

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N**

Davor Stojanović

**YAHOO QUERY LANGUAGE KAO JEZIK ZA
PRETRAŽIVANJE DRUŠTVENOG WEBA**

ZAVRŠNI RAD

Varaždin, 2011.

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N**

Davor Stojanović

Redoviti student

Broj indeksa: 37050/08-R.

Smjer: Informacijski sustavi

Preddiplomski studij

**YAHOO QUERY LANGUAGE KAO JEZIK ZA
PRETRAŽIVANJE DRUŠTVENOG WEBA**

ZAVRŠNI RAD

Mentor:

Dr. sc. Markus Schatten, viši asistent

Varaždin, kolovoz 2011.

Sadržaj

1. Uvod	1
2. Što je YQL?	2
3. Sintaksa	3
3.1. SELECT	3
3.2. Ažuriranje podataka	4
3.2.1. INSERT	5
3.2.2. UPDATE	6
3.2.3. DELETE	7
3.3. SHOW	7
3.4. DESC	8
4. Filtriranje	10
4.1. Lokalno filtriranje	10
4.2. Udaljeno filtriranje	12
4.3. Kombinirano filtriranje	13
4.4. Limitiranje	14
4.4.1. Lokalno limitiranje	15
4.4.2. Udaljeno limitiranje	16
5. Pokretanje YQL naredbi	17
5.1. YQL konzola	17
5.2. GET metoda	18
6. Autorizacija	21
6.1. Two-legged OAuth	21
6.2. Three-legged OAuth	22
7. Pretraživanje HTML sadržaja	24
8. Aplikacija	27
8.1. Sučelje	27
8.2. Kod	30
9. Kritički osvrt	33
10. Zaključak	34
Literatura	35

1. Uvod

Cilj ovog rada je prikazati sintaksu i semantiku Yahoo! Query Language jezika (u nastavku teksta: YQL), koji omogućava pretraživanje Interneta putem upita sličnih SQL¹ naredbama. Temu sam odabrao upravo zbog mogućnosti samog YQL jezika; želio sam naučiti način dohvaćanja podataka iz društvenih mreža koje sadrže ogromne količine podataka koji su nedostupni "običnim" tražilicama.

Rad je podijeljen u deset poglavlja. U prvom poglavlju će biti objašnjeno što je to zapravo YQL i bit će navedene neke njegove prednosti. Zatim će se detaljnije objasniti sintaksa YQL i način filtriranja podataka. Jedno poglavlje će se posvetiti načinima pokretanja YQL naredbi, kao i načinima autorizacije prilikom dohvaćanja privatnih podataka, nakon čega će biti prikazan način pretraživanja i dohvaćanja HTML² sadržaj uz pomoć YQL upita.

Gore navedena poglavlja će činiti teoretski dio rada. Također, bit će izrađena i aplikacija koja će pretraživati društvene mreže u potrazi za podacima i komentarima o hrvatskom turizmu. Aplikacija će biti izrađena uz pomoć Titanium Appcelerator platforme te će biti dokumentirana u posebnom poglavlju.

Za kraj ću kroz kritički osvrt i zaključak iznijeti vlastito mišljenje o YQL jeziku te prenijeti vlastita iskustva prikupljena tijekom istraživanja ove teme.

¹ Structured Query Language

² HyperText Markup Language

2. Što je YQL?

Kao što je rečeno u uvodu, YQL je jezik vrlo sličan SQL-u koji omogućava dohvaćanje, filtriranje i kombiniranje podataka primljenih od strane velikog broja Web servisa. Neki od najpoznatijih servisa kojima je moguće pristupiti uz pomoć YQL upita su Twitter, Flickr, Amazon i WordPress. Što se tiče tipova podataka, moguće je pretraživanje podataka koji se nalaze u XML³, JSON⁴, RSS⁵, HTML, FEED, ATOM ili CSV⁶ formatu. Međutim, dohvaćeni podaci mogu biti samo u XML ili JSON formatu, ovisno o izboru korisnika.

Iako je predstavljen prije samo nekoliko godina (listopad 2008.godine ⁷), YQL je vrlo brzo prepoznat kao vrlo koristan jezik te se počeo masovno koristiti. Danas postoji vrlo dobra korisnička podrška u vidu blogova i foruma na kojima se mogu naći odgovori na neka zahtjevnija pitanja.

Sintaksa je vrlo jednostavna jer sadrži jednake naredbe kao i SQL jezik, što omogućava vrlo brzo učenje jezika. Najčešće se koriste upiti koji sadrže ključnu riječ SELECT, a moguće je koristiti i INSERT, UPDATE, DELETE, SHOW i DESC ključne riječi. Također, moguće je filtrirati podatke putem ključne riječi WHERE te je moguće stvarati ugniježdene SELECT upite. U nastavku rada će svaki od gore spomenutih pojmova, kao i mnogi drugi, biti detaljnije opisani.

³ eXtensible Markup Language

⁴ JavaScript Object Notation

⁵ Really Simple Syndication

⁶ Comma Separated Values

⁷ Trevor J.: YQL Launched, [Internet], <<http://pcworld.about.net/od/software/services/Yahoo-Extends-YQL-Web-Data-Que.htm>>, učitano: 21.08.2011.

3. Sintaksa

Sintaksa YQL-a, kao što je već rečeno u uvodu, nalikuje na SQL sintaksu, koja je poznata razvojnim programerima, ali i većini ostalih informatičara imalo upoznatih s radom s bazama podataka. Najčešće korištene naredbe su SELECT, INSERT, DELETE i UPDATE kojima se dohvaćaju, manipuliraju i filtriraju željeni podaci. Također, koriste se i naredbe poput SHOW TABLES, DESC, USE ili SET koje će, zajedno s prethodno navedenim naredbama, biti detaljnije opisane u nastavku poglavlja.

3.1. SELECT

Kod YQL-a naredba SELECT omogućava nam dohvaćanje željenih informacija iz različitih web izvora, od Yahoo! Web servisa do web sadržaja zapisanih u HTML, XML, FEED, ATOM, JSON, CSV ili RSS formatu. Bez obzira iz kojih izvora dohvaćali podatke, oni će uvijek biti vraćeni u obliku XML ili JSON dokumenta, ovisno o našem izboru.

Sintaksa SELECT naredbe je specifična i potrebno je poznavati pravila i redoslijed upotrebe ključnih riječi kako bi se ispravno koristila. Pravilna sintaksa glasi:

SELECT podaci FROM tablica WHERE filter [| funkcija]

Dio gornje naredbe "podaci" se odnosi na polja s podacima koja želimo dohvatiti. Svako traženo polje odgovara XML elementu ili JSON objektu vraćenom sa SELECT naredbom. Prilikom pretraživanja moguće je odabrati samo jedno polje (npr. SELECT ime), kombinaciju više polja (npr. SELECT ime, prezime) ili odabrati sva polja (SELECT *).

Dio naredbe "tablica" odnosi se na tablicu iz koje se dohvaćaju podaci, a koja može biti u obliku Open Data tablica (XML dokument) ili u predefiniranom YQL obliku (npr. local.search). Važno je napomenuti da se može pretraživati samo jedna tablica, a ne više njih kao u SQL-u.

Filter služi za filtriranje podataka po određenim kriterijima, što će detaljnije biti obrađeno u jednom od sljedećih poglavlja.

Posljednji dio naredbe "funkcija" je opcionalan i odnosi se na ugrađene funkcije kojima se postižu dodatne funkcionalnosti. Ove funkcije se primjenjuju nakon što SELECT naredba završi sve ostale operacije, uključujući i filtriranje. Sljedeća tablica sadrži objašnjenje tih funkcija te njihove primjere.

Tablica 3.1. Funkcije kod SELECT upita ⁸

Funkcija	Argument	Primjer	Opis
sort	field [descending]	sort(field="nadimak", descending="true")	Sortira rezultat prema specificiranom <i>field</i> argumentu. Početna vrijednost opcionalnog argumenta <i>descending</i> jest <i>false</i> .
tail	count	tail(count=4)	Dohvaća posljednji <i>count</i> element.
truncate	count	truncate(count=4)	Dohvaća prvi <i>count</i> element.
reverse	(none)	reverse()	Obrće se redoslijed elemenata.
unique	field	unique(field="Ocjena.Pro sječnaOcjena")	Uklanja elemente s dvostrukom vrijednošću specificiranom u <i>field</i> argumentu. Prvi element sa specificiranom vrijednošću ostaje u rezultatu.
sanitize	[field]	sanitize(field='nadimak')	Pročišćava izlaz za sigurno HTML renderiranje. Za pročišćavanje svih vraćenih polja potrebno je ispustiti argument <i>field</i> .

