

[laboratori] Lab 02: Introducció a tecnologies Web. PHP i XML

Lab 02: Introducció a tecnologies Web. PHP i XML

Introducció

En aquesta segona sessió de laboratori continuarem practicant la comunicació client/servidor amb [HTTP](#). En aquest cas, però, en el costat del servidor tindrem un script [PHP](#) executant-se en un servidor Apache. A la part client, tindrem un altre script PHP. A més, ens introduïrem en la utilització d'[XML](#) com a llenguatge d'intercanvi de dades.

Tasques a Realitzar

Prèvia

Se suposa que 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 (15 punts si es completa abans de la fi de la sessió de laboratori)

Instal·lació d'Apache

1. Des del terminal de Linux, instal·leu Apache amb la comanda `install_apache2`
2. Executeu la comanda `dades/apache2/scripts/apache2.init start`
3. Des del vostre navegador, aneu a <http://localhost:8080>. Haurieu de poder veure-hi el missatge "It works FIB!".

Descàrrega i execució de l'aplicació "Wall of Tweets 2"

1. Aneu al repo https://bitbucket.org/fib_was/waslab02 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:



Access level ☒ This is a private repository

▼ Advanced settings

Description

It's encouraged to write a little about why you are forking.

Forking

Permissions ☒ Inherit repository user/group permissions

2. Cliqueu sobre el botó "Clone" del **vostre repositori** (no l'original!) i copieu-ne la comanda git (HTTPS).
3. Des del terminal de Linux, aneu al vostre directori `dades/apache2/htdocs` i executeu la comanda git que heu copiat abans (p.ex: `git clone https://el_vostre_username@bitbucket.org/el_vostre_username/waslab02.git`)
4. Des del vostre navegador, aneu a <http://localhost:8080/waslab02/wall.php>. Hauríeu d'obtenir quelcom semblant a [capture01.html](#).

Primer commit

1. Editeu el fitxer `waslab02/README.md`, poseu-hi els vostres noms i cognoms i salveu-lo.
2. Des del terminal de Linux feu:

```
$ cd waslab02
$ git add README.md
$ git commit -m "tasca #1 finalitzada"
$ git push origin
```

(La darrera instrucció actualitza el vostre repositori a bitbucket.org)

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

Qualsevol usuari que utilitzi l'aplicació "Wall of Tweets 2" 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. Tal com ja ens va passar en l'anterior laboratori, aquesta funcionalitat encara no està disponible. L'objectiu d'aquesta tasca serà implementar-la.

Per dur-ho a terme, haureu d'afegir unes quantes línies de codi en el bloc "case: 'POST'" del vostre fitxer wall.php, a partir de la línia 20 fins a abans del "break". Més concretament, caldrà que obtingueu el valor dels camps del formulari html tot utilitzant la variable PHP [\\$_POST](#), per després cridar l'operació insertTweet (ja implementada) de l'objecte \$dbhandler.

Un cop ja funciona, des del terminal de Linux feu el **segon commit**:

```
$ git add wall.php
$ git commit -m "tasca #2 finalitzada"
$ git push origin
```

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

Fins ara hem utilitzat el navegador com a client de l'aplicació "Wall of Tweets 2". Ara utilitzarem un simple script PHP com a client: client.php. Per executar-lo només caldrà utilitzar la següent comanda des del terminal Linux: php client.php

El resultat que heu de veure al vostre terminal de Linux ha de ser quelcom semblant a [capture02.txt](#). Com podeu comprovar, el resultat obtingut, els tweets publicats al "Wall of Tweets 2", està en format XML i costa una mica de llegir-lo.

Per tant, i per a completar aquesta tasca, heu de modificar el vostre script `client.php` per tal que generi una sortida semblant a [capture03.txt](#). Utilitzeu [SimpleXML](#) ([aquí teniu alguns exemples d'ús](#)) per extreure del document XML retornat per `wall.php` les dades que cal mostrar.

Un cop ja funciona, des del terminal de Linux feu el **tercer commit**:

```
$ git add client.php
$ git commit -m "tasca #3 finalitzada"
$ git push origin
```

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

Ara volem que `client.php` també pugui demanar la inserció d'un nou tweet a `wall.php`. Per tal diferenciar-les de les peticions d'inserció del navegador, `client.php` utilitzarà el mètode HTTP PUT per fer les seves. Caldrà modificar els scripts `client.php` i `wall.php`.

client.php:

Afegiu el codi per fer la petició PUT a `wall.php` (la `$URI` serà la mateixa) a continuació del que ja teniu implementat. Per tal d'utilitzar la funció `file_get_contents` per fer la petició serà necessari proporcionar arguments extra. Per fer-ho, adapteu el codi de l'Example #1 que trobareu a <http://es.php.net/manual/en/context.http.php> amb les següents modificacions:

1. la variable `$posdata` ha de ser un string que contindrà un document XML amb un format semblant a:

```
<?xml version="1.0"?>
<tweet><author>Test Author</author><text>Test Text</text></tweet>
```

Per obtenir aquest string, utilitzeu també SimpleXML, de manera semblant de com ho fa `wall.php` per generar l'xml que retornava a l'apartat anterior (linies 43-57 de https://bitbucket.org/fib_was/waslab02/src/9dc30dcb313c557954a9bba77620aad90f7db3e9/wall.php?fileviewer=file-view-default#wall.php-43:57). Per simplificar, feu que els valors dels elements `<author>` i `<text>` estiguin *hardcoded*.

