

[laboratori] Lab 01: Introducció a tecnologies Web. Java Servlets i HttpClient

Lab 01: Introducció a tecnologies Web. Java Servlets i HttpClient

Introducció

En aquesta primera de sessió de laboratori ens introduïrem a la comunicació client/servidor amb HTTP. Més concretament, en el costat del servidor tindrem un Java Servlet executant-se en un servidor Apache Tomcat. A la part client, tindrem un programa Java que utilitzarà la llibreria Apache HttpClient per fer les peticions HTTP al Servlet esmentat.

Tasques a Realitzar

Prèvia

Se suposa que ja heu entrat als PCs escollint LINUX com a Sistema Operatiu. Les tasques s'han de dur a terme mantenint l'ordre seqüencial amb què són descrites a continuació.

Tasca #1 (20 punts si es completa abans de la fi de la sessió de laboratori)

1.1 Instal·lació d'Apache Tomcat

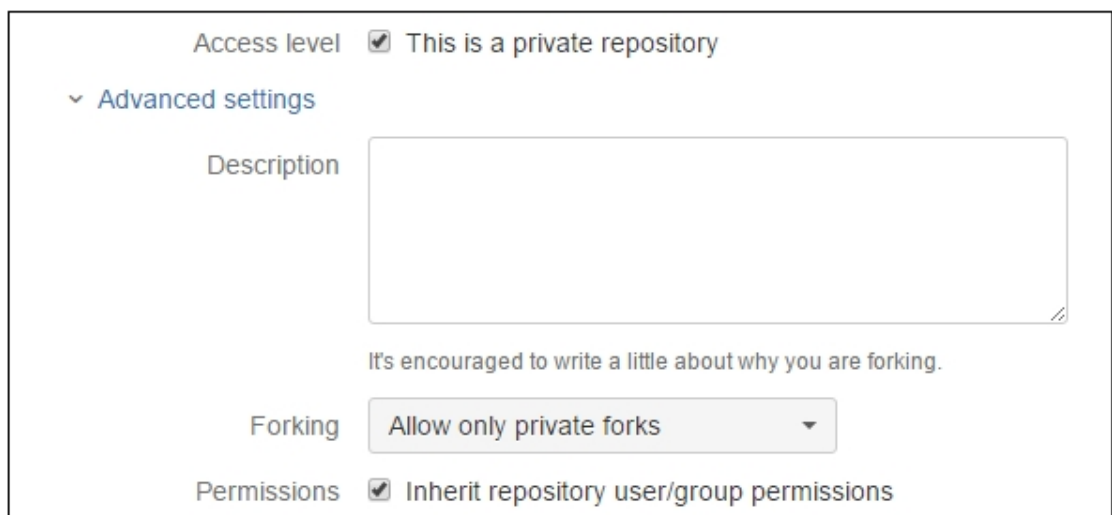
Des del terminal de Linux, executeu la comanda `insttomcat7`. Comproveu que s'ha creat el directori tomcat dins del vostre directori \$HOME.

1.2 Posada en marxa d'Eclipse J2EE i configuració del Servidor Tomcat a Eclipse

Des del terminal de Linux, executeu la comanda `eclipse`. Tanqueu la pestanya "Welcome" d'eclipse per poder veure l'àrea de treball. A la part inferior hi veureu la pestanya "Servers". Aneu-hi i cliqueu "No servers are available. Click this link ...". A la finestra emergent seleccioneu "Apache>Tomcat v7.0 Server" i cliqueu "Next >". En aquesta nova pantalla, feu "Browse..." fins al vostre directori arrel de tomcat (aquell que s'ha creat a l'apartat 1.1) i cliqueu "Finish". Veureu que a la pestanya "Project Explorer" (al *pane* de més a l'esquerra) ara hi teniu una nova carpeta "Servers".

1.3 Fork & Clone del repositori "waslab01"

1. Des del vostre navegador, aneu al repositori https://bitbucket.org/fib_was/waslab01 i feu-ne un "Fork". Un dels dos membres de la parella s'haurà de "loguejar" amb les seves credencials de bitbucket.org. No cal que canvieu el nom del vostre repositori però sí que es necessari que les opcions **"This is a private repository"** (Access Level) i **"Inherit repository user/group permissions"** (Permissions) **estiguin seleccionades** tal com es mostra a la figura:



The image shows a screenshot of the Bitbucket 'Fork' form. At the top, there is a checkbox labeled 'This is a private repository' which is checked. Below this is a section titled 'Advanced settings' with a dropdown arrow. Under 'Advanced settings', there is a 'Description' label followed by a large text input area. Below the input area, there is a note: 'It's encouraged to write a little about why you are forking.' Below this note, there is a 'Forking' label followed by a dropdown menu showing 'Allow only private forks'. At the bottom, there is a 'Permissions' label followed by a checkbox labeled 'Inherit repository user/group permissions' which is checked.