3.2. Ažuriranje podataka

Na prethodnim stranicama opisana je naredba SELECT te je navedeno da služi za dohvaćanje strukturiranih podataka. S druge strane, u nastavku će biti opisane naredbe INSERT, UPDATE i DELETE, kojima je moguće manipulirati podacima, točnije upisivati podatke, mijenjati ih ili brisati. Međutim, za manipuliranje podacima je potrebna autentikacija poput korisničkog imena i lozinke ili tajnog API ⁹ ključa. U nastavku će navedene stvari biti detaljnije objašnjene.

⁸ Yahoo! Inc.: YQL Guide: Sort and Other Functions, [Internet], <<http://developer.yahoo.com/yql/guide/sorting.html>>, učitano: 21.08.2011.

⁹ Application programming interface

3.2.1. INSERT

Naredba INSERT služi za upisivanje podataka u već postojeće tablice koje se nalaze u formatu prikladnom za YQL. Ovime je omogućeno upisivanje novih Twitter statusa, dodavanje komentara na blog ili jednostavno pohranjivanje podataka u bazu podataka.¹⁰ Ipak, kako bi se spriječile moguće malverzacije, kod korištenja INSERT naredbe je potrebna određena vrsta autentikacije koja može biti izvršena na nekoliko načina, poput korištenja posebnih ključeva ili kombinacijom korisničkog imena i lozinke. Kasnije će biti prikazan primjer koji će ilustrirati navedeno.

Sintaksa YQL INSERT naredbe je vrlo slična SQL INSERT naredbi i glasi:

*INSERT INTO (tablica) (lista elemenata odvojenih zarezom) VALUES (lista elemenata odvojenih zarezom)*¹¹

U ovom slučaju "(tablica)" se odnosi na tablice s podacima kojima određeni korisnici mogu mijenjati sadržaj, poput YQL predefiniраниh tablica (twitter.status) ili Open Data tablica.

Zatim se navodi lista elemenata koja označavaju mjesto gdje YQL INSERT naredba umeće nove podatke. U slučaju dodavanja novog Twitter statusa navode se elementi (status, korisničko_ime, lozinka).

Argument VALUES označava da u nastavku slijede podaci koji se umeću. Važno je naglasiti da nakon argumenta VALUES slijede navodnici unutar kojih se nalaze vrijednosti koje će biti upisane. Vrijednosti se odvajaju zarezom, dvostrukim ili jednostrukim, i osjetljive su na veličinu slova. Primjer INSERT naredbe¹²:

INSERT into twitter.status (status,username,password) VALUES ("Playing with INSERT, UPDATE and DELETE in YQL", "twitterusername", "twitterpassword")

¹⁰ Choi J.:YQL:INSERT INTO internet, [Internet], <http://developer.yahoo.com/blogs/ydn/posts/2009/07/yql_insert/>, učitano: 21.08.2011.

¹¹ Yahoo! Inc.: YQL Guide: Syntax of I/U/D, [Internet], <http://developer.yahoo.com/yql/guide/iud-syntax.html>

¹² Willison S.: Simon Willison's Weblog, [Internet], <<http://simonwillison.net/2009/Jul/8/yql/>>, učitano: 21.08.2011.

3.2.2. UPDATE

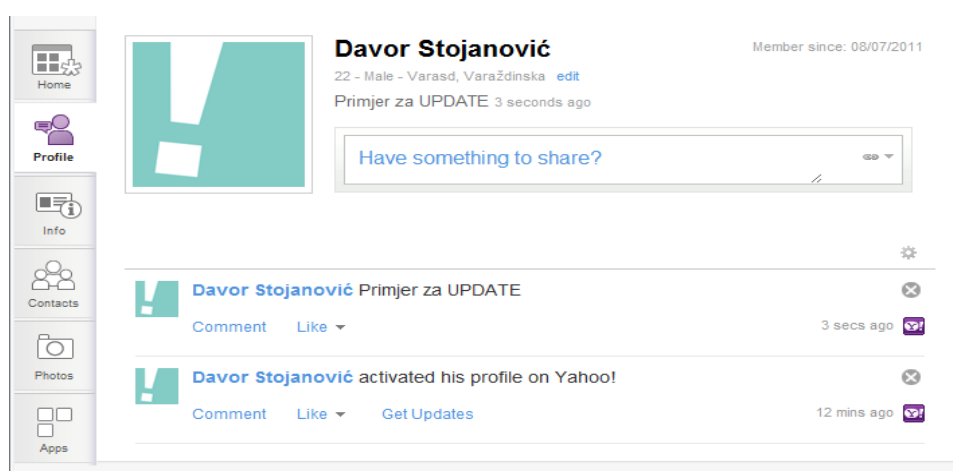
Pomoću UPDATE naredbe moguće je promijeniti već postojeće podatke koji se nalaze na Webu, kao npr. promijeniti svoj Yahoo! status. Sintaksa je sljedeća:

UPDATE (tablica) SET element=vrijednost WHERE filter

Nakon ključne riječi SET navode se elementi čiju vrijednost želimo promijeniti (npr. SET status = "Primjer za UPDATE"). Ovime smo prosljedili informacije koje želimo promijeniti, ali to samo po sebi nije dovoljno. Potrebno je detaljnije definirati gdje točno želimo ubaciti nove podatke, a za to nam služi argument WHERE. Nakon ključne riječi WHERE uz pomoć filtera točno se specificiraju čiji podaci će biti ažurirani (najčešće se navodi ID). U nastavku će biti prikazano ažuriranje statusa na Yahoo! profilu preko YQL konzole, o kojoj će kasnije nešto više biti rečeno.

Upit za ažuriranje statusa na Yahoo! profilu je vrlo jednostavan. Ključna stvar je tablica preko koje se može pristupiti Yahoo! statusima, a njoj se pristupa preko "social.profile.status" tablice. Također, važno je znati i koji će se filter iskoristiti. U slučaju ažuriranja statusa preko YQL konzole dovoljan je filter "WHERE guid=me"., gdje je "guid"¹³ jedinstveni identifikator kojem je pridružen jedinstveni broj; u slučaju promjene vlastitog statusa dovoljno je upisati "me". Cijeli upit izgleda ovako:

UPDATE social.profile.status SET status="Primjer za UPDATE" WHERE guid=me



Slika 3.1. Primjer korištenja UPDATE naredbe¹⁴

¹³ Globally unique identifier

¹⁴ Stojanović D.: Yahoo! PULSE, [Internet], <<http://pulse.yahoo.com/y>>, učitano 21.08.2011.

3.2.3. DELETE

Uz pomoć naredbe DELETE moguće je brisati podatke koji se nalaze u tablicama prilagođenima YQL-u. Sintaksa je vrlo jednostavna i glasi:

DELETE FROM (tablica) WHERE filter

U ovom slučaju, kao i prethodnim, navodi se ime tablice iz koje želimo obrisati podatke. Podaci koje navodimo u filteru se odnose na jednoznačno određivanje podatka kojeg želimo obrisati. U slučaju brisanja statusa s Yahoo! profila ovdje se navodi guid vlasnika, ID poruke i izvor s kojeg je poruka kreirana, što se može vidjeti na sljedećem primjeru ¹⁵.

DELETE FROM social.updates WHERE guid="me" and suid="9.3341" AND source="y.presence"

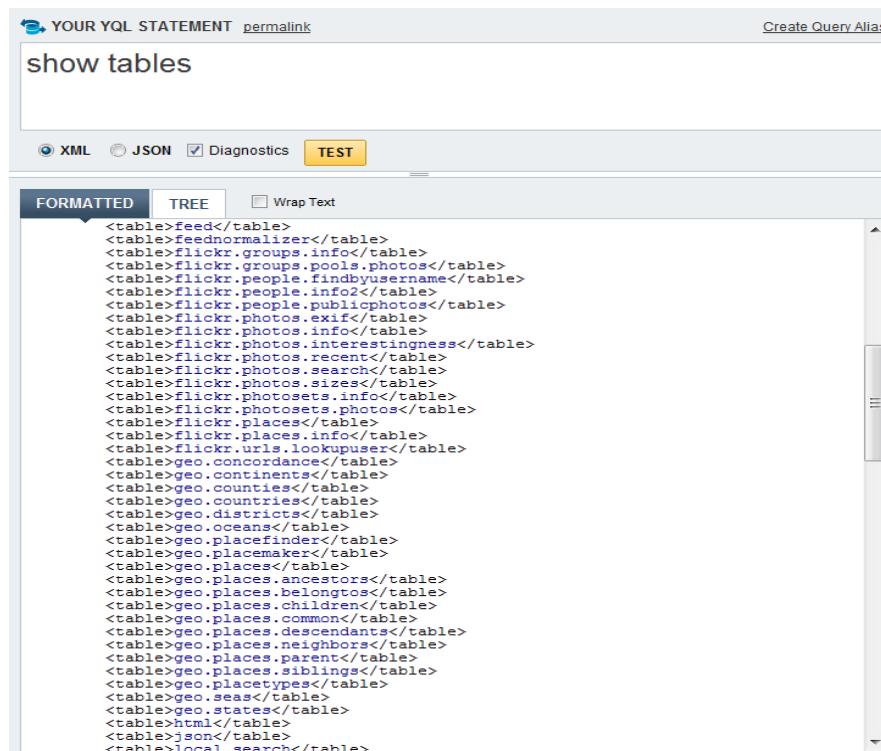
3.3. SHOW

Naredba SHOW je vrlo jednostavna, ali to ne umanjuje njezinu korisnost. Prima samo jedan argument i u cijelosti izgleda ovako:

SHOW tables

Njezina zadaća je prikaz svih tablica koje korisnik može koristiti u svojim YQL naredbama. To je vrlo korisno jer razvojni programeri na jednom mjestu imaju popis svih tablica kojima mogu pristupiti u svojim aplikacijama. Rezultat naredbe je XML ili JSON dokument, ovisno o korisnikovom izboru. Na sljedećoj slici je prikazan dio rezultata dobiven SHOW naredbom u XML formatu.

