**Bài9**

**Tổng quan vềứngdụng Web**

Module: JWBD

# Kiểm tra bàitrước

Hỏivàtraođổivềcáckhókhăngặpphảitrongbài“Bootstrap"

Tómtắtlạicácphầnđãhọctừbài“Bootstrap”

## Mụctiêu

* GiảithíchđượcmôhìnhWeb
* Trìnhbàyđượccácthànhphầncủamộthệthốngweb
* TạođượcmộtứngdụngwebJavacơbản
* Sử dụng đượcServlet

# Mô hìnhweb

Website làgì

Stac web và dynamicweb

HTTP

WebServerlàmviệcthếnào

Mộtsốcôngnghệphíaserver

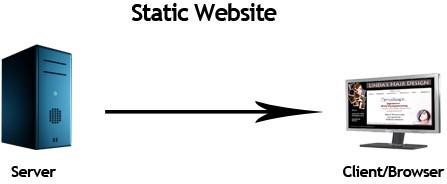
**Website làgì?**

* Tập hợp các trang web có thể chứa văn bản, hình ảnh, âm thanh, video.
* Trangđầutiêncủawebsiteđượcgọilàtrangchủ.
* MỗiwebsitecóđịachỉcụthểtrêninternetgọilàURL
* Website được lưu trữ trên một hoặc nhiều máychủ (Server)vàcó thể được truycập bằng cách tìm tới trang chủ củaWebsite thông quamạngmáytính.
* Websiteđượcquảnlýbởicánhân,côngtyhoặcmộttổchức.
* Cóhailoạiwebsitecơbảnlàwebđộng(dynamicweb)vàwebtính

(staticweb)

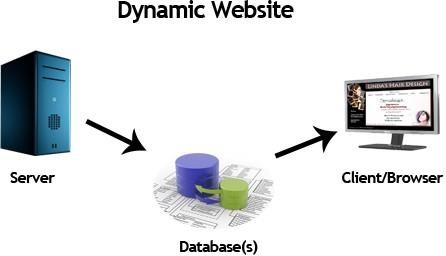
## Static Web

* Làloạiwebsitecơbảnđượctạomàkhôngcầnphảibiếtđếncác ngônngữlậptrìnhwebnhưJava,PHP,… hoặcthiếtkếcơsởdữ liệu.
* NhữngtrangwebtrongwebsitetĩnhđượcviếtbằngmãHTMLhoặc thêmCSS,JavaScriptđểthêmcáchiệuứng.



## DynamicWeb

* Websiteđộnglàwebsitecónộidungthayđổi.
* Sựthayđổi có thể là tùytheo thời gian,tùytheo người dùng,tùy theongữcảnh
* Đểtạođượcwebsiteđộng,chúngtathườngsửdụngđếncácngôn ngữ phía server (server-side), chẳng hạn như Servlet, JSP, PHP, Python,C#...
* Một websiteđộngthườngđượcđặttrênmộtmáychủdịchvụweb (WebServer)



# HTTP

Giao thức HTTP

Request,Response

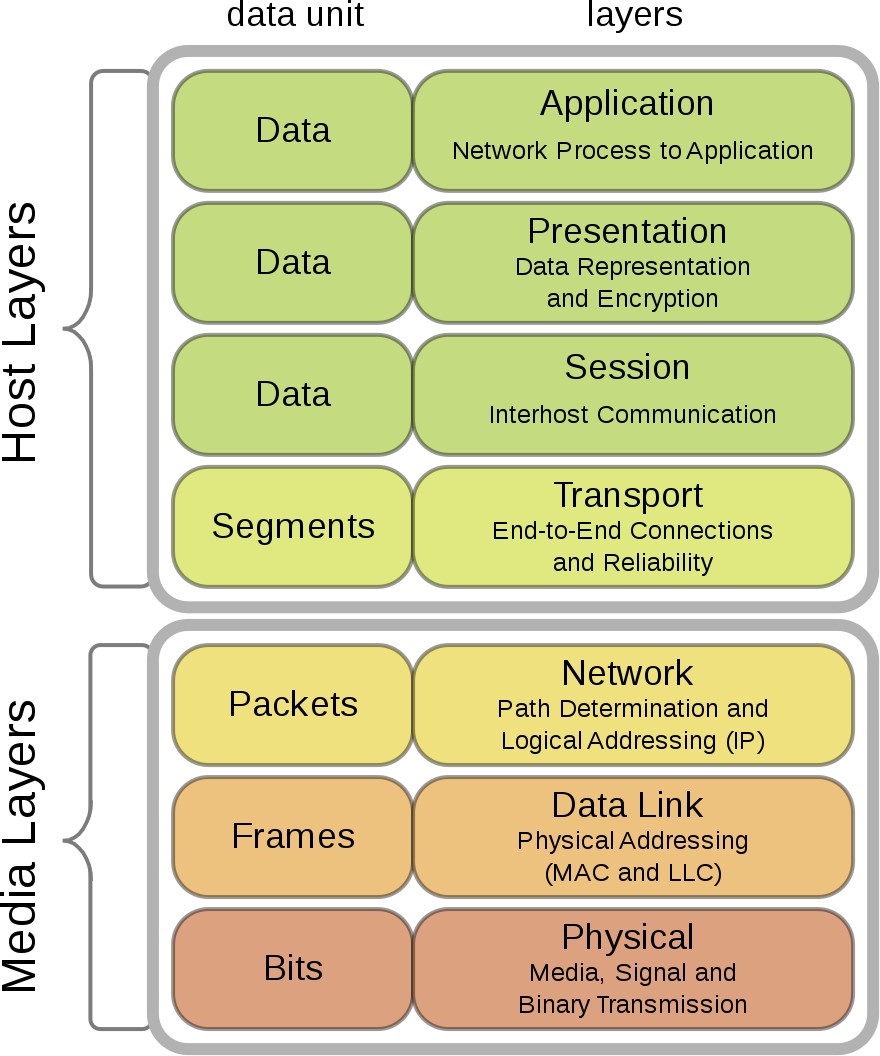
8

## Mô hìnhOSI

* **Mô hình OSI**(OpenSystemsInterconnectionReferenceModel, viếtngắnlà OSI Model hoặc OSI Reference Model) -tạm dịch là **Mô hình tham chiếu kết nối các hệ thốngmở**
* Làmộtthiếtkếdựavàonguyênlýtầngcấp,lýgiảimộtcáchtrừutượng kỹthuậtkếtnốitruyềnthônggiữacácmáyvitínhvàthiếtkếgiaothức mạng giữachúng.
* MôhìnhnàyđượcpháttriểnthànhmộtphầntrongkếhoạchKếtnốicác hệthốngmở(OpenSystemsInterconnection)doISOvàIUT-Tkhởixướng.

