

Javaプログラマ育成コース

第七回 ウェブ開発とServlet

ソフトシンク株式会社

代表取締役

周 順彩

zhousc@soft-think.com

目次

- ウェブ開発
- Servlet
- 練習課題
- 次回の予習タスク

ウェブ

◆ ウェブとは

ワールド-ワイド-ウェブの略。インターネット上で文字・画像などをレイアウトして見せ、簡単にアクセスできるようにするための仕組み。

◆ 分類:

静的ページ

動的ページ

静的ページ

◆ 静的ページとは

作成したページがそのまま表示されるようなページのことです。テキストファイルにHTMLやCSSをタグ打ちしてページを作成した場合は一旦サーバーにページをアップロードすると、修正したものを再度アップロードし直したり悪意のある攻撃によって改ざんされたりしない限りは、いつ誰が見ても同じ内容が表示されます。このようなページのことを静的ページと呼びます。

◆ 利点

静的ページは最初から用意されているページを表示するだけなので、ユーザがアクセスしてきたときに高速で表示させることができる。

ウェブサーバ

◆ ウェブサーバとは

Webブラウザからのリクエストに応じて静的画面や画像を表示する働きをするものだ。



ウェブサーバ

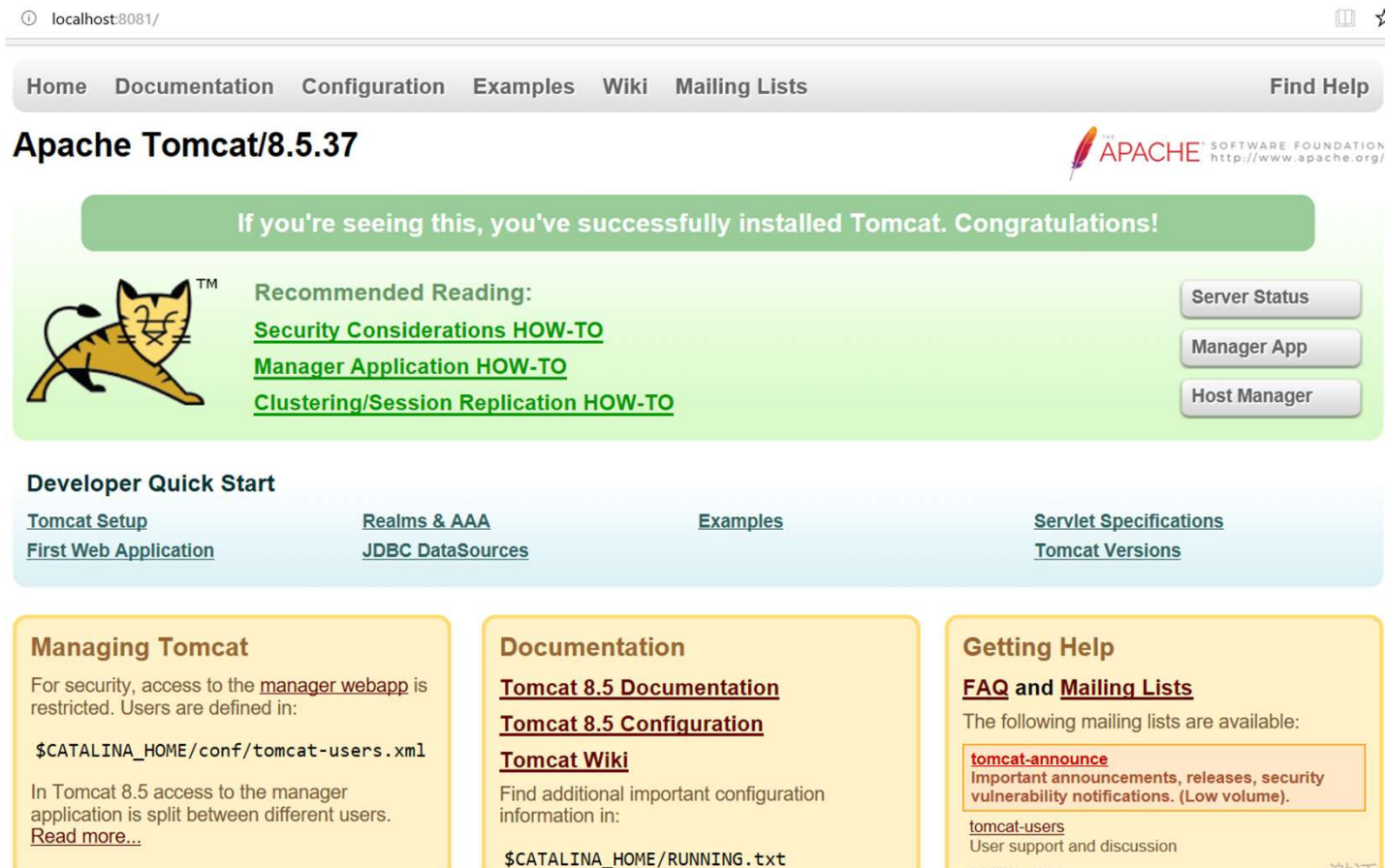
主なウェブサーバ

- | | |
|-----------------|-------------|
| ① WebLogic | ---Oracle |
| ② WebSphere | ---IBM |
| ③ Apache/Tomcat | ---apache |
| ④ IIS | ---Software |

Tomcat

Tomcatとは

Apache Tomcat(アパッチトムキャット)は、Java ServletやJavaServer Pages (JSP)を実行するためのWebコンテナ(サーブレットコンテナ、サーブレットエンジン)である。




localhost:8081/

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/8.5.37

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 **Recommended Reading:**

- [Security Considerations HOW-TO](#)
- [Manager Application HOW-TO](#)
- [Clustering/Session Replication HOW-TO](#)

Server Status
Manager App
Host Manager

Developer Quick Start

- [Tomcat Setup](#)
- [First Web Application](#)
- [Realms & AAA](#)
- [JDBC DataSources](#)
- [Examples](#)
- [Servlet Specifications](#)
- [Tomcat Versions](#)

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 8.5 access to the manager application is split between different users.
[Read more...](#)

Documentation

[Tomcat 8.5 Documentation](#)
[Tomcat 8.5 Configuration](#)
[Tomcat Wiki](#)

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Getting Help















[FAQ](#) and [Mailing Lists](#)

The following mailing lists are available:

- [tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).
- [tomcat-users](#)
User support and discussion

