KTH

DM2517

XML For Publishing

Project

Author:
Nikhil Thakrar 910817
Aref Mohammedi 881012

15 januari 2016

Innehåll

1 Introduktion **2**

2 Beskrivning **3**
2.1 Databas och DTD 3
2.2 XSL transformationer 4
2.3 Sidor 4
2.3.1 Registrering 4
2.3.2 Startsida 5
2.3.3 Användarsida 6

3 Diskussion **7**
3.1 Diskussion 7

A Appendix **8**
A.1 Databastabeller 8
A.2 DTD 9

KAPITEL 1

Introduktion

Projektet som vi har utvecklat i kursen DM2517 XML för publicering är en internettjänst som visar olika TV serier (569 stycken) och som man som en registrerad användare kan följa. På hemsidan kan man söka efter valfri serie, få information om serien som t.ex. handling, genre, språk, nomineringar, hantera sin egen kollektion av TV-serier, se vilka de senaste avsnitten är och när de visades, exportera till Excel vilket gör att man kan kolla igenom sin kollektion på ett excelark etc.

All kod kan hittas via https://github.com/nthakrar/DM2517-Project

KAPITEL 2

Beskrivning

Vår hemsida är en XML-baserad internettjänst som hämtar information om serier och användare från MySQL databastabeller med hjälp av PHP. Informationen sparas i XML format och med hjälp av olika XSL stylesheets omvandlas XML till XHTML för webbläsare och för Excel och Open Office Calc. I följande sektion beskrivs de databastabeller som används för tjänsten, var vi hämtat data ifrån och vilken struktur/standard som vi har definierat för XML.

2.1 Databas och DTD

Vi använder oss totalt 4 databastabeller:

- series
- series_id
- users
- subsribes

All information om TV-serier har vi hämtat från en gratis online databas: OMDb API som tillhandahåller ett gratis API (http://www.omdbapi.com). Genom att skicka med olika GET parametrar till en URL så får man tillbaka informationen i antingen JSON eller XML format. Nedan har vi inkluderat ett exempel med parametrarna t för title och r för format.

```
http://www.omdbapi.com/?t=glee&r=xml
```

Vilket ger tillbaka resultat i följande format:

```
<?xml version="1.0" encoding="UTF-8"?><root response="True"><
    movie title="Glee" year="2009-2015" rated="TV-PG" released
    ="19 May 2009" runtime="44 min" genre="Comedy, Drama, Music"
    director="N/A" writer="Ian Brennan, Brad Falchuk, Ryan Murphy
    " actors="Chris Colfer, Jane Lynch, Kevin McHale, Lea Michele
    " plot="A group of ambitious misfits try to escape the harsh
    realities of high school by joining a glee club, where they
    find strength, acceptance and, ultimately, their voice, while
     working to pursue dreams of their own." language="English"
    country="USA" awards="Won 4 Golden Globes. Another 68 wins &
    176 nominations." poster="http://ia.media-imdb.com/images/M/
    MV5BMTU0OTA5MjA0MF5BMl5BanBnXkFtZTcwNzY0NTY0OA@@._V1_SX300.
    jpg" metascore="N/A" imdbRating="6.7" imdbVotes="114,316"
    imdbID="tt1327801" type="series"/></root>
```

Det finns totalt 569 serier i databasen. Namnen för dessa hämtade vi från sidan https://showrss.info/?cs=feeds. Till varje serie ges det ett ID på showrss för en RSS feed vilket vi har använt oss av i projektet för att kunna visa för användarna vilka de senaste avsnitten är och när de visades. Vi sparade par av namn och ID för serierna i en fil som vi läser ifrån i php funktionen:

```
function query_omdb($mysqli)
```

Varje XML respons som vi får från OMDb API parsar vi och lägger till i databastabellen 'series'. I tabellen series_id sparar vi par av namn och ID.

Eftersom vår hemsida ska kunna ha användare har vi en users tabell där vi sparar användarnamn, lösenord (hashad), salt, email, förnamn och efternamn.

I grova drag har vi organiserat XML så att vi har en <root> tag och inne i <root> har vi definierat <usercatalog> och <seriescatalog>. I <usercatalog> kan det finnas högst en <user>, dvs en inloggad användare eller ingen inloggad alls och i series kan det finnas 0 eller många <series>. Vi inkluder create statements för alla databastabeller inklusive DTD i Appendix.

2.2 XSL transformationer

All information som hämtas från databasen sparar vi i XML. All information om en användare sparas i <user> och all information om serierna <series>. Med XSL transformerar vi XML till XHTML för websidan och till Excel. Alla XSL filer finns i xsl katalogen.