Nócònđượcgọilà**Mô hìnhbảytầngcủaOSI**.

## cấp của mẫu hìnhOSI

•Tầng 1: Tầng vậtlý (Physical Layer)

•Tầng2:Tầngliênkếtdữliệu(Data-LinkLayer)

•Tầng 3: Tầng mạng (NetworkLayer)

•Tầng4:Tầnggiaovận(TransportLayer)

•Tầng 5: Tầng phiên (Sessionlayer)

•Tầng6:Tầngtrìnhdiễn(Presentationlayer)

•Tầng 7: Tầng ứng dụng (Applicationlayer)

## ứng dụng -kháiniệm

* **Tầngứngdụng**làtầngthứbảytrongbảytầngcấpcủamôhìnhOSI.

•Tầngnàygiaotiếptrựctiếpvớicáctiếntrìnhứngdụngvàthihành nhữngdịchvụthôngthườngcủacáctiếntrìnhđó

* Tầngnàycòngửicácyêucầudịchvụtớitầngtrìnhdiễn
* Nhữngdịchvụthôngthườngcủatầngứngdụngcungcấpsựchuyển đổivềngữnghĩagiữacáctiếntrìnhứngdụngcóliênquan.

## ứng dụng -Vídụ

* HTTP, S-HTTP, (Secure) HyperText TransferProtocol

•Giaothứctruyềnsiêuvănbản(antoàn) • MIME,S-MIME:

•MultipurposeInternet MailExtensionsvàSecureMIME

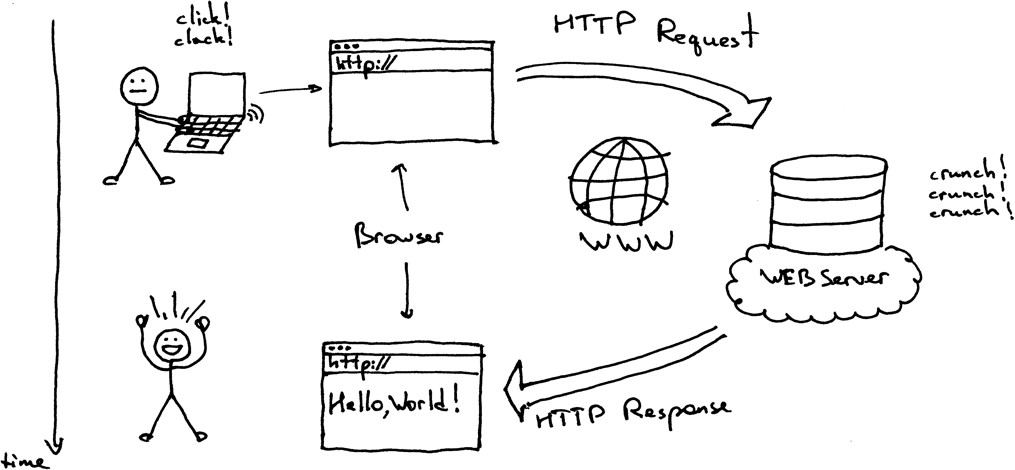
* SMTP

•SimpleMailTransferProtocol

## Giao thứcHTTP

* HyperText TransferProtocol
* Giao thức truyền tải siêuvăn bản
* LàmộttrongnămgiaothứcchuẩnvềmạngInternet,đượcdùngđể liênhệthôngtingiữaMáycungcấpdịchvụ(Webserver)vàMáysử dụngdịchvụ(Webclient)
* LàgiaothứcClient/ServerdùngchoWorldWideWeb-WWW
* **HTTP**làmộtgiaothứcứngdụngcủabộgiaothứcTCP/IP(cácgiao thứcnềntảngchoInternet)

## Mô hình Client -Server



## HTTP StatusCode

* MãtrạngtháiHTTPđượcserverphảnhồilạimỗikhinhậnđượchttp resquest
* YếutốStatus-Codelàmộtsốnguyên3kýtự

•Kýtựđầutiêncủamãhóatrạngtháiđịnhnghĩa hạng(loại)phảnhồi

•Haikýtựcuốikhôngcóbấtcứvaitròphânloạinào • Ký tự đầu gồm có: 1,2,3,4,5 loạilỗi.

## HTTP StatusCode



## HTTPRequest

•MộtHTTPclient(máykhách)gửimộtHTTPrequest(yêucầu)lênserver(máy chủ)nhờmộtthôngđiệpcóđịnhdạngnhưsau:

•MộtdòngRequest-line

•Khôngcóhoặccóthêmcácheader(General|Request|Entity)theosaubởi mộtkýhiệuCRLF(carriagereturnlinefeed-báohiệutrởvềđầudòngtiếp theo)

•Mộtdòngtrống(VDmộtdòngmàkhôngcógìởtrướcCRLF)báohiệukết thúcphầnheader

•Cóhoặckhôngcóphầnnộidungthôngđiệp

## HTTPRequest



## HTTP Response

• Khi nhận và phiên dịch một HTTP Request

•Server sẽ gửi tín hiệu phản hồi là một HTTP Response bao gồm các thành phần sau:

•Một dòng trạng thái (Status-Line)