¹⁵ Yahoo! Inc.: YQL Guide: Updates Tables, [Internet], <http://developer.yahoo.com/social/rest_api_guide/updates_table.html>, učitano 21.08.2011.



Slika 3.2. Dio rezultata dobiven naredbom *SHOW table*¹⁶

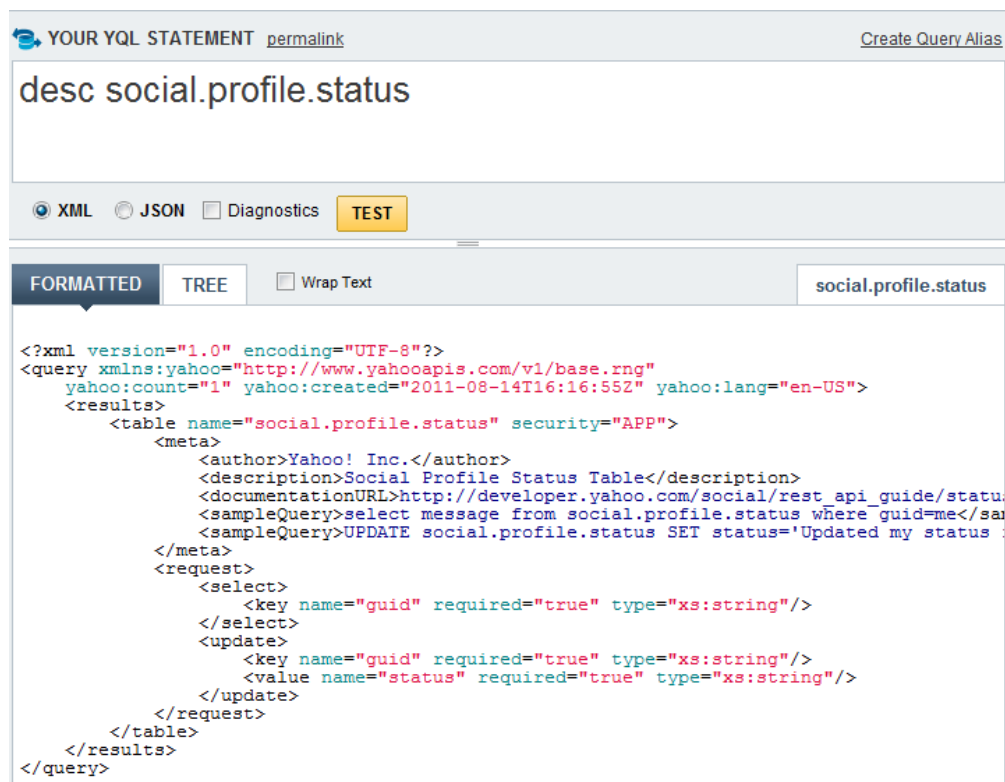
3.4. DESC

Uz pomoć naredbe DESC možemo saznati detaljnije informacije o dostupnim YQL tablicama. Točnije, saznajemo strukturu određene tablice te tako doznajemo na koji joj način možemo pristupiti i koje podatke možemo izvući iz nje. Upotreba DESC naredbe bit će demonstrirana na primjeru tablice koja sadrži Yahoo! profilne statuse, koju smo upoznali kod korištenja UPDATE naredbe. Naredba glasi:

DESC social.profile.status

Sljedeća slika predstavlja rezultat naredbe u XML formatu.

¹⁶ Yahoo! Inc.: YQL Console, [Internet], <<http://developer.yahoo.com/yql/console/#h=show%20tables>>, učitano 21.08.2011.



Slika 3.3. Dio rezultata dobiven DESC naredbom ¹⁷

Iz rezultata možemo očitati podatke poput autora tablice, njezinog opisa ili primjere upita nad tablicom, a koji se nalaze unutar "meta" taga. Međutim, razvojnim programerima je važniji tag "request" unutar kojega je vidljivo koje naredbe je moguće vršiti nad tablicom te koji se argumenti moraju proslijediti. U ovom slučaju kod naredbe SELECT i UPDATE unutar "key" taga, koji označava potrebne argumente, nalazi se atribut "guid", što znači da je on neophodan prilikom kreiranja upita. Također, vidljivo je i da je kod UPDATE naredbe potrebno navesti i vrijednost atributa "status", što je već demonstrirano na jednom od prethodnih primjera.

¹⁷ Yahoo! Inc.: YQL Console, [Internet], <<https://developer.yahoo.com/yql/console/?q=show%20tables#h=desc%20social.profile.status>>, učitano 21.08.2011.

4. Filtriranje

O filtriranju rezultata je već bilo riječi u prethodnom poglavlju. Navedeno je da filter slijedi nakon ključne riječi `WHERE` te da služi za dohvaćanje točno određenih podataka. Drugim riječima, bez filtera bi bilo nemoguće izvući željene podatke, što potvrđuje i činjenica da neki servisi niti ne vraćaju podatke ukoliko im nisu proslijeđeni određeni filteri. Do podataka koji su filteri potrebni programeri mogu doći koristeći naredbu `DESC`, što je objašnjeno u prethodnom primjeru. Osim ilustriranja načina na koji se podaci mogu filtrirati, u ovom poglavlju će biti i riječi o različitim vrstama filtriranja, lokalnom i udaljenom.

4.1. Lokalno filtriranje

Da bismo mogli filtrirati podatke potrebno ih je prethodno dohvatiti. Kao što je već rečeno, postoje dvije vrste filtriranja. Lokalno filtriranje se odnosi na filtriranje nad cjelokupno dohvaćenim podacima (nad cijelom tablicom), što znači da se filtriranje provodi na korisničkoj strani, dok se kod udaljenog filtriranja ne dohvaća cijela tablica, već samo podaci koji zadovoljavaju filter, što znači da se filtriranje vrši na strani poslužitelja. U nastavku će biti detaljnije objašnjeno lokalno filtriranje. Sintaksa lokalnog filtera izgleda ovako:

`WHERE element operator_uspoređivanja vrijednost`

Element se odnosi na naziv XML elementa ili JSON objekta prema kojem želimo filtrirati rezultat (npr. `title`, `id`). Kod atributa vrijednost unutar navodnika moguće je upisati tekst, broj ili decimalni broj. Što se tiče operatora uspoređivanja, moguće je koristiti samo one koji su integrirani u YQL. Slijedi tablica koja sadrži popis tih operatora i njihov kratak opis.