2. 'method' ha de ser 'PUT'
3. 'header' ha de ser 'Content-type: text/xml'

Després de cridar la funció `file_get_contents`, feu un "echo" del resultat que retorna `wall.php`

wall.php:

Afegiu el codi necessari en el bloc "case:'PUT'", abans de l'"exit". Primer, caldrà obtenir el document XML amb les dades enviades pel `client.php`. Useu la instrucció `file_get_contents('php://input')`. Amb el resultat que aquesta us retornarà, creeu un nou `SimpleXMLElement` per tal d'obtenir-ne els paràmetres necessaris (author i text). Llavors ja podeu cridar l'operació `insertTweet` tal com heu fet a la Tasca #2. Finalment, per tal de generar una resposta per a `client.php` com a "comprovant" de què la inserció s'ha fet correctament, feu un "echo" de l'string que contindrà un document XML amb un format semblant a:

```
<?xml version="1.0"?>
<response><tweetid author="Test Author">13</tweetid></response>
```

on 13 és l'id del tweet tot just creat (el retorna `insertTweet`). Igual que abans, useu `SimpleXML` per generar aquest XML.

Per acabar, des del terminal de Linux feu el **quart commit**:

```
$ git add *.php
$ git commit -m "tasca #4 finalitzada"
$ git push origin
```

Tasca #5 (15 punts si es completa abans de la fi de la sessió de laboratori)

Ara volem que `client.php` pugi demanar l'esborrat d'un tweet determinat donant el seu id. Novament, caldrà modificar els scripts `client.php` i `wall.php` tal com s'explica a continuació:

client.php:

Afegiu el codi per fer la petició HTTP DELETE a wall.php a continuació de tot el codi que ja teniu implementat. Tal com heu fet en la tasca prèvia, haureu de proporcionar arguments extra a la funció `file_get_contents`. Aquest cop, però, només caldrà que us assegureu que `'method'=>'DELETE'`. El camp `'content'` no és necessari perquè les informació que cal enviar, l'id del tweet que es vol esborrar, estarà en la query string de l'URI que utilitzareu per fer la petició (p.ex: `http://localhost:8080/waslab02/wall.php?twid=13`). Després d'invocar `file_get_contents`, feu un `"echo"` per tal de veure quin resultat us retorna wall.php.

wall.php:

Afegiu el codi necessari en el bloc `"case:'DELETE'"`, abans de l'`"exit"`. Primer, caldrà obtenir l'id del tweet que client.php vol esborrar, tot utilitzant la variable PHP [\\$_GET](#). Un cop teniu l'id, crideu l'operació `deleteTweet`, ja implementada, de l'objecte `$dbhandler`. Aquesta operació retorna 1 si l'esborrat del tweet té èxit, o 0, en cas contrari. Finalment, per tal de generar una resposta per a client.php perquè sàpiga com s'ha resolt la seva petició, feu un `"echo"` de l'string que contindrà un document XML amb un format semblant a:

```
<?xml version="1.0"?>
<response><deletion tweetid="13">result</deletion></response>
```

on `result := success|fail`. Com a la tasca anterior, useu SimpleXML per generar també aquest XML.

Per acabar, des del terminal de Linux feu el **cinquè commit**:

```
$ git add *.php
$ git commit -m "tasca #5 finalitzada"
$ git push origin
```

Tasca #6 (Fins a 25 punts si es completa com a molt tard a les 22:00 del dilluns 07.03.2016)

Finalment, es tracta d'escriure el codi PHP necessari a la classe rss.php perquè aquesta generi un RSS (2.0) feed a partir dels tweets de l'aplicació "Wall of Tweets 2". El RSS 2.0 és un format de dades basat en XML. Trobareu més informació a: http://www.w3schools.com/xml/xml_rss.asp. Aquestes són les restriccions que haureu de satisfer:

- Utilitzeu SimpleXML per generar el contingut RSS 2.0 a partir dels tweets de la base dades, de manera similar a com wall.php genera el l'xml amb el contingut dels tweets que retorna a client.php (linies 43-57 de https://bitbucket.org/fib_was/waslab02/src/9dc30dcb313c557954a9bba77620aad90f7db3e9/wall.php?fileviewer=file-view-default#wall.php-43:57)
- Els elements <channel> i <item> han de contenir, com a mínim, els (sub)elements obligatoris del format RSS 2.0.
- L'element <link> de cada <item> ha d'apuntar a la vostra aplicació "Wall of Tweets 2" (wall.php), posicionant la pàgina allà on es troba el tweet corresponent.

El contingut generat per rss.php hauria de poder ser consumit sense problemes per qualsevol aplicació lectora/agregadora de feeds. Sense anar més lluny, el navegador Firefox també és capaç de llegir el format RSS 2.0 i mostrar-lo a l'usuari d'una manera més o menys amigable. Així, per exemple, en el Firefox de Linux s'hauria de poder veure quelcom semblant a [capture04.jpg](#).

Commits & pushes: Feu-ne de manera freqüent.

Lliurament de la Pràctica

Si heu creat el vostre repositori de bitbucket.org tal com s'indicava a la Tasca #1, 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 #1-5, si no s'acaben dins de la sessió de laboratori, es poden finalitzar amb posterioritat però, en aquest cas, **valdran 5 punts** cadascuna.