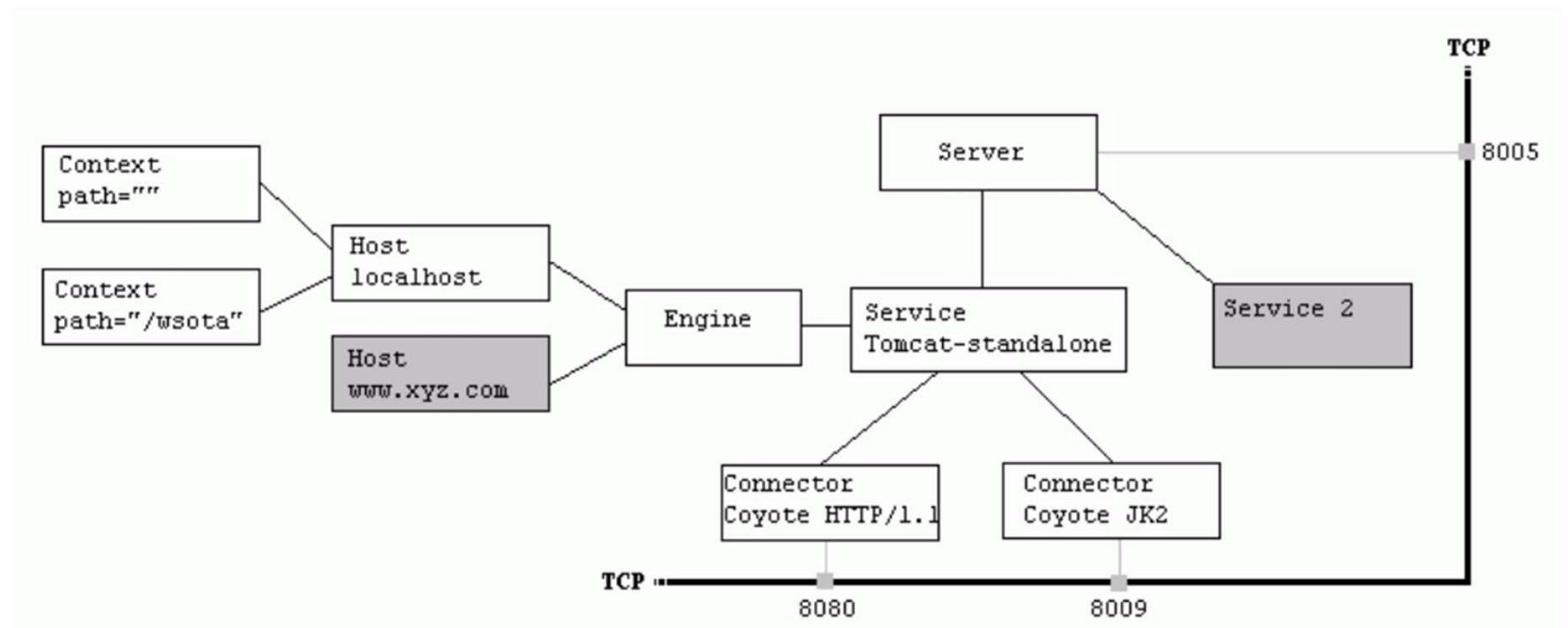
Tomcatディレクトリ構造

脳 > Windows (C:) > tomcat > apache-tomcat-8.5.37

名称	↑	修改日期	类型	大小
 bin		2018/12/12 12:07	文件夹	
 conf		2019/2/10 14:55	文件夹	
 lib		2018/12/12 12:07	文件夹	
 logs		2019/2/10 14:55	文件夹	
 temp		2018/12/12 12:07	文件夹	
 webapps		2019/2/10 16:08	文件夹	
 work		2019/2/10 14:55	文件夹	
 BUILDING.txt		2018/12/12 12:07	文本文档	20 KB
 CONTRIBUTING.md		2018/12/12 12:07	MD 文件	7 KB
 LICENSE		2018/12/12 12:07	文件	57 KB
 NOTICE		2018/12/12 12:07	文件	2 KB
 README.md		2018/12/12 12:07	MD 文件	4 KB
 RELEASE-NOTES		2018/12/12 12:07	文件	8 KB
 RUNNING.txt		2018/12/12 12:07	文本文档	17 KB

コンフィグファイル

Tomcatの設定はいくつかの設定ファイルで行われますが、サーバについての主要な設定を行うのがserver.xmlです。server.xmlはいくつかのタグで情報構成されており、それぞれのタグの組み合わせにより入れ子の階層型に構築された設定を持ちます



URL

"URL"とは「Uniform Resource Locator」の略称で、インターネット上に存在する情報資源（文書や画像など）の場所を指し示す技術方式です。通常「プロトコル://ドメイン名/ディレクトリパス名/ファイル名」という形式で構成されるホームページページの住所（アドレス）のことです。

例:

<http://www.zxb.cn:8080/JavaWebApp/hello.html>

- ① www.zxb.cnをもって、ローカルのHostsファイルにIPアドレスを探す。
- ② ①は探せないと、DNSサーバに探していく。
- ③ IPアドレスよりウェブサーバにつながる。
- ④ 請求を送信する。
- ⑤ 結果を返す。

HTTP

Hypertext Transfer Protocol (HTTP) は HTML などのハイパーメディア文書を転送するためのアプリケーション層プロトコルです。このプロトコルはウェブブラウザ (クライアント) とウェブサーバー間の通信を目的として設計されていますが、他の用途でも使用されることがあります。