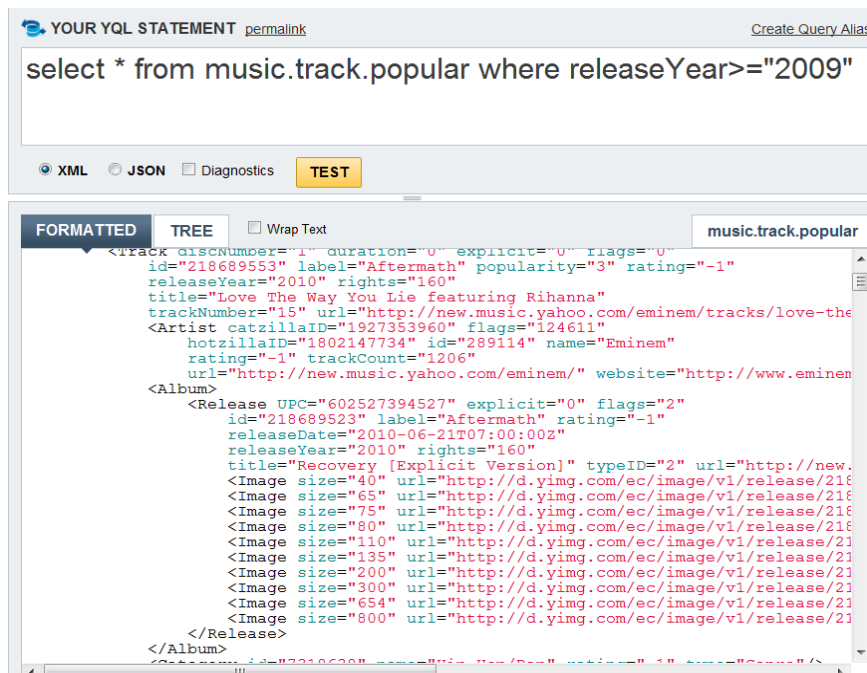
Tablica 4.1. Operatori uspoređivanja ¹⁸

Operator	Opis
=	Jednako
!=	Nije jednako
>	Veće od
<	Manje od
>=	Veće od ili jednako
<=	Manje od ili jednako
[NOT] IN	Provjerava nalazi li se određena vrijednost u skupu vrijednosti. Nakon ovog operatora može slijediti još jedno pretraživanje (sub-select) ili skup vrijednosti odvojen zarezom.
IS [NOT] NULL	Provjerava postoji li određeni element u rezultatu. Izraz IS NULL je točan ako se element ne nalazi u rezultatu.
[NOT] LIKE	Provjerava postoji li poklapanje teksta. Usporedba teksta je osjetljiva na veličinu slova. Znak "%" označava nula ili više znakova. Na primjer, sys% će se poklapati s bilo kojom riječi koja počinje na sys.
[NOT] MATCHES	Provjerava postoji li preklapanje teksta, regularni izrazi su dozvoljeni. Usporedba je osjetljiva na veličinu slova.

Primjer jednog upita s lokalnim filtriranjem. Pretražujemo Yahoo! servis popularne glazbe i tražimo glazbu nastalu 2009. godine ili kasnije. Dio rezultata u XML formatu se može vidjeti na slici ispod upita.

*SELECT * FROM music.track.popular WHERE releaseYear>="2009"*

¹⁸ Yahoo! Inc.: YQL Guide: Filtering Query Results (WHERE), [Internet], <<http://developer.yahoo.com/yql/guide/filters.html>>, učitano 21.08.2011.



Slika 4.1. Lokalno filtriranje ¹⁹

4.2. Udaljeno filtriranje

Kao što je rečeno u uvodu poglavlja, udaljeno filtriranje se vrši na strani poslužitelja, a tek nakon filtriranja se dohvaćaju traženi podaci. Sintaksa udaljenog filtriranja je sljedeća ²⁰:

WHERE *ulazni_parametar* = *vrijednost*

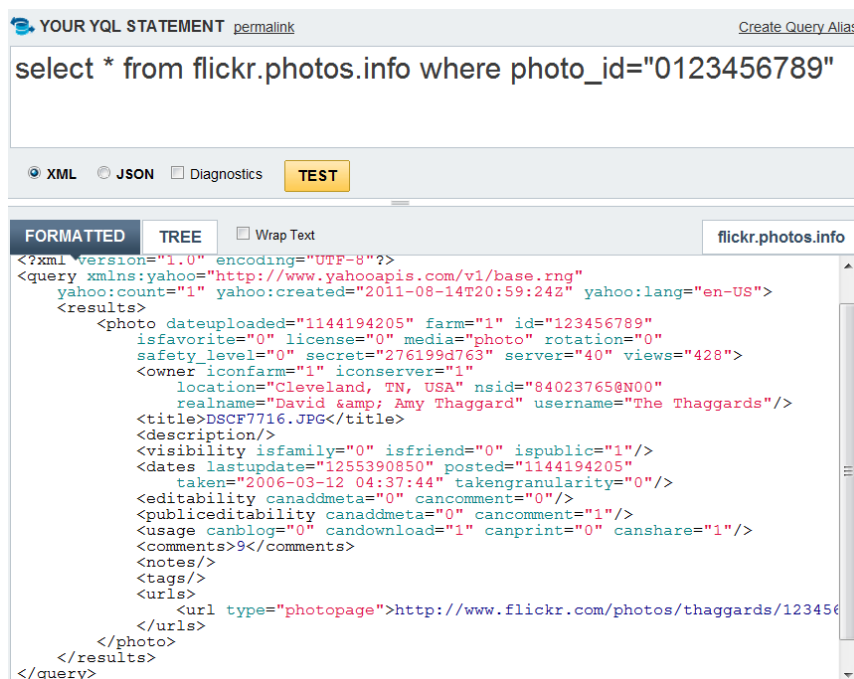
Argument *ulazni_parametar* predstavlja parametar koji YQL šalje vanjskom servisu na temelju kojeg se vrši filtriranje. Ulazne parametre je moguće saznati koristeći naredbu `DESC`, što je već objašnjeno u jednom od prethodnih primjera. Između argumenata *ulazni_parametar* i *vrijednost* dopušteno je koristiti samo operator jednakosti (`=`), ne i ostale Boolean operatore kao kod lokalnog filtriranja. Kod argumenta *vrijednost* dopušteno je koristiti tekstualne znakove, brojeve i decimalne brojeve. Slijedi primjer koji ilustrira gore objašnjeno.

¹⁹ Yahoo! Inc.: YQL Console, [Internet], http://developer.yahoo.com/yql/console/#h=select%20*%20from%20music.track.popular%20where%20releaseYear%3E%3D%222009%22, učitano 21.08.2011.

²⁰ Yahoo! Inc.: YQL Guide: Filtering Query Results (WHERE), [Internet], <http://developer.yahoo.com/yql/guide/filters.html>, učitano 21.08.2011.

Pretražuje se Flickr i traži se fotografija koja ima ID jednak 0123456789. Upit je sljedeći, a rezultat se može vidjeti na slici ispod.

```
SELECT * FROM flickr.photos.info where photo_id = "0123456789"
```



Slika 4.2. Udaljeno filtriranje²¹

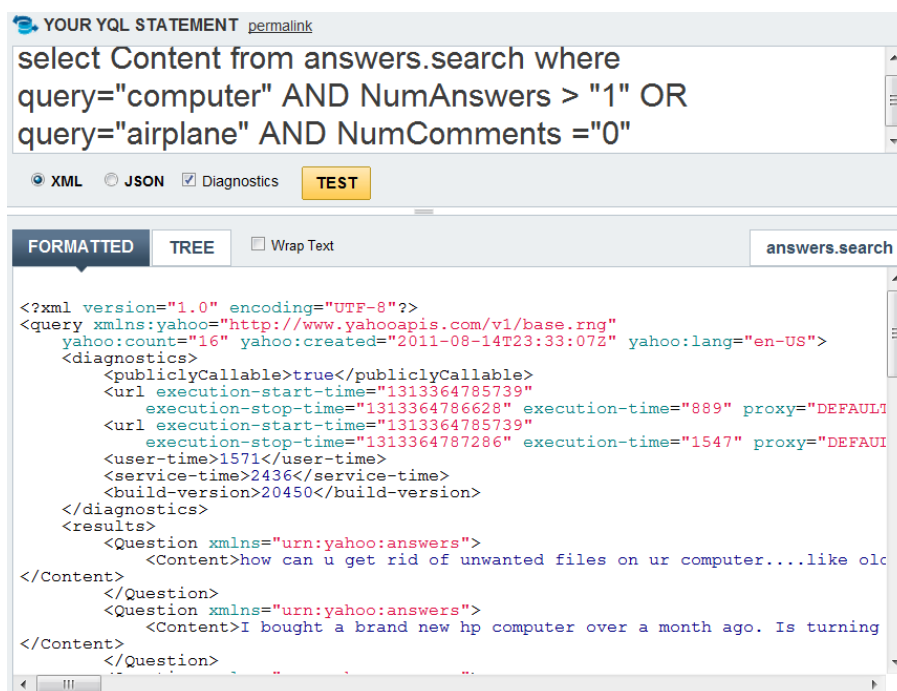
4.3. Kombinirano filtriranje

Kod složenijih upita moguće je kombinirati lokalno i udaljeno filtriranje uz pomoć Boolean operatora AND i OR. Operator AND ima značenje logičkog "i", dok operator OR ima značenje logičkog "ili". Slijedi primjer pretraživanja Yahoo! answers servisa gdje tražimo pitanja koja sadrže ključnu riječ "computer" i imaju više od jednog odgovora ili pitanja koja sadrže ključnu riječ "airplane" i nemaju niti jedan komentar. Upit izgleda ovako:

```
SELECT Content FROM answers.search WHERE query="computer" AND NumAnswers > "1"
OR query="airplane" AND NumComments = "0"
```

²¹ Yahoo! Inc.: YQL Console, [Internet], <http://developer.yahoo.com/yql/console/#h=select%20*%20from%20flickr.photos.info%20where%20photo_id%3D%20123456789%22>, učitano 21.08.2011.

Parametri `query="computer"` i `query="airplane"` su primjeri udaljenog filtriranja jer predstavljaju obvezne ulazne parametre, dok su parametri `NumAnswers > "1"` i `NumComments ="0"` primjeri lokalnog filtriranja. Rezultat upita je prikazan na sljedećoj slici.