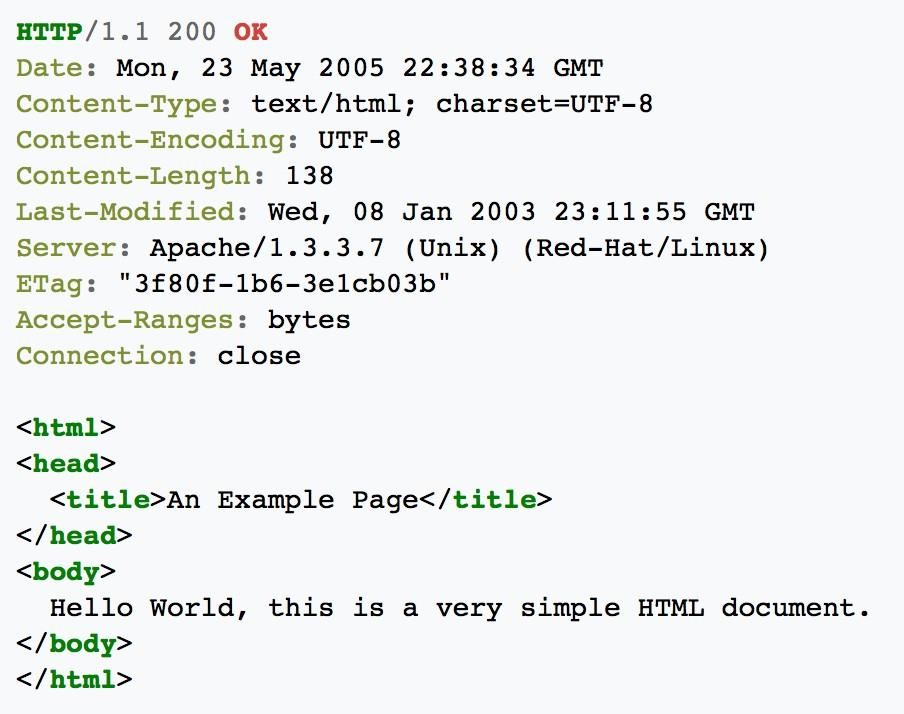
•Không hoặc nhiều hơn các trường Header

(General|Response|Entity) được theo sau CRLF

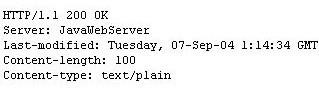
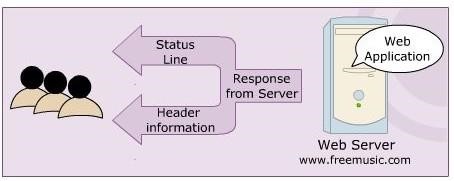
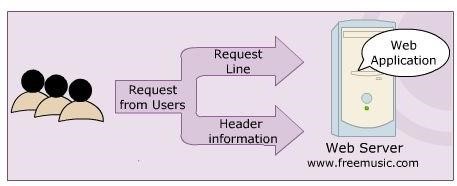
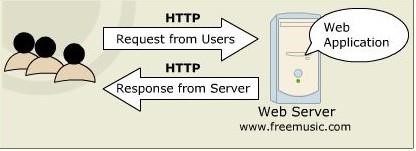
•Một dòng trống chỉ dòng kết thúc của các trường Header

•Một phần thân thông báo tùy ý

## HTTP Response



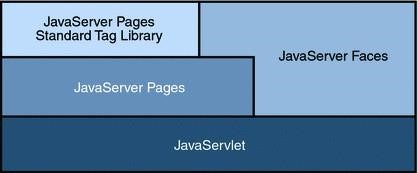
**Web Server làm việc như thếnào?**



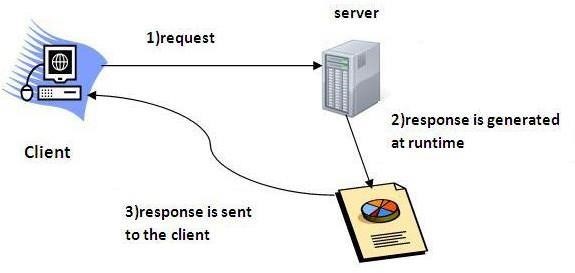
### Một số công nghệ phíaserver

* CGI -Common GatewwayInterface
* SSJS -Server-side JavaScripts
* PHP –Personal HomePage
* JavaServlets
* JSP -Java ServerPages
* ASP -Active ServerPages

### Công nghệ JavaWeb



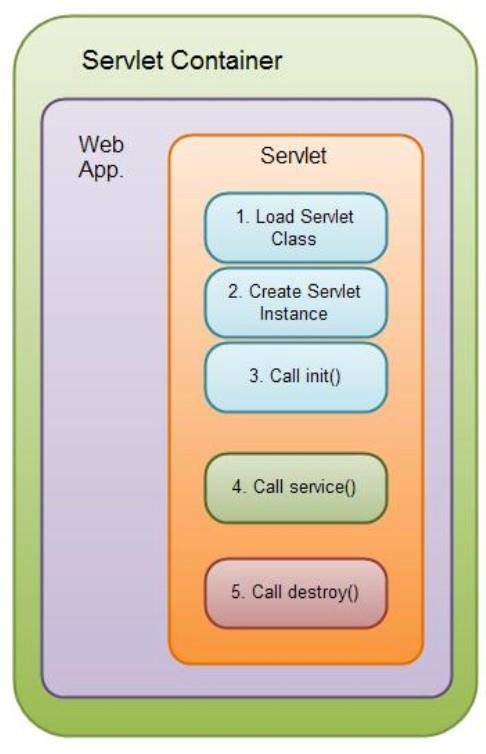
### JavaServlet

* **Servlet**làmộtcôngnghệđượcsửdụngđểtạoracácứngdụngweb (đượctriểnkhaiởphíaserverđểtạoracáctrangwebđộng).
* Công nghệ Servlet rất "khỏe mạnh" và có khả năng mở rộng nhờsử dụngngônngữlậptrìnhjava.
* Servlet làmgì?
* Nhận clientrequest
* Lấy thông tintừ request
* Xửlýnghiệpvụhoặcphátsinhnộidungbằngcáchtruycậpdatabase,triệu gọi EJB...
* Tạovàgửiresponsetớiclienthoặcchuyểnrequesttớimộttrangservlet hoặc JSPkhác. 

### Servlet Container (Bộ chứa cácServlet)

* Servlet container là bộ phận tương tác với servlet để xử lý các yêu cầutừngườidungtớicáctrangwebđộng.
* ServletContainerthựchiệncáctácvụdướiđây:
* Quảnlývòngđời
* Hỗtrợxửlýđaluồng
* ObjectPooling
* Bảomật

### Vòng đời củaServlet

* ServletContainerbảotrìvòngđờicủa một thực thểservlet.
* Gồm 5bước:
* B1:TảiServletClassvàobộnhớ
* B2: Tạo đối tượngServlet
* B3: Gọi phương thứcinit()
* B4: Gọi phương thứcservice()
* B5: Gọi phương thứcdestroy()

### Vòng đời củaServlet

* Bước1,2,3đượcthựcthimộtlầnduynhấtkhimàservletđượcnạp lầnđầu.Mặcđịnhcácservletkhôngđượctảilênchotớikhinónhận một dòi hỏi đầu tiên từ người dùng. Bạn có thể buộc Servlet

Containertảicácservletkhinókhởiđộng.

