>http2.SimpleImagePush</servlet-class>
</servlet>
```



```
<servlet-mapping>
  <servlet-name>simpleimagepush</servlet-name>
  <url-pattern>/servlets/serverpush/simpleimage</url-pattern>
</servlet-mapping>
```

```
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.xhtml</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

...и другие параметры

```
@WebServlet("/main")  
public class TestServlet extends HttpServlet{
```

- @WebServlet
- @WebFilter
- @WebListener

...и другие параметры

- **doGet** (HttpServletRequest req, HttpServletResponse resp)
- **doPost** (HttpServletRequest req, HttpServletResponse resp)
- **doPut** (HttpServletRequest req, HttpServletResponse resp)
- **doDelete** (HttpServletRequest req, HttpServletResponse resp)

- **doOptions** (HttpServletRequest req, HttpServletResponse resp)
- **doHead** (HttpServletRequest req, HttpServletResponse resp)

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
    PrintWriter out = resp.getWriter();  
    out.println("Hello world");  
}
```