Slika 4.3. Kombinirano filtriranje²²

4.4. Limitiranje

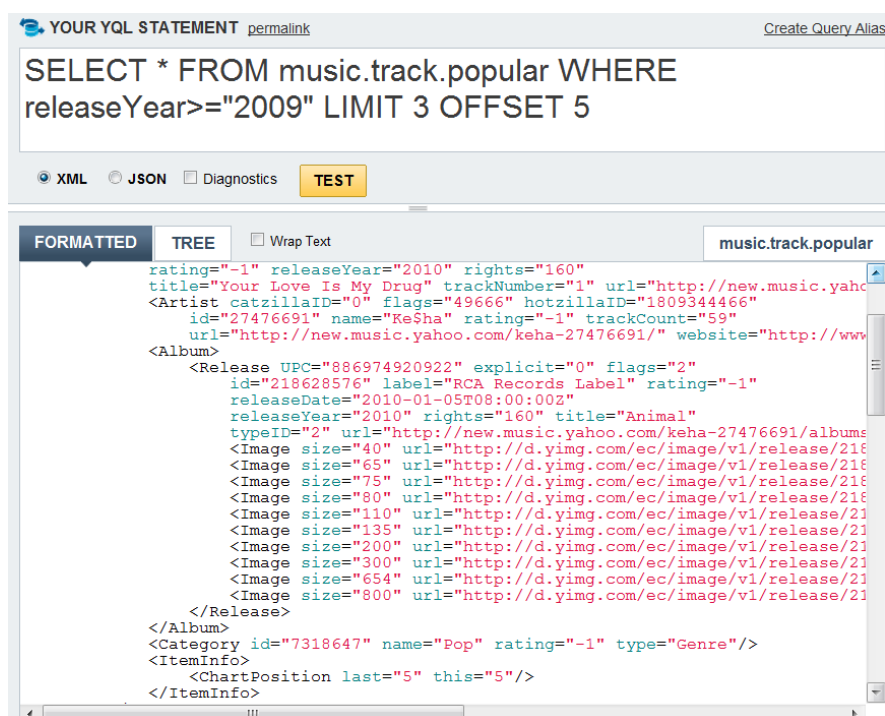
Iako strogo gledano limitiranje veličine tablice primljenih podataka ne predstavlja filtriranje podataka, bit će opisano u ovom poglavlju zbog nekih dodirnih točaka s uporabom filtera. Kao što je poznato, određene tablice podataka sadrže ogromne količine podataka. Kako bi programeri imali kontrolu nad količinom podataka koju žele primiti i prikazati u svojoj aplikaciji, koristi se limitiranje veličine tablice. Kao i kod filtriranja, ovdje također postoje dvije vrste kontroliranja podataka, lokalno i udaljeno, koje će u nastavku biti objašnjene.

²² Yahoo! Inc.: YQL Console, [Internet], <http://developer.yahoo.com/yql/console/#h=select%20*%20from%20answers.search%20where%20query%3D%22computer%22%20AND%20NumAnswers%20%3E%20%221%22%20OR%20query%3D%22airplane%22%20AND%20NumComments%20%3D%220%22>, učitano 21.08.2011.

4.4.1. Lokalno limitiranje

Lokalno kontroliranje podataka se izvodi na strani klijenta. Točnije, limitiranje slijedi nakon što se dohvate svi podaci, koji zadovoljavaju SELECT upit, od stane servisa koji se poziva. Bez limitiranja maksimalan broj polja podataka primljenih SELECT upitom iznosi 5000. Ključne riječi kojima se ostvaruje limitiranje su LIMIT i OFFSET, nakon kojih slijedi cijeli broj koji označava njihovu vrijednost. Ključnom riječju LIMIT se određuje broj redova koji se dohvaćaju, a ključnom riječju OFFSET se određuje od kojeg reda se dohvaćaju podaci. Upotreba OFFSET-a je opcionalna, a njezina početna vrijednost je 0, što znači da se podaci prikupljaju od prvog reda tablice. U nastavku slijedi primjer koji je korišten kod lokalnog filtriranja, kojemu su dodani parametri "LIMIT 3 OFFSET 5", što znači da će se dohvatiti 3 rezultata počevši od petog.

*SELECT * FROM music.track.popular WHERE releaseYear>="2009" LIMIT 3 OFFSET 5*



Slika 4.4. Primjer lokalnog filtriranja²³

²³ Yahoo! Inc.: YQL Console, [Internet], <http://developer.yahoo.com/yql/console/#h=select%20*%20from%20music.track.popular%20where%20releaseYear%3E%3D%222009%22%20LIMIT%203%20OFFSET%205>, učitano 21.08.2011.

4.4.2. Udaljeno limitiranje

Za razliku od lokalnog limitiranja, udaljeno se izvodi na strani poslužitelja. Točnije, udaljenim limitiranjem se kontrolira broj redova podataka koji će YQL aplikacija primiti od servisa koji joj dostavlja podatke. To znači da se ne dohvaćaju svi podaci te se naknadno limitiraju, već se dohvaća unaprijed limitirani broj redova. Parametri udaljenog limitiranja se unose unutar zagrade () nakon specificiranja imena tablice iz koje se dohvaćaju podaci tako da se prvo navodi parametar OFFSET, a zatim LIMIT, npr.

```
SELECT * FROM music.track.popular(0,7) WHERE releaseYear>="2009"
```

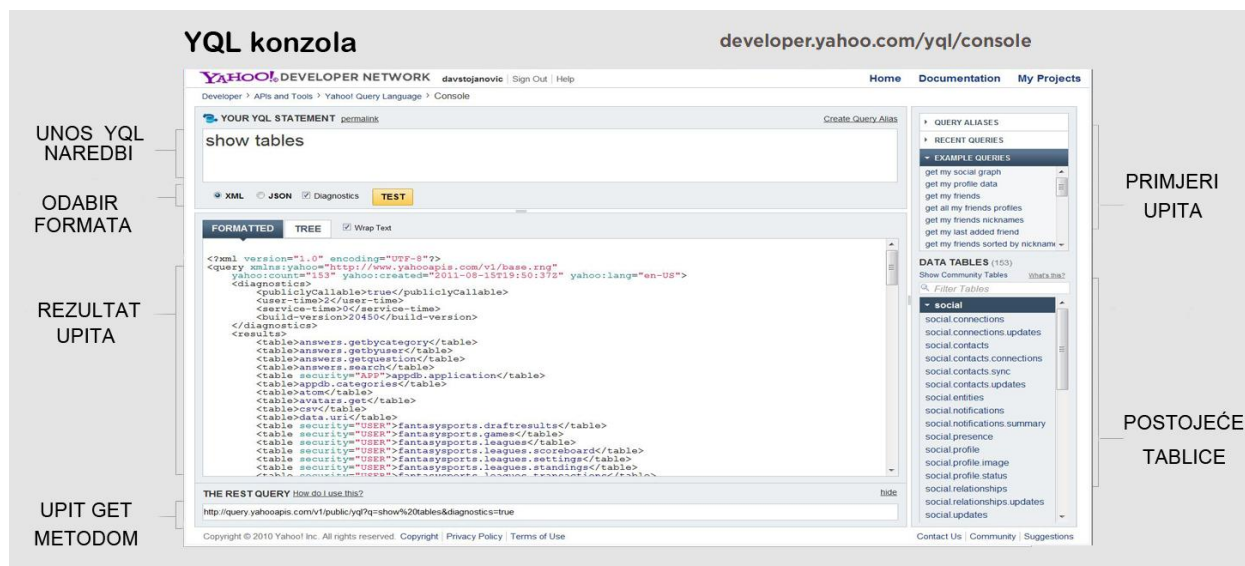
U ovom primjeru vrijednost OFFSET parametra jest 0, što znači da se podaci dohvaćaju od prvog reda koji zadovoljava upit. Vrijednost LIMIT parametra jest 7, što znači da će servis poslati samo 7 redova koji zadovoljavaju upit, a ne sve.

5. Pokretanje YQL naredbi

Na prethodnim stranicama je kroz primjere demonstrirano pokretanje određenih YQL naredbi preko YQL konzole. U ovom poglavlju će detaljnije biti objašnjeno sučelje konzole i njezin princip rada. Također, bit će objašnjen još jedan način pokretanja YQL naredbi; preko adrese <http://query.yahooapis.com> koristeći GET metodu.