2.3 Sidor

2.3.1 Registrering

Är man nykommen till hemsidan så kan man registrera sig. Det finns 4 obligatoriska fält: username, password, confirm password och email som man måste fylla i. I JavaScripts funktionen:

```
function regformhash(form, uid, email, password, conf)
```

säkerställer vi att alla fälten är ifyllda, att password och confirm password är identiska, att man väljer ett lösenord som är minst 6 tecken lång etc. Vi bestämde oss för att inte skicka iväg lösenord i klartext till server och att därför hasha det av användaren givna lösenordet med sha512. På serversidan i register_inc.php valideras email adressen. Om den inte är giltig så skrivs ett error meddelande tillbaka till klienten. Om det redan finns ett identiskt username i users tabellen så gör vi användaren medveten om det med ett felmeddelande. Vid en lyckad registrering läggs det till en post i users tabellen med kolumnerna user_id, password, salt, email, firstname och lastname. Saltet genereras med hash funktionen i php och password är resultatet av hashningen av saltet tillsammans med lösenordet användaren angett. Slutligen skickas användaren tillbaka till startsidan för att slutföra sin inloggning.

2.3.2 Startsida

Startsidan (och alla andra sidor för den delen) är designad så att det finns en user kolumn till vänster där man kan logga in/registrera sig eller välja mellan olika menyval om man är inloggad. Resten av sidan är ägnad åt TV-serier. De serier som visas här har alla utgivningsår 2015 och det är detta som vi kallar för 'What's trending'. Varje serie presenteras med titel, poster och utgivningsår. Vill man se alla serier oavsett utgivningsår kan man göra det genom att trycka på 'View all series'.

Man kan söka på de serier som finns sparade i databasen vilket vi har implementerat med hjälp av JQuery-UI autocomplete. Sökningen sker med hjälp av ett Ajax request som skickas till servern.

Trycker man på en godtycklig serie visas all information inklusive handling, skådespelare, språk, nomineringar etc.

Log in

I php funktionen

```
login($user_id, $password, $mysqli)
```

söker vi efter det angivna användarnamnet och hämtar ut det sparade lösenordet och saltet som genererades vid registreringen. Lösenordet som användaren angivit hashar vi tillsammans med saltet och om resultatet är identiskt med lösenordet sparat i databastabellen så har användaren lyckat autentiera sig. Vi använder oss av sessioner så att en användare förblir inloggad på alla sidor när han/hon navigerar hemsidan. Sessionsvariablerna user_id och login_string definieras och dessa används varje gång vi vill kolla om en användare är inloggad i funktionen

```
function login_check($mysqli)
```

Startsidan ser annorlunda ut beroende på om användaren är inloggad eller inte. Detta testar vi genom att lägga till ett attribut i <user> status som kan ha värdet logged_in eller logged_off. När vi matchar taggar i XSL så kollar vi efter status och lägger till passade HTML.

När man trycker på subscribe så skickas ett ajax request till servern. På serversidan verifieras att användaren är inloggad och att användaren inte redan följer den serien och en post läggs till i subsribes tabellen. Den här serien kommer nu visas på användarens privata sida.

2.3.3 Användarsida

Hit kommer man vid en lyckad inloggning och här visas ens kollektion av Tv-serier. I menyn kan man ta sig till 'What's trending', uppdatera sin profil genom att ändra på sitt förnamn, efternamn och/eller email adress. Man kan exportera till Excel vilket gör det möjligt att ladda ner XML i ett format som kan kännas igen av Excel/Open Office Calc. Man kan kolla Kalender där de senaste avsnitten och deras publiseringsdatum visas för alla serier man följer. Detta har vi fixat genom att parsa RSS feeds för var och en av serierna via showrss som nämnts tidigare. Vi byggde URL för RSS feeds med hjälp av att bygga URL på följande format `http://showrss.info/feeds/[ID].rss` där ID hämtas från series_id tabellen. Slutligen så kan man logga ut vilket medför att alla sessionvariabler töms och sessionen förstörs.

Det finns en speciell användare: admin. Admin ser alla serier som finns sparade i databasen och kan ta bort valfri serie alternativt alla serier. Admin ser också vilka användare som är registrerade och kan ta bort valfri/alla användare. Vidare kan admin exportera till Excel vilket fungerar på samma sätt som beskrivet ovan bara att XML filen är betydligt större.

KAPITEL 3

Diskussion

3.1 Diskussion

Det har varit väldigt lärorikt att jobba med det här projektet och hela kursen i sig har varit väldigt lärorik. Vi förstår mycket bättre nu hur kraftfullt XML är som ett verktyg för att beskriva/förmedla data på ett standardmässigt sätt och hur det kan användas ihop med andra programmeringsspråk för att underlätta kommunikation. Med hjälp av XSL tranformationer blir det möjligt att transformera XML till olika publiceringskanaler som t.ex. XHTML för webbsidor eller till Word/Excel (även PDF med XSL-FO). Vi använde oss av RSS feeds för att få reda på vilka de senaste avsnitten är för en viss serie och fick därmed en hel del övning med hur man kan parsa XML med PHP och det blev ännu tydligare hur det kan underlätta att ha en standard som XML som i fallet med RSS feeds för att bygga applikationer.