2. A la pàgina del vostre repositori nou, copieu l'URI que hi ha al costat del camp HTTPS
(p.ex: `https://el_vostre_username@bitbucket.org/el_vostre_username/waslab01.git`)

3. A Eclipse, feu "Window> Show View > Other...". A la finestra emergent seleccionau "Git > Git repositories" i feu "OK". Veureu que us surt una nova pestanya "Git repositories" al *pane* inferior. Seleccionau-hi "Clone a Git repository". A la finestra emergent copieu-hi l'URI del vostre repo, poseu-hi el vostre password a bitbucket.org i feu "Next >". A la següent pantalla no toqueu res i feu "Next >". I a l'altra pantalla seleccionau l'opció "Import all existing projects ..." i feu "Finish". Us haureu d'esperar una estoneta i finalment veureu que al *pane* de l'esquerra ("Project Explorer") hi teniu un dos nous projectes: "waslab01_cs" i "waslab01_ss".

1.4 Execució de les aplicacions "waslab01_cs" i "waslab01_ss"

Comencem primer per "waslab01_ss". Aquesta és una aplicació web també anomenada "Wal of Tweets". Per executar-la, només cal que cliqueu amb el botó dret del ratolí a sobre del projecte waslab01_ss i seleccionau Run As>Run on Server. En principi us hauria de sortir com a preseleccionat el "Tomcat v7.0 Server at localhost". Llavors simplement premeu Finish. Veureu que en el *pane* central d'Eclipse s'obre una pestanya "Wall of Tweets" mostrant la pàgina web de l'aplicació. També la podeu veure des del vostre navegador si aneu a http://localhost:8080/waslab01_ss/. Hauríeu de veure quelcom semblant a [capture01.html](#).

El projecte "waslab01_cs" conté un simple classe Java que fa de client de l'aplicació anterior. Per executar-la, cliqueu amb el botó dret a sobre del projecte waslab01_cs i seleccionau "Run As>Java Application". A la consola d'eclipse hauríeu veure quelcom semblat a això: [capture02.txt](#).

1.5 Primer Commit

1. Aneu a la pestanya "Git repositories" i deplegueu el vostre repositori "waslab01 [master]". Dins de "Working Directory" hi ha el fitxer README. Feu-hi doble click per editar-lo i escriure-hi els vostres noms. Llavors salveu-lo.
2. De nou a la pestanya "Git repositories", cliqueu amb el botó dret del ratolí a sobre l'arrel del vostre repositori "waslab01 [master]" i seleccionau "Commit ...". A la finestra emergent seleccionau el fitxer README.md. On diu "Commit message"

escriuiu-hi "**tasca #1 finalitzada**" i premeu el botó "**Commit and Push**" (així també s'actualitzarà el repositori que teniu bitbucket.org)

Tasca #2 (15 punts si es completa abans de la fi de la sessió de laboratori)

Tal com haureu pogut veure, el resultat que obté waslab01_cs, més concretament la seva classe SimpleFluentClient.java, com a client de l'aplicació "Wall of Tweets" (waslab01_ss) és un document en format HTML. L'objectiu d'aquesta tasca serà obtenir un contingut en format "text/plain", amb la mateixa informació però amb un format més adequat per ser mostrat al Console d'Eclipse. Per discriminar el tipus de contingut desitjat utilitzarem un mecanisme propi del protocol HTTP: els HTTP Header fields.

La idea en sí és bastant simple: SimpleFluentClient.java afegirà un Header field a la seva petició HTTP. Més concretament hi afegirà la capçalera "Accept: text/plain", on "Accept" és el nom del camp i "text/plain" n'és el valor. A la banda del servidor (waslab01_ss), la classe WoTServlet.java serà l'encarregada de comprovar la capçalera que li arriba en la petició GET per tal de generar el contingut en el format desitjat.

Per tant, us tocarà de modificar la implementació de la classe waslab01_cs>src>asw01cs>SimpleFluentClient.java i la del mètode doGet de la classe waslab01_ss>Java Resources>src>wallOfTweets>WoTServlet.java. En el primer cas, haureu d'inserir la crida a l'operació [addHeader](#) tot just abans de la crida execute(). En el cas del mètode doGet de WoTServlet.java, haureu de cridar el mètode [getHeader](#) de l'objecte HttpServletRequest req per saber si el client vol o no un "text/plain". En cas afirmatiu, caldrà invocar, i implementar, una operació alternativa, printPLAINresult, que s'encarregui de generar el contingut en el format desitjat.