5.1. YQL konzola

Sve dosadašnje naredbe su pokretane preko YQL konzole, što se može vidjeti na gornjim slikama. No, što je YQL konzola zapravo? Ona je web stranica na kojoj razvojni programeri i svi ostali korisnici mogu pokretati YQL naredbe i dohvaćati željene podatke, proučavati strukturu tablica s podacima, eksperimentirati s različitim upitima ili proučavati gotove primjere. Podaci se prikazuju u XML ili JSON formatu, ovisno o korisnikovom izboru. Ne postoji mogućnost eksportiranja podataka, ali zato korisnik ručno može spremiti dohvaćene podatke ukoliko to želi. Sučelje je vrlo jednostavno i intuitivno, što se može vidjeti na sljedećoj slici.



Slika 5.1. YQL konzola ²⁴

Na vrhu konzole se nalazi prostor za unos naredbi gdje korisnici upisuju svoje upite. Uneseni upit mora biti u skladu s YQL sintaksom, inače će konzola kao rezultat vratiti "error". Početna

²⁴ Yahoo! Inc.: YQL Console, [Internet], <<http://developer.yahoo.com/yql/console/>>, učitano 21.08.2011.

naredba koja se automatski pokreće prilikom otvaranja konzole jest "SHOW tables", koja ispisuje tablice podataka koje je moguće dohvatiti. Naredbe se pokreću klikom na gumb "TEST" ili pritiskom tipke "Enter".

Ispod prostora za unos naredbi se nalazi prostor gdje korisnici biraju u kojem formatu žele da im konzola prikaže rezultat. Moguće je odabrati XML i JSON format. Također, tu je moguće odabrati opciju "Diagnostics" koja detaljnije opisuje dohvaćeni XML ili JSON rezultat.

Srednji i najveći dio konzole zauzima prostor u kojem se ispisuje rezultat upita u odabranom formatu. Tu postoje opcije "FORMATTED" i "TREE" kojima je moguće promijeniti strukturu rezultata.

U gornjem desnom kutu postoji nekoliko opcija. Prva od njih je "QUERY ALIASES" koja služi za pohranjivanje upita. Pohranjenom upitu odabiremo ime kojeg možemo upotrijebiti kod pokretanja upita GET metodom. Slijedi opcija "RECENT QUERIES" koja nam izlistava listu prethodno upotrijebljenih upita. Treća opcija je "EXAMPLE QUERIES" koja sadrži nekoliko primjera gotovih upita koje korisnici mogu upotrijebiti kod učenja YQL jezika.

Ispod opcije "EXAMPLE QUERIES" nalazi se opcija "DATA TABLES" koja sadrži sve tablice s kojima je moguće komunicirati koristeći YQL. Prvotno su prikazane samo tablice čiji je autor Yahoo! kojih trenutno ima 153, dok odabirom opcije "Show Community Tables" prikazuju se i tablice ostalih servisa s kojima YQL može komunicirati, što ukupno daje 1193 tablica podataka.

U dnu stranice automatski se stvara tzv. "THE REST QUERY", korisnički upit u obliku linka, koji služi za pokretanje YQL upita GET naredbom. O tome će više riječi biti u sljedećem poglavlju.

5.2. GET metoda

U aplikacijama nije moguće direktno koristiti YQL konzolu za pokretanje YQL naredbi i dohvaćanje podataka, zbog čega je razvijen način pokretanja YQL naredbi GET metodom. Yahoo! je stvorio vlastiti Web servis koji omogućava izvršavanje YQL upita, a ovisno o vrsti podataka koje želimo prikupiti nalazi se na dvije Web adrese:

- [http://query.yahooapis.com/v1/public/yql?\[query_params\]](http://query.yahooapis.com/v1/public/yql?[query_params]) - omogućava pristup javnim podacima za koje nije potrebna autorizacija
- [http://query.yahooapis.com/v1/yql?\[query_params\]](http://query.yahooapis.com/v1/yql?[query_params]) - omogućava pristup javnim i privatnim podacima, autorizacija se vrši preko OAuth protokola

Nakon URL-a potrebno je navesti parametre upita, a moguće parametre i njihov opis dan je u sljedećoj tablici.

Tablica 5.1. Parametri kod GET metode ²⁵

Parametar upita	Obvezno?	Zadana vrijednost	Opis
q	Da	-	YQL upit koji će biti izvršen.
format	Ne	xml	Format u kojem će rezultat biti prikazan (XML ili JSON). Zadana vrijednost je XML.
callback	Ne	-	Naziv JavaScript povratne funkcije za JSONP format.
diagnostics	Ne	true	Dodatne informacije o upitu poput vremena proteklog za dohvaćanje podataka.

Ovo je primjer jednog takvog upita koji je automatski nastao u YQL konzoli:

<http://query.yahooapis.com/v1/public/yql?q=show%20tables&diagnostics=true>

Gornji upit ('SHOW tables') služi za izlistavanje svih tablica s kojima YQL može komunicirati. Pokretanjem ovog upita preko web preglednika kao rezultat dobivamo stranicu u XML formatu koji sadrži tražene podatke o tablicama.

²⁵ Yahoo! Developer Network, Yahoo! Query Language (YQL) Guide, [Internet], <http://openhacklondon.pbworks.com/f/yql_guide.pdf>, učitano 21.08.2011.

```
<?xml version="1.0" encoding="UTF-8" ?>
<query xmlns:yahoo="http://www.yahooapis.com/v1/base.rng" yahoo:count="153" yahoo:created="2011-08-16T12:45:17Z"
yahoo:lang="en-US">
  <diagnostics>
    <publiclyCallable>true</publiclyCallable>
    <user-time>2</user-time>
    <service-time>0</service-time>
    <build-version>20450</build-version>
  </diagnostics>
  <results>
    <table>answers.getbycategory</table>
    <table>answers.getbyuser</table>
    <table>answers.getquestion</table>
    <table>answers.search</table>
    <table security="APP">appdb.application</table>
    <table>appdb.categories</table>
    <table>atom</table>
    <table>avatars.get</table>
    <table>csv</table>
    <table>data.uri</table>
    <table security="USER">fantasysports.draftresults</table>
    <table security="USER">fantasysports.games</table>
    <table security="USER">fantasysports.leagues</table>
    <table security="USER">fantasysports.leagues.scoreboard</table>
    <table security="USER">fantasysports.leagues.settings</table>
    <table security="USER">fantasysports.leagues.standings</table>
    <table security="USER">fantasysports.leagues.transactions</table>
    <table security="USER">fantasysports.players</table>
    <table security="USER">fantasysports.players.ownership</table>
  </results>
</query>
```

Slika 5.2. Primjer rezultata upita GET metodom ²⁶

²⁶ Yahoo! Inc.: YQL Console, [Internet], <<http://query.yahooapis.com/v1/public/yql?q=show%20tables&diagnostics=true>>, učitano 21.08.2011.

6. Autorizacija

Kao što je već rečeno, YQL omogućava pristup javnim i privatnim podacima. Pristup određenim javnim podacima (npr. RSS vijestima) je slobodan i nije potrebna autorizacija. Međutim, javni podaci većine web servisa (npr. Twitter, LinkedIn) nisu dostupni bez autorizacije aplikacije koja im želi pristupiti. Što se tiče pristupa privatnim podacima, autorizacija je obavezna i mnogo je složenija od one kod pristupa javnim podacima.

Autorizacija YQL aplikacija se vrši pomoću OAuth protokola, a ovisno o vrsti podataka kojima se želi pristupiti, postoje dvije vrste autorizacije, tzv. "two-legged" i "three-legged" autorizacija. OAuth je protokol koji aplikacijama omogućava pristup korisničkim informacijama iz različitih web servisa (Yahoo!, Google, Twitter, LinkedIn), pri čemu aplikacija mora biti odobrena od strane krajnjeg korisnika ²⁷. Podaci krajnjih korisnika se prenose sigurno bez otkrivanja identiteta korisnika. U nastavku će u vrlo kratkim crtama biti opisane dvije vrste autorizacija jer bi detaljnije ulaženje u njihov opis predstavljalo materijal još jednog završnog rada.

6.1. Two-legged OAuth

Kod aplikacija koje koriste two-legged OAuth protokol autorizacija se odvija između dvije strane: aplikacije (korisnika) i izvora javnih podataka (pružatelja usluge) ²⁸. Autorizacija se odvija uz pomoć dva tajna ključa, "Consumer Key" i "Consumer Secret", koji su karakteristični za svaku aplikaciju, a izraženi su kao kombinacija brojeva i slova. Sljedeća slika predstavlja jedan takav primjer.



```
Consumer Key: dj0yJmk9aDg3MWS3bjNvNmQ0JmQ9WVdrOVNtVklUVFJlTm5NbWNHbz1ORFkzT0RJNU16WXkmcz
Consumer Secret: 18dffc07eb6b5526467d7b724742e64a3746565c
```

Slika 6.1. OAuth ključevi ²⁹

²⁷ Yahoo! Inc.:YQL Guide: Introduction, [Internet], <<http://developer.yahoo.com/oauth/guide/about.html>>, učitano 21.08.2011.