* Bước 4được thực thi nhiều lần, mỗi khi có đòi hỏi từ phía người dùngtớiservlet.
* Bước5đượcthựcthikhiServletContainertrútbỏServlet.

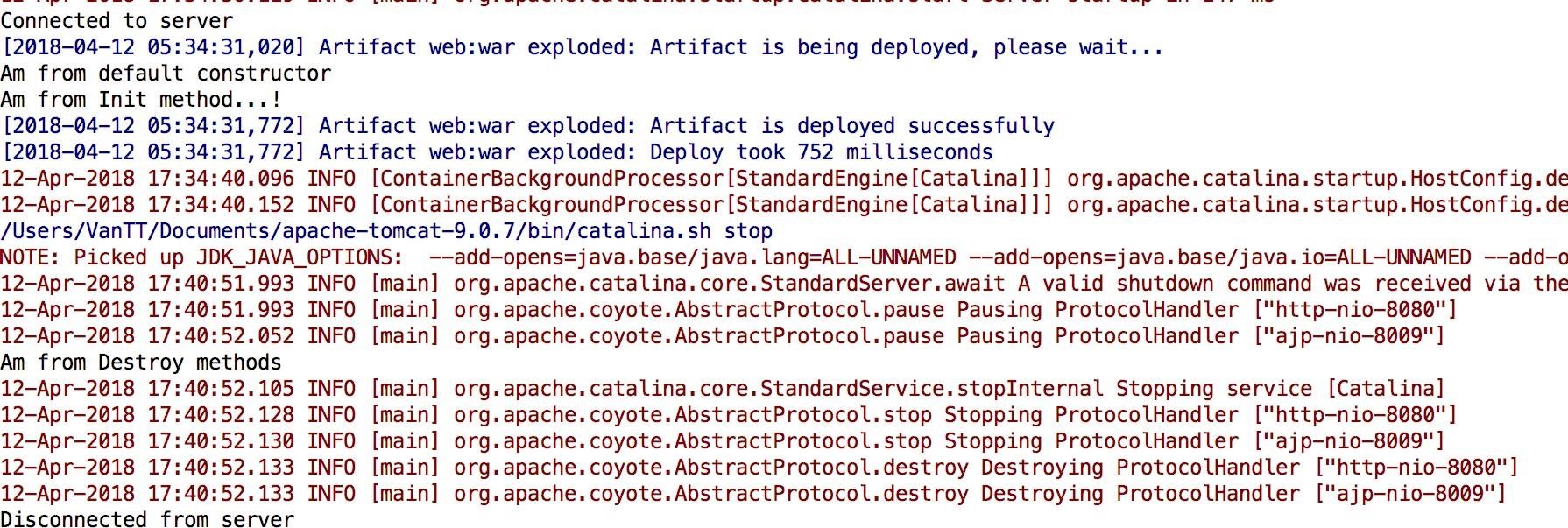
### Ví dụ: Quan sát vòng đời củaServlet

• Xem ví dụ tạiđây: https://github.com/codegym-vn/java-webservlet-life-cycle

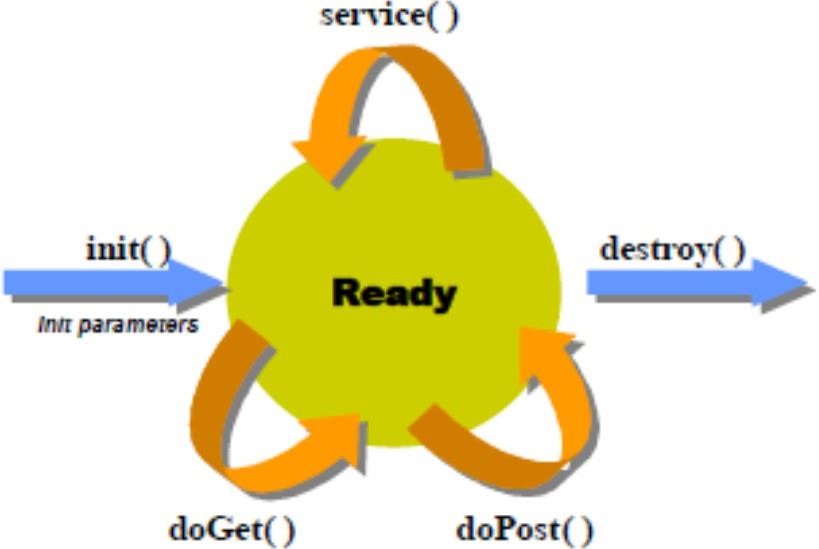


### Ví dụ: Quan sát vòng đời củaServlet

• Kết quả trảvề:



### Các phương thức trong vòng đờiServlet



### Các phương thức trong vòng đờiServlet

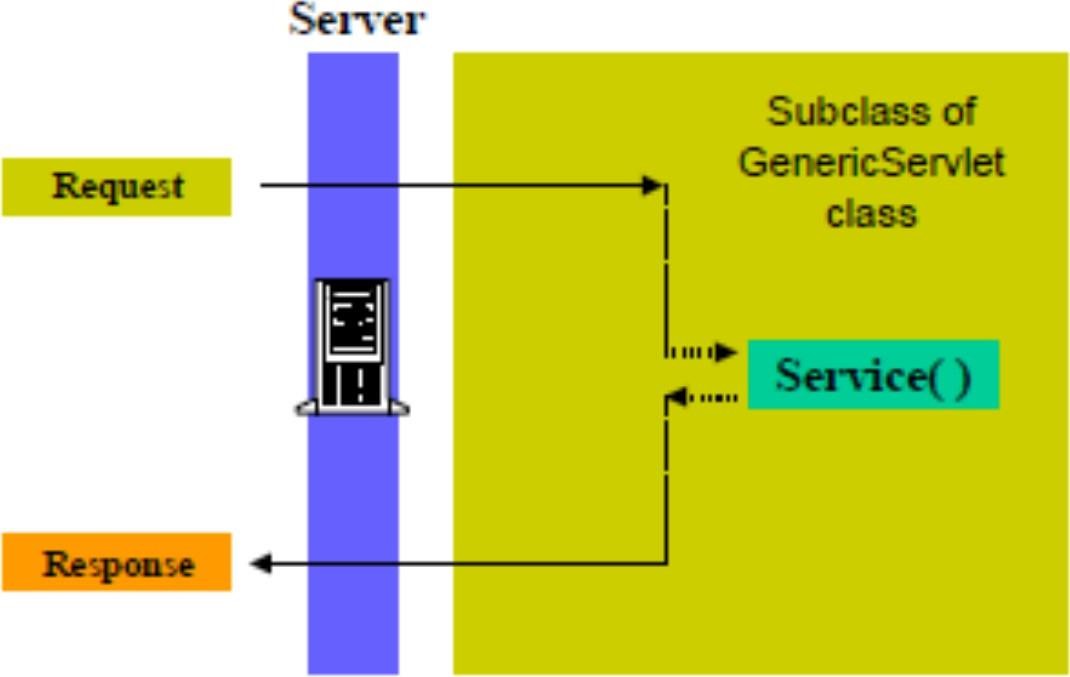
* Phương thứcinit()
* Đượcgọimộtlầnkhiservletđượctạo
* Thựchiệncáckhởi tạotrongphươngthứcnàynhưtạomộtkếtnốitới

CSDL, mởfile...

* Phương thứcdestroy()
* Đượcgọitrướckhihuỷmộtđốitượngservlet
* Thựchiệnthaotácdọndẹpnhưđóngfile,đóngkếtnốiCSDL

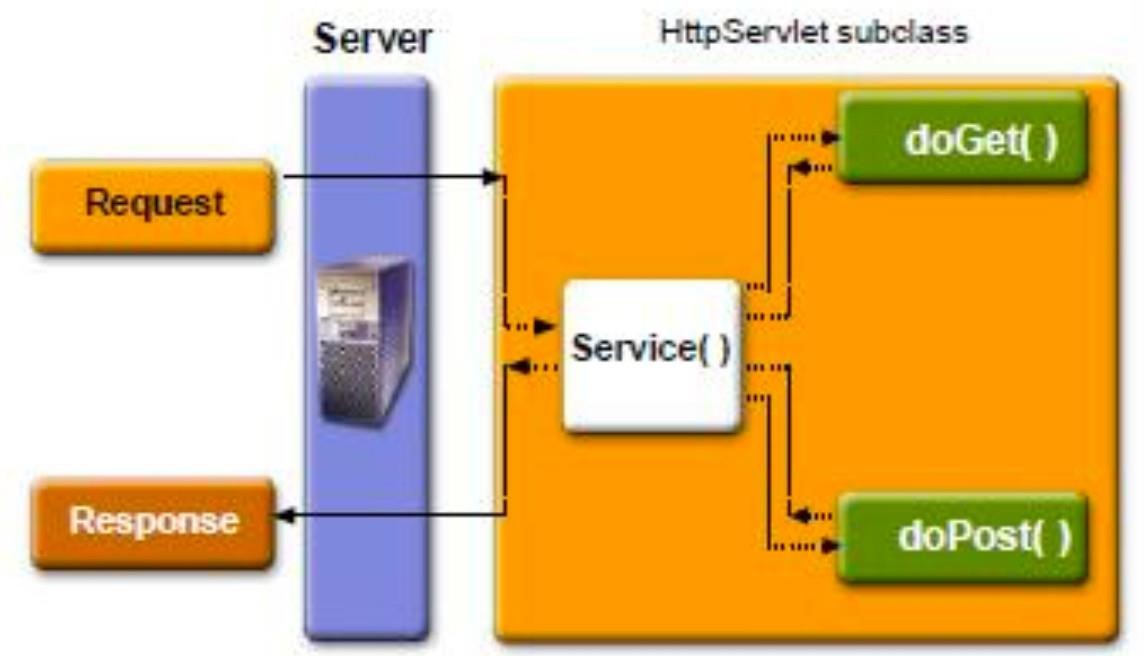
### Các phương thức trong vòng đờiServlet

* Phương thứcservice()
* Đượcgọimỗikhicórequesttừclientđến
* Tuỳvàoloạihttprequestcụthểnógọiđếnmộttrongcácphươngthức doGet() hoặcdoPost().
* Tạicácservlet,chúngtacầnghiđèvàxửlýcácphươngthứcnày.

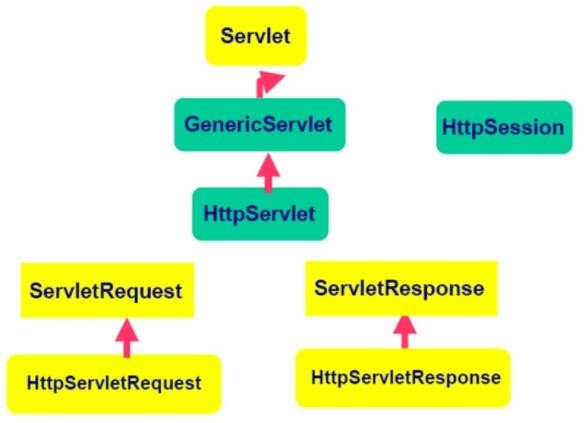


### Các phương thức trong vòng đờiServlet

* Phương thứcdoGet()/doPost()
* Tríchxuấtcácthôngtingửitừclient(HTTPparameter)từHTTPrequest
* Thiếtlập/truycậpcácthuộctínhcủacácscopeobjects
* Thựchiệncácxửlýnghiệpvụ(businesslogic)hoặctruycậpCSDL
* TuỳchọnforwardrequesttớicácWebcomponentskhác(ServlethoặcJSP)
* SinhHTTPresponsevàtrảvềchoclient



### Bộ thư viện Servlet-1

* Trong Servlet có 2 gói quan trọng là javax.servlet và javax.servlet.http.Haigóinàycungcấpcácinterfacevàlớpđểtạora cácServlet.
* InterfaceServletđịnhnghĩacácphươngthứctrongvòngđờicủa mộtServlet
* LớpGenericServletthựcthitừServlet
* Lớp HttpServlet kế thừa GenericServet, cung cấp các phương thức đểxửlýcácphươngthứcHTTPnhưdoGet()xửlýGET,doPost()xửlý post 

