

Військовий інститут телекомунікацій та інформатизації  
Кафедра №22 «Бойового застосування АСУВ»



# Веб-сервер

## *Apache Tomcat 6*

Доповідач: курсант 222 навч. групи  
солдат Казаков А.В.

# Зміст доповіді

1. Ключові питання організації та веб-сервера
  1. Поняття веб-сервера і його функції
  2. Протокол HTTP
  3. Аутентифікація та авторизація користувачів
  4. Управління сесіями та Cookies
  5. Постійне HTTP-з'єднання
2. Веб-сервер Apache Tomcat 6
  1. Структура каталогів сервера



# 1.1 Поняття веб-сервера і його функції

Веб-сервер - це програмне забезпечення, що забезпечує доставку контенту кінцевому користувачу по мережі.

Веб-сервер реалізує серверну частину протоколу HTTP.



# Функції та особливості веб-серверів:

- передача контенту користувачеві;
- отримання контенту від користувачів;
- підтримка динамічно генеруюємих сторінок;
- аутентифікація та авторизація користувачів;
- ведення журналу звернень користувачів до ресурсів;
- підтримка HTTPS для захищених з'єднань з клієнтами.





# Реалізації веб-сервісів:

*Історично першим вважається веб-сервер  
CERN httpd (1991 рік)*

## Java:

- Apache Tomcat
- Jetty

## Інші:

- Apache
- Nginx
- Lighttpd



## 1.2 Протокол НТТР

- HyperText Transfer Protocol - текстовий протокол передачі даних прикладного рівня
- Поточна версія - 1.1. Прийнята в 1999 році. У цій версії доданий режим «постійного з'єднання» з сервером.



## HTTP-пакет складається з:

- Стартовий рядок:  
GET /page.php HTTP / 1.1 - запит  
HTTP / 1.1 404 Not Found - відповідь
- Заголовки - розділені двокрапкою пари параметр-значення.  
Content-Type: text / plain; charset = utf8
- Тіло повідомлення - будь-який текст, у тому числі закодований або стиснений.

## HTTP-методи:

HEAD, GET, OPTIONS, TRACE - безпечні  
POST, PUT, DELETE - небезпечні



# 1. 3 Аутентифікація і авторизація користувачів

- Basic access authentication - логін і пароль кодуються за допомогою Base64, підтримується всіма браузерами
- Digest access authentication - логін і пароль шифруються алгоритмом MD5
- При доступі до захищеному контенту клієнт отримує відповідь:
- HTTP / 1.1 401 Authorization Required
- WWW-Authenticate: Basic realm = "Secure Area"
- або
- WWW-Authenticate: Digest realm = "testrealm@host.com", nonce = "dcd98b7102dd2f0e8b11d0f600bfb0c093"
- Авторизація не підтримує протоколом HTTP і повністю покладається на веб-сервер





# 1. 4 Управління сесіями і Cookie

Веб-додаткам буває необхідно контролювати пересування клієнтів по сайту. HTTP не відстежує стан своїх клієнтів (stateless протокол). Можливі рішення даної проблеми:

- Додавання параметра `session_id` в URL:
- функції з контролю часу життя сесії покладаються на веб-сервер
- сесія не зберігається після перезапуску браузера
- Зберігання `session_id` в cookies браузера.

**Cookies**- невеликий фрагмент даних, створений веб-сервером і зберігається на комп'ютері користувача у вигляді файлу, який браузер може пересилати веб-серверу в HTTP-запиті:

- функції з контролю часу життя сесії покладаються на браузер
- сесія зберігається після перезапуску браузера



# 1.5 Постійне НТТР-з'єднання

Протокол НТТР підтримує можливість відправки декількох запитів веб-серверу відразу, не створюючи при цьому нових ТСП-з'єднань.

Для цього в першому запиті в заголовку передається параметр Connection: Keep-Alive.

За замовчуванням всі сполуки є постійними.



## Переваги постійного з'єднання:

- менше навантаження на пам'ять, мережу і процесор, тому створюється менше ТСП-з'єднань.
- можливість відправки відразу декількох запитів, не чекаючи відповідей (http-pipeline).
- можливість очікування відповіді на протязі тривалого проміжку часу (server-push).



## 2. Apache Tomcat 6

- Apache Tomcat 6 є одним з найбільш популярних Web-серверів, що реалізують специфікацію JEE 5 (Java Enterprise Edition). Він безкоштовно розповсюджується для комерційного та некомерційного використання під ліцензією Apache Software License з відкритими початковими кодами.
- Сам продукт, документацію та вихідні коди можна знайти на

**<http://tomcat.apache.org>**



## Apache Tomcat складається з таких компонентів:

1. Web Connector Coyote, який реалізує протокол HTTP / 1.1, за допомогою якого користувач, використовуючи Інтернет-браузер, може надсилати запити до сервера й отримувати відповідь.
2. Web Container Catalina реалізує специфікацію Servlet API 2.5 з JEE 5. Servlet API є основою для всіх інших технологій Java стосуються Web і дає можливість динамічно генерувати будь-який Web-контент, використовуючи будь бібліотеки, доступні для java.
3. Jasper Compiler - компілятор JSP-сторінок (підтримує специфікацію JSP 2.1). JSP сторінка є найбільш популярним (але не єдиним) засобом створення динамічно-генеруючих HTML, XML та інших документів, що мають текстове представлення.





# 2.1 Структура каталогів сервера

**/bin/** - скрипти включення, виключення та ін.

**/startup.bat (startup.sh)** – включення.

**/shutdown.bat (shutdown.sh)** – виключення.

**/catalina.bat (catalina.sh)** – скрипт включення та налаштування параметрів (викликається з *startup.bat*)

**/conf/** - конфігураційні файли.

**/server.xml** – основний конфігураційний файл, тут задаються порти, коннектори та ін.

**/web.xml** – Включається в усі web.xml файли додатків за замовчуванням.

**/lib** – бібліотеки сервера, всі бібліотеки з цієї папки доступні всім додаткам. Сюди необхідно покласти JDBC-драйвер, якщо додаток використовує БД.

**/log** – логи (протоколи) сервера. Використовуються для статистики, діагностики та налагодження.

**/temp** – папка для тимчасових файлів сервера і додатків.

**/webapps** – папка для додатків. Все, що тут лежить буде розгорнуто і запущено при старті сервера, а так само під час його роботи (hot-deploy).

**/ROOT** – спеціальне ім'я для кореневого контексту. Додаток в цій папці буде прив'язано до контексту "/", тобто доступно через `http://localhost:8080`

**/docs, examples, host-manager, manager, ROOT** – приклади додатків, документація та адміністративна консоль, що входять в поставку сервера. Їх можна без шкоди видалити.

**/work** – робоча папка сервера. Сюди будуть складатися скомпільовані на льоту JSP файли, серіалізовані сесії користувачів (щоб рестарт сервера не викинув користувачів із системи) та ін. Вміст можна (і, іноді, потрібно) без шкоди видалити.



*Дякую  
за увагу!*