Un cop fetes aquestes modificacions, el resultat generat per l'execució de SimpleClient.java (Run As>Java Application) hauria de ser semblant a [capture03.txt](#).
Nota 1: El número que apareix al costat de cada tweet (p.ex. el '6' de 'tweet #6') correspon al seu identificador; hi ha una operació, getTwid(), per obtenir-lo. Nota 2:

Les modificacions que feu a la classe WoTServlet.java no haurien de tenir efectes "col·laterals". És a dir, la pàgina web http://localhost:8080/waslab01_ss/ s'hauria de veure al navegador de la mateixa manera que ja es veia a l'apartat 1.5.

Un cop tot funcioni correctament, procediu a fer el **Segon Commit**. Torneu a la pestanya "Git repositories", cliqueu amb el botó dret del ratolí a sobre l'arrel del vostre repositori "waslab01 [master]" i seleccioneu "Commit ...". Seleccioneu els dos fitxers que heu modificat: waslab01_cs/src/asw01cs/SimpleFluentClient.java i waslab01_ss/src/wallOfTweets/WoTServlet.java. On diu "Commit message" escriviu-hi "**tasca #2 finalitzada**" i premeu el botó "Commit and Push".

Tasca #3 (15 punts si es completa abans de la fi de la sessió de laboratori)

Qualsevol usuari que utilitzi l'aplicació "Wall of Tweets" amb el seu navegador hauria de ser capaç de publicar els seus propis tweets omplint el formulari html que a tal efecte hi ha al principi de la pàgina generada per l'aplicació. Malauradament, aquesta funcionalitat encara no està disponible. L'objectiu d'aquesta tasca serà implementar-la.

Bàsicament, allò que haureu de fer és afegir el codi necessari al mètode doPost de la classe waslab01_ss>Java Resources>src>wallOfTweets>WoTServlet.java, sense eliminar la línia de codi que ja hi ha (i que ha de romandre al final). L'operació per insertar un nou tweet a la base de dades ja està implementada: Database.insertTweet(String author, String text). Òbviament per poder-la invocar caldrà que obtingueu abans els paràmetres necessaris (author i text). Aquests paràmetres es corresponen amb els dos camps definits al formulari html ("author" i "tweet_text", respectivament) que són enviats pel navegador un cop l'usuari prem el botó "Tweet!". Per obtenir-los, heu de cridar el mètode [getParameter](#) de l'objecte HttpServletRequest req.

Un cop ja pugueu inserir tweets des del navegador, procediu a fer el **Tercer Commit**. Torneu a la pestanya "Git repositories", cliqueu amb el botó dret del ratolí a sobre l'arrel del vostre repositori "waslab01 [master]" i seleccioneu "Commit ...". Seleccioneu l'únic fitxer que heu modificat: waslab01_ss/src/wallOfTweets/

WoTServlet.java. On diu "Commit message" escriviu-hi **"tasca #3 finalitzada"** i premeu el botó "Commit and Push".

Tasca #4 (15 punts si es completa abans de la fi de la sessió de laboratori)

En aquesta tasca fareu que el vostre client SimpleFluentClient.java pugui demanar la inserció d'un nou tweet tal com ho fa el navegador quan l'usuari omple el formulari html. Per comprovar que el tweet s'afegeix amb èxit, afegiu el troç de codi necessari abans del codi existent que fa la consulta del contingut del "Wall of Tweets".

Per implementar aquest troç de codi nou, inventeu-vos el valors dels paràmetres "author" i "tweet_text" i adapteu el codi que trobareu al punt 4 de la pàgina <http://hc.apache.org/httpcomponents-client-ga/quickstart.html>, tenint en compte que l'URI amb el qual heu d'invocar el mètode POST és "http://localhost:8080/waslab01_ss/wot".

Fetes aquestes modificacions, si executeu ara l'aplicació waslab01_cs veureu que us surt l'error "Exception in thread "main" org.apache.http.client.HttpResponseException..." a la Console d'Eclipse. Tanmateix, si comproveu la pàgina web de l'aplicació "Wall of Tweets" hauríeu de poder veure el tweet que volíeu crear s'ha acabat afegint. El motiu de l'excepció que us surt a la Console és que el mètode doPost de la classe waslab01_ss>JavaResources>src>wallOfTweets>WoTServlet.java fa un "redirect" al final del seu codi (la línia que hi era des del principi) que, de retruc, fa que peti el getResult() final del mètode Post del SimpleFluentClient.java. Per solucionar-ho, tornem a utilitzar l'estratègia d'afegir un Header "Accept" amb valor "text/plain" a la invocació Post de SimpleFluentClient.java. Això implica que caldrà modificar també el mètode doPost de la classe waslab01_ss>Java Resources>src>wallOfTweets>WoTServlet.java perquè no faci el redirect si el Header "Accept" té el valor "text/plain". Així, en comptes de fer el redirect, el que ha de fer ara és retornar l'identificador del tweet generat (l'operació Database.insertTweet(String author, String text) que heu usat a la Tasca #3 ja retorna aquest identificador). D'aquesta manera, SimpleFluentClient.java podrà genera un output semblant a [capture04.txt](#), on el número del principi correspon a l'identificador del tweet acabat