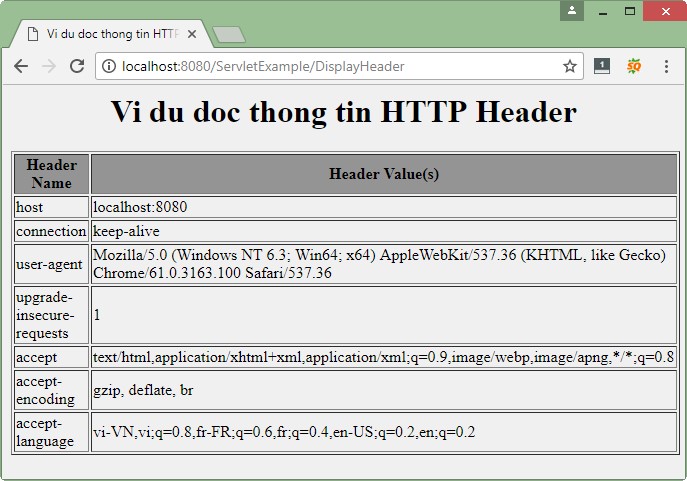
### Bộ thư viện Servlet-2

* ĐểtạomộtServlet,tacầnthựcthiinterfaceServlettrựctiếphoặc giántiếpthôngquaGenericServlethoặcHttpServlet.
* Trongthựctế,lớpServletthườngkếthừaHttpServletđểcóthểxửlý các phương thứcHTTP.

### ServletRequest

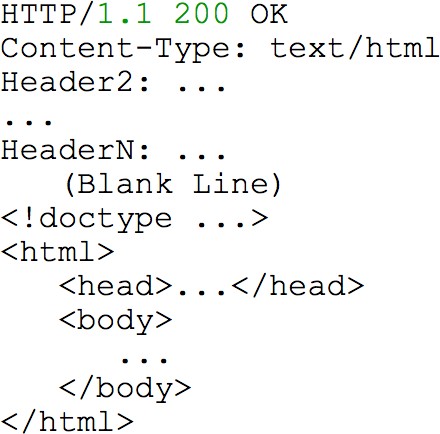
* Khitrìnhduyệtgửiyêucầu(request)đếnmộttrangweb.Nógửirất nhiềuthôngtinđếnwebservernhưngnhữngthôngtinnàykhông thể đọc trực tiếp vì chúng là một phần của Header trong HTTP request.
* CácphươngthứcthuộclớpHttpServletRequestđượcsửdụngđể đọcHTTPHeadertrongchươngtrìnhservlet

### Ví dụ: Đọc thông tin HTTPHeader

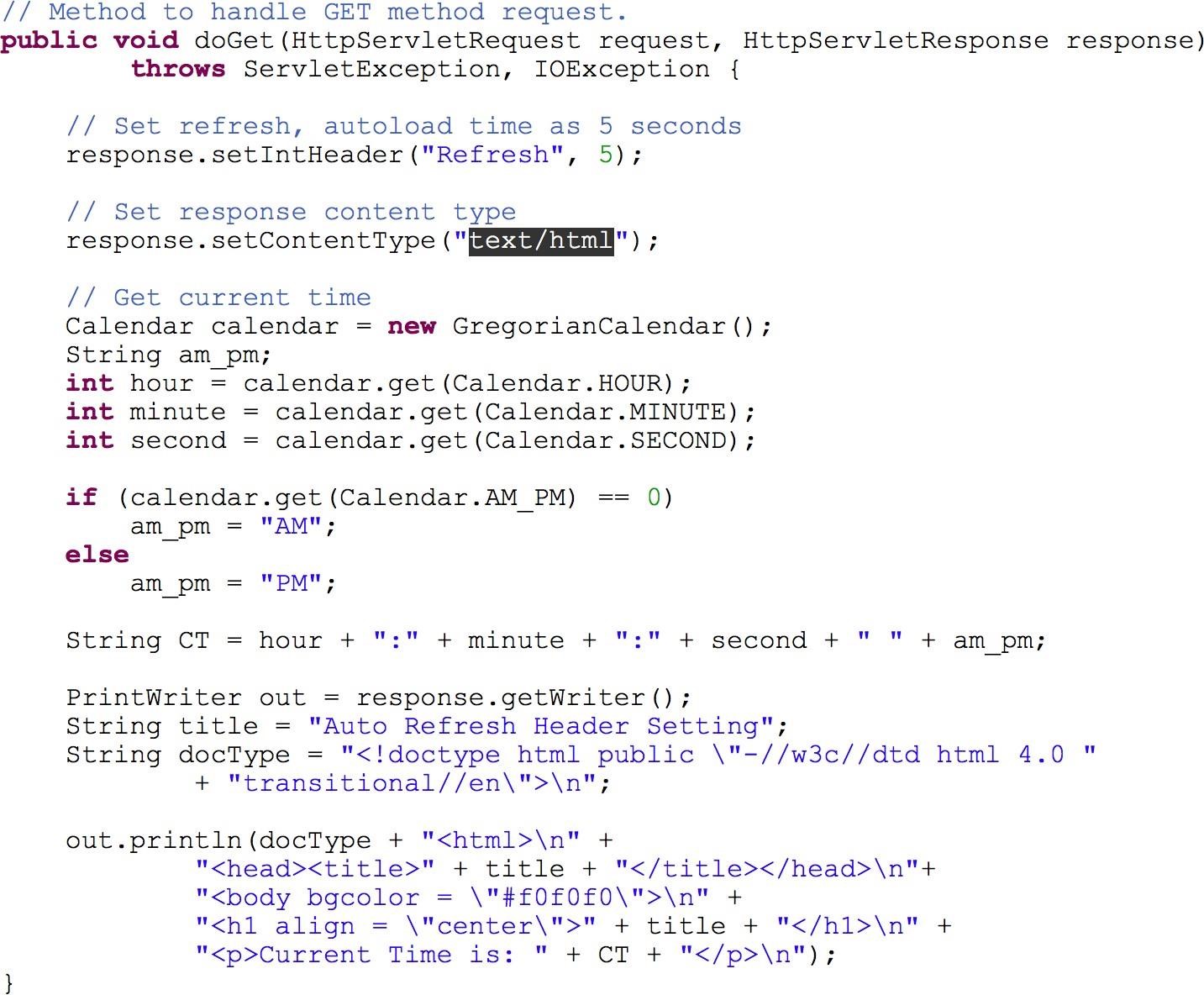


### ServletResponse

• Khiwebserverđápứng(response)yêucầucủaHTTPrequest.Một responsethôngthườngbaogồmtrạngthái(status),header,blank linevàdocument