- HTTPリクエスト
- HTTPレスポンス

HTTPリクエスト

HTTPリクエストは、大きく三つの部分に分かれるように

1. HTTPリクエスト行
2. HTTPヘッダー行
3. HTTPボディ行

```
GET /www.yoheim.net/?key1=value1&key2=value2 HTTP/1.1
Accept:text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Charset:Shift_JIS,utf-8;q=0.7,*;q=0.3
Accept-Encoding:gzip,deflate,sdch
Accept-Language:ja,en-US;q=0.8,en;q=0.6
Cache-Control:max-age=0
Connection:keep-alive
Cookie:utma=something;
Host:www.yoheim.net
User-Agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.54 S
afari/536.5
```

HTTPレスポンス

HTTPリクエストは、大きく三つの部分に分かれるように

1. HTTPステータス行
2. HTTPヘッダー行
3. HTTPボディ行

```
HTTP/1.1 200 OK
Date: Sun, 25 Mar 2018 14:19:50 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
ETag: W/"ZGFmaWhlcmlvaGZnaWVhZ2h1akVybgFoZmdpbGVyaA"
Cache-Control: max-age=0, private, must-revalidate
Set-Cookie: _qiita_login_session=YWZlZ3Jld2d3aW9lZ3Zkbmp2Y252ZGthamZnaWVv; domain=.qiita.com; path=
/; expires=Mon, 25 Mar 2019 14:19:50 -0000; HttpOnly
X-Runtime: 0.253271
Strict-Transport-Security: max-age=2592000
X-Request-Id: 1650d346-2ed3-4bcb-9789-ed404ae13d31
Content-Security-Policy-Report-Only: default-src https: data: 'unsafe-eval' 'unsafe-inline'; report
-uri https://us-central1-qiita-csp-report.cloudfunctions.net/csp-report

<!DOCTYPE html><html><head><meta charset="utf-8" /><title>Qiita</title><meta content="width=device-
```

HTTPレスポンスステータスコード

HTTP レスポンス状態コードは、特定の HTTP リクエストが正常に完了したかを示します。レスポンスは情報レスポンス、成功レスポンス、リダイレクト、クライアントエラー、サーバーエラーの 5 つのクラスに分類されます。

コード	説明
100~199	情報
200~299	成功
300~399	リダイレクト
400~499	クライアントエラー
500-599	サーバーエラー

サーブレット

「Java Servlet(以下、サーブレットと呼称します)」とは、かんたんというと「WEBアプリを開発するために作られたJavaプログラムの部品」のことです。

作り方:

- ① serviceのインタフェースを実現する。
- ② web.xmlファイルにservletタグをコンフィグする。

★スレッドセーフ

サーブレットクラス

- ① Servlet
- ② GenericServlet
- ③ HttpServlet

web.xml

① `<servlet/>`と`<servlet-mapping>`

➤ `<servlet-name/>`

➤ `<url-pattern/>`

➤ `<init-param/>`

② `<context-param>`

URLパターン

① URLパターンは次の3種類のURLパターンを指定できる。

- 完全パス指定
- パス指定
- 拡張子指定

② 優先順位

完全一致のマッピング > パスマッピング > 拡張子マッピング

servletに関するクラス

- ① ServletConfig
- ② ServletContext
- ③ HttpServletRequest
- ④ HttpServletResponse

★文字化け

フィルター

Filterとはクライアントとサーバ間のリクエスト、レスポンス時に何らかの処理を行うです。ログの生成、パラメータ値の変更、フォーマット変更、認証などを処理を行う。

作り方:

- ①Filterインタフェースを実現する。
- ②web.xmlファイルにfilterタグをコンフィグする。

★FilterConfig

リスナー

- ①ServletContextListener
- ②HttpSession
- ③ServletRequest
- ④ServletContextAttributeListener
- ⑤HttpSessionAttributeListener
- ⑥ServletRequestAttributeListeners

次回の予習タスク

◆ HtmlとJavascript