Det finns många fördelar med att använda XML för internettjänster för flerkanalspublicering. I vårt fall med vårt projekt var det kanske lite för krystat. Vi har egentligen ingen riktig användning av flerkanalspublicering så som vi har gjort det med Excel. Det är svårt att se vad en användare skulle göra med ett Excel dokument med information om alla serier han/hon följer. I fallet med admin skulle, å andra sidan, ett Excel dokument kunna användas i ett administrativt syfte för statistik eller något liknande.

En nackdel vi märkte var att det blev svårare/mindre flexibelt att designa HTML sidorna för tjänsten med XSL då vi var tvugna att matcha mot de taggar vi definierat för XML. Detta gjorde det svårare att placera HTML element var som helst i dokumentet. Vi kunde heller inte använda PHP script direkt i den html-genererade filen vilket ledde till att vi fick ta några omvägar med t.ex. JavaScript o.dy.

En nackdel med hur vi har byggt tjänsten är att databasen inte uppdateras dynamiskt med nya Tv-serier utan måste ske med manuellt arbete. Man måste först hämta namn och ID (för RSS feeds) för serier från showrss, göra querys till OMDb API:et, parsa responsen och lägga till i databasen. Utöver detta måste vi själva ladda ner alla posters då URLerna som vi erhåller från OMDb är från imdb och som inte tillåter hotlinking.

BILAGA A

Appendix

A.1 Databastabeller

```
CREATE TABLE `series` (
  `title` varchar(255) character set latin1 NOT NULL,
  `year` varchar(50) default NULL,
  `rated` varchar(50) character set latin1 NOT NULL,
  `released` varchar(50) character set latin1 NOT NULL,
  `runtime` varchar(50) character set latin1 NOT NULL,
  `genre` varchar(50) character set latin1 NOT NULL,
  `director` varchar(255) character set latin1 NOT NULL,
  `actors` text character set latin1 NOT NULL,
  `plot` text character set latin1 NOT NULL,
  `language` varchar(50) character set latin1 NOT NULL,
  `country` varchar(50) character set latin1 NOT NULL,
  `awards` varchar(50) character set latin1 NOT NULL,
  `poster` varchar(255) character set latin1 NOT NULL,
 PRIMARY KEY  (`title`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8

CREATE TABLE `series\_id` (
  `id` varchar(50) NOT NULL,
  `name` varchar(255) NOT NULL,
 PRIMARY KEY  (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1


CREATE TABLE `users` (
  `user\_id` varchar(40) NOT NULL,
```

```
  `password` varchar(128) NOT NULL,
  `salt` varchar(128) NOT NULL,
  `email` varchar(100) NOT NULL,
  `firstname` varchar(30) NOT NULL,
  `lastname` varchar(30) NOT NULL,
 PRIMARY KEY  (`user\_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW\_FORMAT=DYNAMIC


CREATE TABLE `subsribes` (
  `user\_id` varchar(40) NOT NULL,
  `title` varchar(255) NOT NULL,
 KEY `user\_id` (`user\_id`,`title`),
 KEY `title` (`title`),
 CONSTRAINT `subsribes\_ibfk\_1` FOREIGN KEY (`user_id`)
    REFERENCES `users` (`user_id`) ON DELETE CASCADE ON UPDATE
    CASCADE,
 CONSTRAINT `subsribes_ibfk_2` FOREIGN KEY (`title`) REFERENCES
    `series` (`title`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

A.2 DTD

```
<!--Elements -->
<!ELEMENT root(usercatalog, seriescatalog)>
<!ELEMENT usercatalog (user?) >
<!ELEMENT user (user_id, firstname, lastname, email) >
<!ELEMENT seriescatalog (series*)>
<!ELEMENT series (title, year, rated, released, runtime, genre,
   director, actors, plot, language, country, poster)>
<!ELEMENT user_id (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT rated (#PCDATA)>
<!ELEMENT released (#PCDATA)>
<!ELEMENT runtime (#PCDATA)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT director (#PCDATA)>
<!ELEMENT actors (#PCDATA)>
<!ELEMENT plot (#PCDATA)>
<!ELEMENT language (#PCDATA)>
<!ELEMENT country (#PCDATA)>
```

```
<!ELEMENT poster (#PCDATA)>

<!-- Attributes -->
<!ATTLIST user status (logged_in|logged_off) #required>
```