²⁸ Yahoo! Inc.:YQL Guide: Private Data v. Public Data, [Internet], <http://developer.yahoo.com/oauth/guide/oauth-private_public_data.html#>, učitano 21.08.2011.

²⁹ Yahoo! Inc.:YQL Social, [Internet], <<https://developer.apps.yahoo.com/projects/JeHM4n6s>>, učitano 21.08.2011.

U aplikaciji je potrebno navesti vrijednosti ovih ključeva prilikom YQL upita kako bi se izvršila autorizacija, nakon čega do tada zaštićeni javni podaci postaju dostupni. Na sljedećoj slici je prikazan primjer uporabe two-legged OAuth autorizacije. Vidljivo je da se nakon deklariranja varijable koja sadrži ključeve ista poziva kod YQL upita.

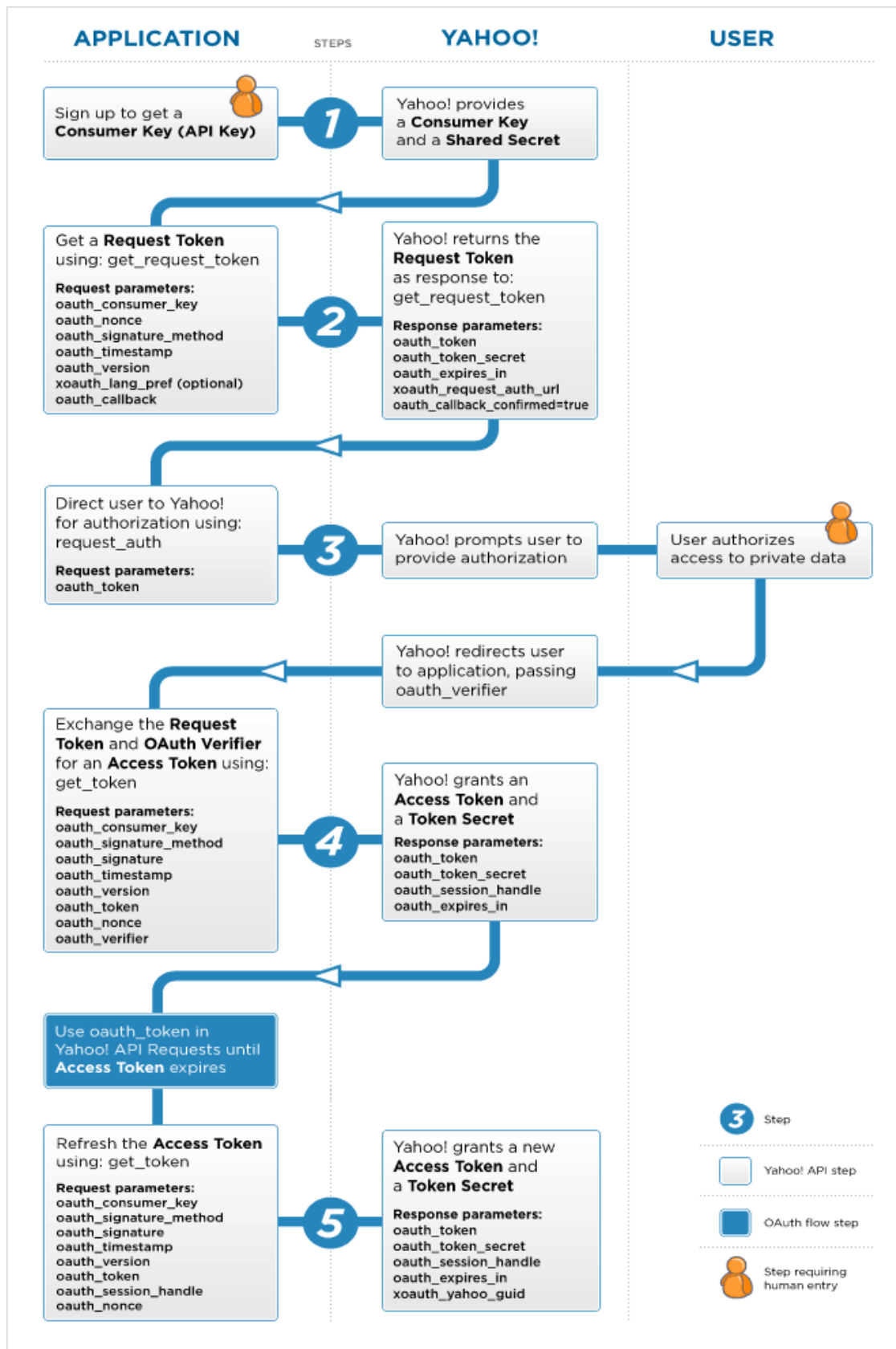
```
define("API_KEY","insert_key_here");
define("SHARED_SECRET","insert_secret_here");
//YahooApplication class is used for two-legged authorization,
$two_legged_app = new YahooApplication(API_KEY,SHARED_SECRET);
if ($two_legged_app == NULL) {
    echo "<p>cannot get two_legged_app</p>";
    exit;
}
//yql query
$firehose = 'select * from social.updates.search where link="engadget.com" and source="y.del
//call the signed query
$firehoseResponse = $two_legged_app->query($firehose);
if ($firehoseResponse == NULL) {
    echo "<p>something went wrong</p>";
} else {
    //echo "<pre>";
    //print_r($firehoseResponse);
    //echo "</pre>";
    //print out the results
$result = $firehoseResponse->query->results->update;
foreach ($result as $r) {
    $title = $r->title;
    $url = $r->link;
    echo "<p><a href=\"\$url\">\$title</a></p>";
}
}
```

Slika 6.2. OAuth primjer³⁰

6.2. Three-legged OAuth

Za pristup privatnim podacima potrebna je autorizacija putem three-legged OAuth protokola. Takva vrsta autorizacije se odvija između tri strane: aplikacije, izvora javnih podataka (pružatelja usluge) i krajnjeg korisnika. U ovom slučaju krajnji korisnik odlučuje hoće li omogućiti aplikaciji pristup do svojih privatnih podataka. Autorizacija se odvija kroz nekoliko koraka, a cilj razvojnog programera je dobiti ključeve "Access Token" i "Token Secret" pomoću kojih će moći pristupiti privatnim podacima. Koraci za dobivanje ključeva potrebnih za pristup privatnim podacima iz Yahoo servisa su opisani na sljedećoj slici.

³⁰ Smith M.: Two-legged OAuth Example for YQL with PHP, [Internet], <<http://mattsmithblog.wordpress.com/2010/06/15/two-legged-oauth-example-for-yql-with-php/>>, učitano 21.08.2011.



Slika 6.3. Three-legged OAuth³¹

³¹ Yahoo! Inc.:YQL Guide: OAuth Authorization Flow, [Internet], <<http://developer.yahoo.com/oauth/guide/oauth-auth-flow.html>>, učitano 21.08.2011.

7. Pretraživanje HTML sadržaja

Dosadašnji dio je većinom bio posvećen dohvaćanju podataka iz XML dokumenata. U ovom poglavlju bit će opisan način dohvaćanja sadržaja iz HTML dokumenata uz pomoć XPath³², jezika koji služi za adresiranje XML i HTML dokumenata.

Notacija XPath jezika koristi putanje za navigaciju kroz hijerarhijsku strukturu XML ili HTML dokumenata³³. Putanje XPath jezika su vrlo slične putanjama koje se koriste kod označavanja URL-a ili datotečnog sustava. Postoji više od 100 ugrađenih funkcija koje služe npr. za manipuliranje znakovima ili uspoređivanje vremena. Također, postoji posebna terminologija koja se temelji na sedam vrsta čvorova: element, atribut, tekst, imenski prostor, instrukcija obrade, komentar, čvor dokumenta. Izrazi koji se koriste za pristup određenim dijelovima dokumenta će biti objašnjeni u sljedećoj tablici, a ostali detalji vezani XPath se mogu naći na stranici službene dokumentacije³⁴.

Tablica 7.1. XPath izrazi za putanju³⁵

Izraz	Opis
<i>nodename</i>	Označava sve čvorove djecu navedenog čvora.
/	Označava sadržaj korijenskog čvora.
//	Označava sve čvorove navedenog čvora bilo gdje se oni nalazili u dokumentu.
.	Označava trenutni čvor.
..	Označava roditelja trenutnog čvora.
@	Označava atribut.

Posebna sintaksa vrijedi kada je u pitanju pretraživanje HTML sadržaja uz pomoć YQL upita. Nakon WHERE klauzule je potrebno navesti URL³⁶ stranice na kojoj se nalazi HTML sadržaj te XPATH kojim se dolazi do točno određenih podataka. Sintaksa je sljedeća:

³² XML Path Language

³³ W3schools.com, XPath Introduction, [Internet], <http://www.w3schools.com/xpath/xpath_intro.asp>, učitano 21.08.2011.