de creat i que coincideix, òbviament, amb el número del primer tweet que apareix a la llista.

Un cop tot funcioni correctament, procediu a fer el **Quart Commit**. Torneu a la pestanya "Git repositories", cliqueu amb el botó dret del ratolí a sobre l'arrel del vostre repositori "waslab01 [master]" i seleccioneu "Commit ...". Seleccioneu els dos fitxers que heu modificat: waslab01_cs/src/asw01cs/SimpleFluentClient.java i waslab01_ss/src/wallOfTweets/WoTServlet.java. On diu "Commit message" escriviu-hi "**tasca #4 finalitzada**" i premeu el botó "Commit and Push".

Tasca #5 (Fins a 35 punts si es completa com a molt tard a les 22:00 del dilluns 29.02.2016)

Ara us podeu relaxar una mica perquè per dur a terme aquesta darrera tasca disposareu de més temps. Com a contrapartida, les instruccions que us donaré no seran tan precises. Es tracta, bàsicament, que l'aplicació web "Wall of Tweets" (waslab01_ss) i més concretament, la classe WoTServlet.java implementi la funcionalitat d'esborrar tweets i que tant el client web (generat pel propi WotServlet a través de l'operació printHTMLresult) com el client Java SimpleFluentClient.java la puguin invocar amb èxit. Aquestes són les restriccions que haureu de satisfer:

- El format de la petició HTTP per demanar l'esborrat d'un determinat tweet és el mateix tant si es fa des del client web (navegador) com des del client Java, de tal manera que la implementació des del punt de vista del servidor (WotServlet.java) serà la mateixa en ambdós casos.
- En el cas del client web, cada tweet mostrat per pantalla tindrà un enllaç o botó que en clicar-lo permetrà fer la petició per esborrar aquell tweet.
- En el cas del client Java, aquest inclourà al final un tros de codi que sol·licitarà l'esborrament del tweet que es crea al principi (tasca #4).
- Només es podran esborrar els tweets que prèviament s'han generat des del mateix client. A la pràctica això vol dir que haureu d'utilitzar cookies. Quan WoTServlet.java creï amb èxit un nou tweet, retornarà al client una cookie que contindrà l'id del tweet creat. Quan WoTServlet.java rebi la petició per esborrar un determinat tweet, inspeccionarà les cookies que li passa el client per comprovar si alguna d'elles inclou l'id del tweet que aquest vol esborrar.

- No us heu d'amoïnar per la gestió de cookies en els clients: tant el navegador com la llibreria que usa el vostre client Java les gestiona automàticament, reenviant al servidor totes les cookies que prèviament aquest li ha enviat abans en les seves respostes a peticions anteriors.
- Utilitzeu alguna funció de hash criptogràfica (MD5, SHA-256, ...) per codificar l'id del tweet dins de la cookie de tal manera que només el servlet que hagi creat aquella cookie sigui l'únic capaç de determinar la validesa de la cookie en el moment d'esborrar el tweet.

Commits. En aquesta darrera tasca no us limiteu a fer un únic "Commit & Push" final, sinó, al contrari, heu de fer "Commit & Push" cada cop que modifiqueu un fitxer, tot i que no estigueu segurs de si aquell canvi funciona o no. Això em permetrà a mi de disposar d'una traçabilitat de com heu anat desenvolupant aquesta tasca.

Lliurament de la Pràctica

Si heu creat el vostre repositori de [bitbucket.org](#) tal com s'indicava a l'apartat 1.4, jo també hi tinc accés. Per tant, el repositori en sí és el vostre "lliurament". Això sí, **aneu comprovant de tant en quant que els vostres commits es propaguen correctament al vostre repositori de [bitbucket.org](#).**

Avaluació

Al principi de cada tasca ja s'especifica quants punts val si es realitza en el termini establert. En el cas de les tasques #1-4, si no s'acaben dins de la sessió de laboratori, es poden finalitzar amb posterioritat però, en aquest cas, **valdran 5 punts** cadascuna.