### Ví dụ: Hiển thị ngày giờ hệthống

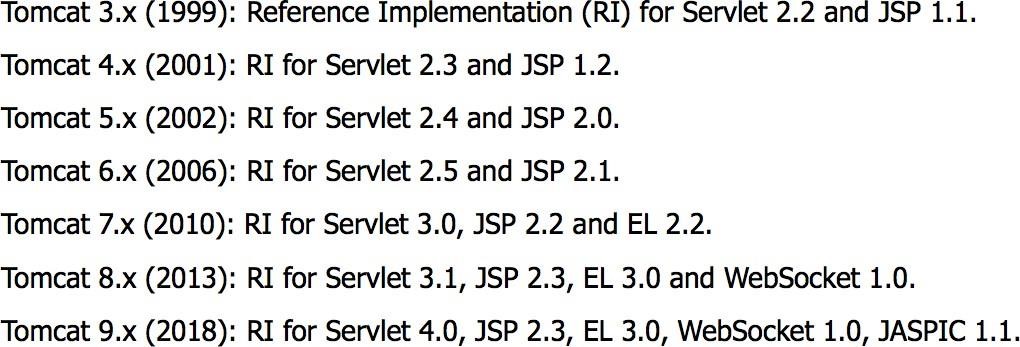


# ApacheTomcat

ApacheTomcat

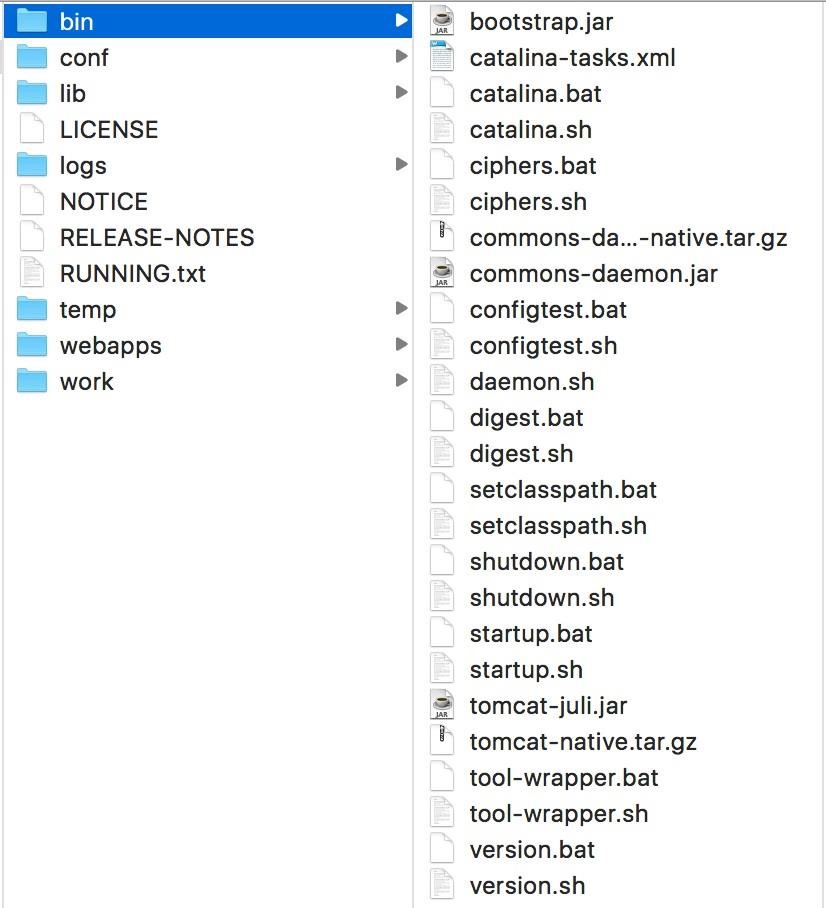
Tạo projecttrên IntellJ

## ApacheTomcat

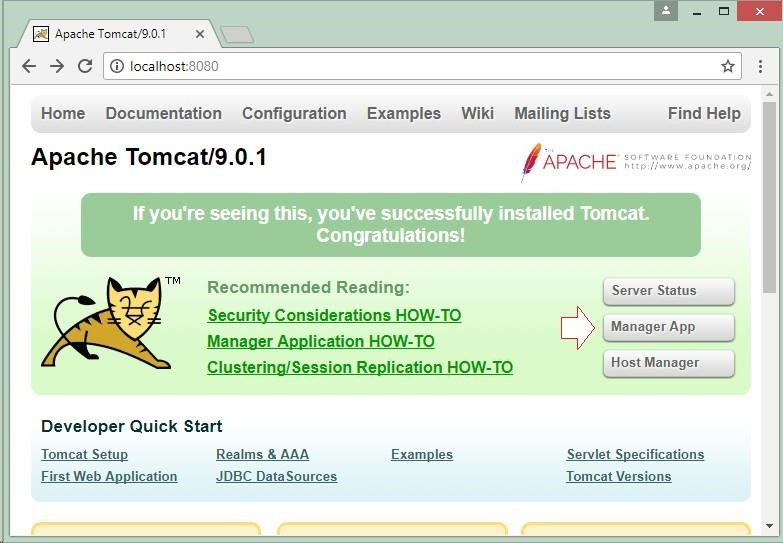
* Apache Tomcat là một ứng dụng chủ (Application Server),là một phầnmềmmãnguồnmởđượccungcấpbởiApache.
* Tomcat thi hành các ứng dụng Java Servlet và JavaServer Pages (JSP) từ Sun Microsystems, và cung cấp một máy chủ HTTP cho ngôn ngữ Java thuần túy để thực thi các chương trình lệnh viết bằngngônngữJava.
* Mộtsốphiênbản:

## Cài đặt Web serverTomcat

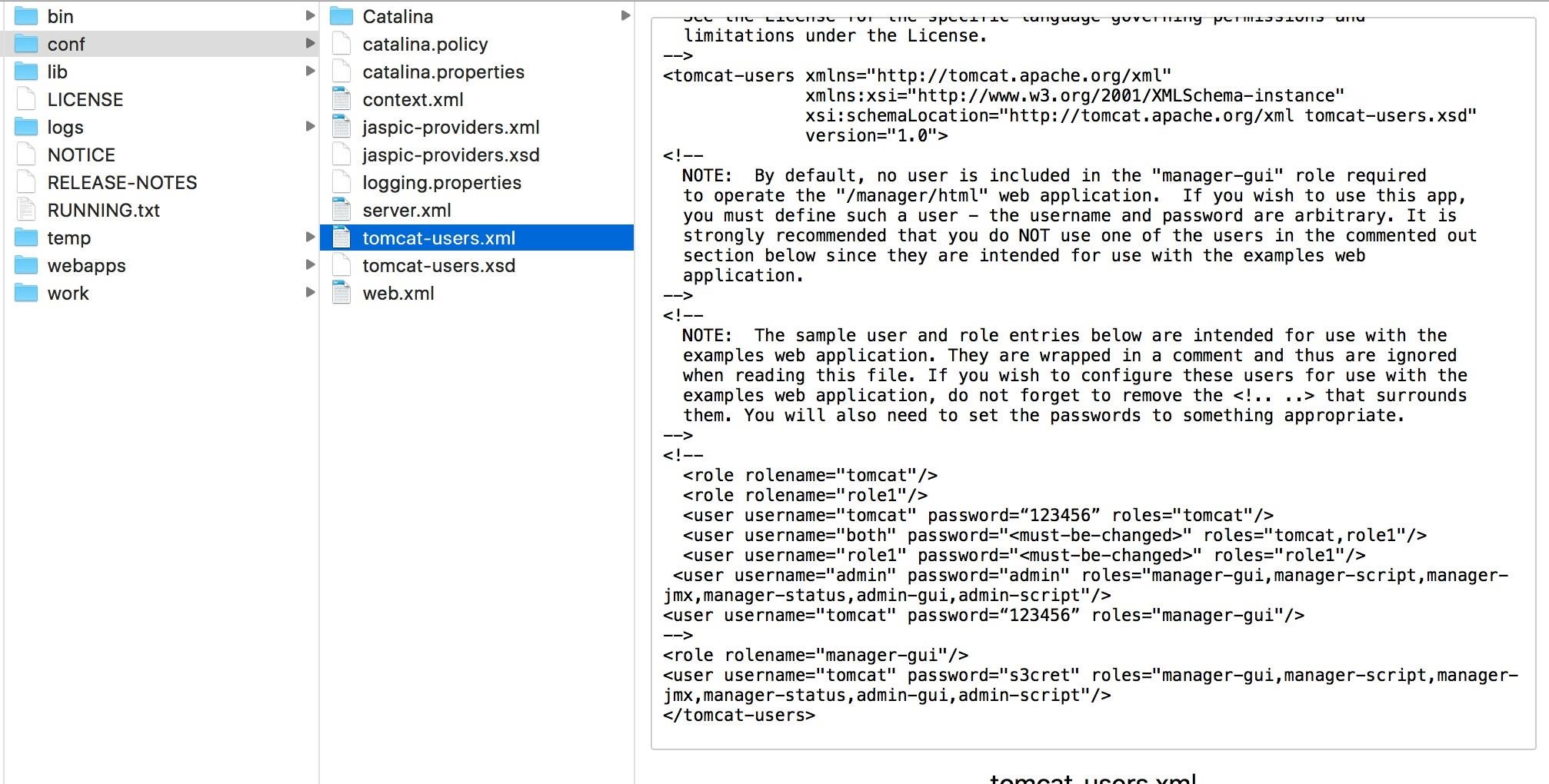
* Tải và cải Tomcat tại:[http://tomcat.apache.org](http://tomcat.apache.org/)
* Cấu trúc thư mụctomcat:



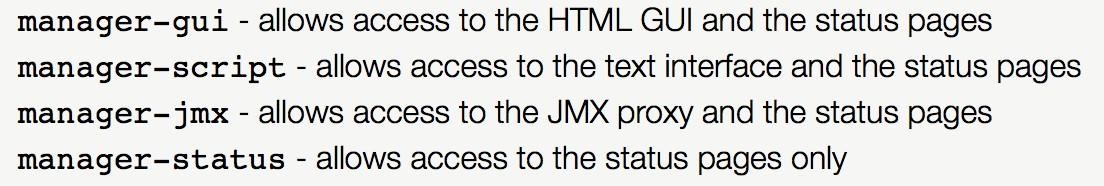
* Start Tomcat Server: Sử dụng startup.bat/ startup.shtrong windown/linux
* Stop Tomcat Server:Sử dụng shutdown.bat/ shutdown.sh trong windown/linux
* Saukhistattomcat,trêntrìnhduyệt,truycậpvàođịachỉ:

http://localhost:8080 

* Cấu hình các **user** được phép sử dụng **Tomcat**. Mở file **tomcatusers.xml**



* **Tomcat**đãđịnhnghĩađịnhnghĩatrước4vaitrò(role)sau:



* Thêm vào filetomcat-users.xml như sau:

**Demo: T*ạ*o project trênIntellJ**

## Tóm tắt bàihọc

* HTTPlàgiaothứctruyềntảisiêuvănbản,sửdụngđểthiếtlậpgiao tiếp giữamáychủ cungcấpdịch vụweb (Server)vàmáykháchsử dụngdịchvụweb(Client)
* Apache Tomcat là một ứng dụng chủ (Application Server),là một phầnmềmmãnguồnmởđượccungcấpbởiApache.
* Servletlàmộtcôngnghệđượcsửdụngđểtạoracácứngdụngweb.
* Các phương thức trong vòng đời của Servlet: init(), service(), destroy(), doGet(),doPost()

# Hướngdẫn

Hướngdẫnlàmbàithựchànhvàbàitập Chuẩnbịbàitiếptheo:JSP & JSTL