³⁴ W3.org, XML Path Language (XPath) 2.0 (Second Edition), [Internet], <<http://www.w3.org/TR/xpath20/>>, učitano 21.08.2011.

³⁵ W3schools.com, XPath Syntax, [Internet], <http://www.w3schools.com/xpath/xpath_syntax.asp>, učitano 21.08.2011.

³⁶ Uniform Resource Locator

*select * from HTML where URL="xyz" and XPATH='xyz'*

Slijedi primjer koji će ilustrirati kako se koriste gore navedeni izrazi i kako izgleda stvarni upit kod dohvaćanja HTML sadržaja. Pretraživat će se web stranica [net.hr](http://www.net.hr) i dohvaćat će se glavna vijest naslovnice. Upit izgleda ovako:

*select * from html where url="http://www.net.hr/" and
xpath='/html/body/div[2]/div[2]/div/div[3]/div/div/ul/li/div'*

Postavlja se pitanja kako smo došli do putanje koja se navodi kod xpath-a. Najlakši način je koristeći Firebug dodatak za Firefox preglednik kojeg je moguće pronaći na njegovim službenim stranicama ³⁷, a olakšava programerima izradu i analizu web stranica. Otvorimo stranicu u Firefoxu, desni klik na sadržaj kojeg želimo dohvatiti te odaberemo opciju "Pregledaj element".

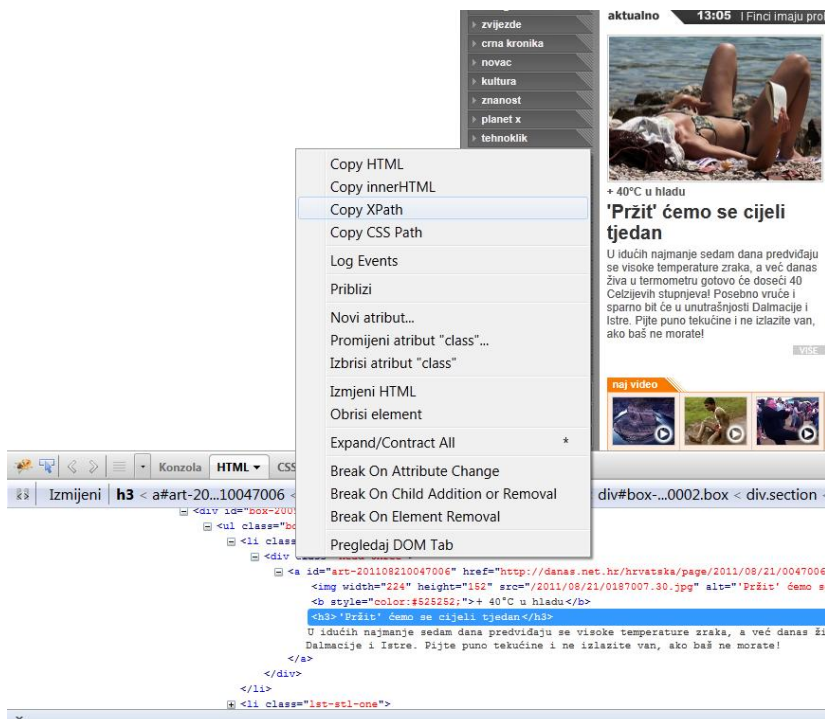


Slika 7.1. Pregledaj element ³⁸

Odabirom opcije "Pregledaj element" otvara se alat Firebug i prikazuje se struktura čitave HTML stranice. Kursor se pozicionira na div element koji sadrži označeni tekst. Desnim klikom na taj div element se otvaraju mnogobrojne opcije od kojih je za dohvaćanje XPath putanje potrebno odabrati opciju "Copy XPath", što se vidi na sljedećoj slici.

³⁷ Firebug, [Internet], <<http://getfirebug.com/>>, učitano 21.08.2011.

³⁸ Net.hr, [Internet], <<http://www.net.hr/>>, učitano 21.08.2011.



Slika 7.2. Copy XPath³⁹

Time smo dohvatili XPath putanju i možemo ju upotrijebiti u YQL upitu. Pokretanjem YQL upita u YQL konzoli kao rezultat dobivamo XML dokument koji kao rezultat sadrži tekst glavne vijesti stranice net.hr. Ovo se primjerice može iskoristiti u kreiranju aplikacije koja prikuplja najvažnije novosti hrvatskih portala.

³⁹ Net.hr, [Internet], < <http://www.net.hr/>>, učitano 21.08.2011.

8. Aplikacija

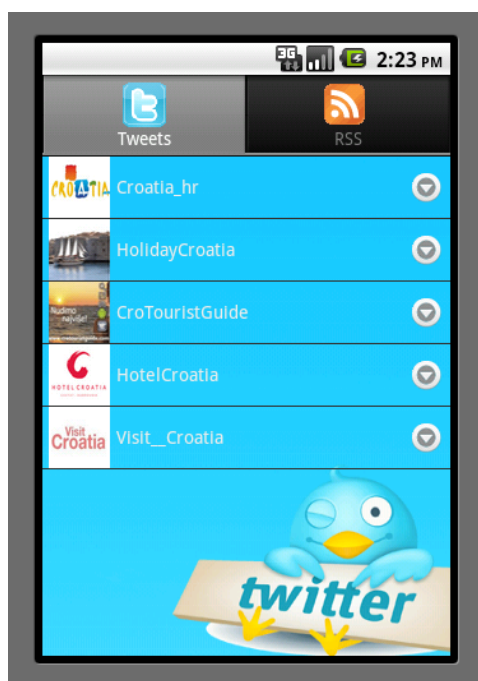
Za razliku od prethodnih poglavlja koja su bila posvećena opisu YQL-a, ovo poglavlje će biti posvećeno konkretnoj primjeni YQL-a u aplikaciji. Kao što je već rečeno, pomoću YQL-a je moguće pretražiti gotovo sav sadržaj na Internetu. Pošto je ljetno vrijeme i vrijeme turizma, odlučeno je da aplikacija bude orijentirana na pronalaženje turističkih vijesti o Hrvatskoj. Točnije, pretraživat će se Twitter te nekoliko hrvatskih portala s turističkim vijestima u obliku RSS-a. Aplikacija je izrađena za Android mobilne uređaje, a izrađena je pomoću alata Titanium Appcelerator, koji omogućava izradu Android i iPhone aplikacija koristeći web tehnologije poput JavaScript, HTML i CSS jezika. Više informacija o Titanium Appceleratoru moguće je pronaći na službenim stranicama ⁴⁰.

8.1.Sučelje

Sučelje aplikacije je vrlo jednostavno. Sastoji se od 2 "taba" pomoću kojih se odabire prikaz Twitter statusa ili RSS vijesti. Svaki od tabova sadrži listu od 5 portala s kojih se dohvaćaju informacije, a svakom od njih je pored njegova imena pridružen i njegov logo. Odabirom jednog od portala prikazuju se njegove vijesti ili Twitter statusi, a u slučaju RSS vijesti klikom na tu vijest je moguće i pogledati originalnu web stranicu na kojoj je ona objavljena. Također, za svaku vijest se navodi i vrijeme objavljivanja. U nastavku slijedi nekoliko slika koje prikazuju sučelje aplikacije.

⁴⁰ <http://www.appcelerator.com/>

Početni ekran aplikacije je predstavljen na sljedećoj slici. Sadrži listu Twitter korisnika čije je Twitter statue moguće dohvatiti.



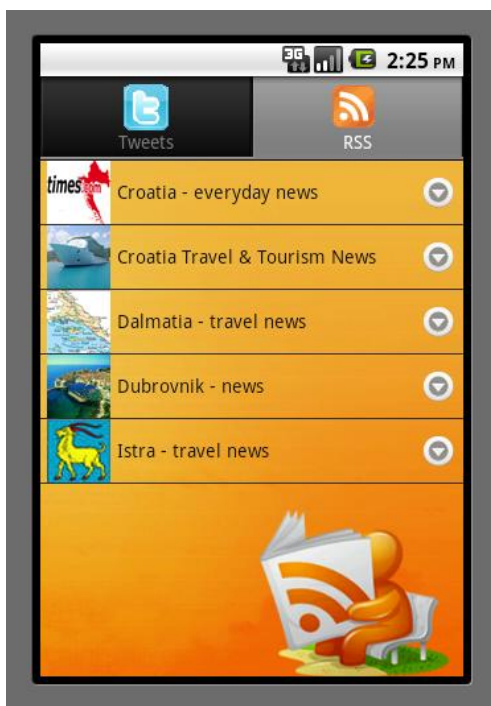
Slika 8.1. Sučelje - Twitter lista

Nakon što se odabere jedan od ponuđenih korisnika, pojavi se sučelje s njegovim Twitter statusima. Primjer takvog sučelja za korisnika CroTouristGuide, koji objavljuje turističke vijesti iz područja Hrvatske, prikazan je na sljedećoj slici.



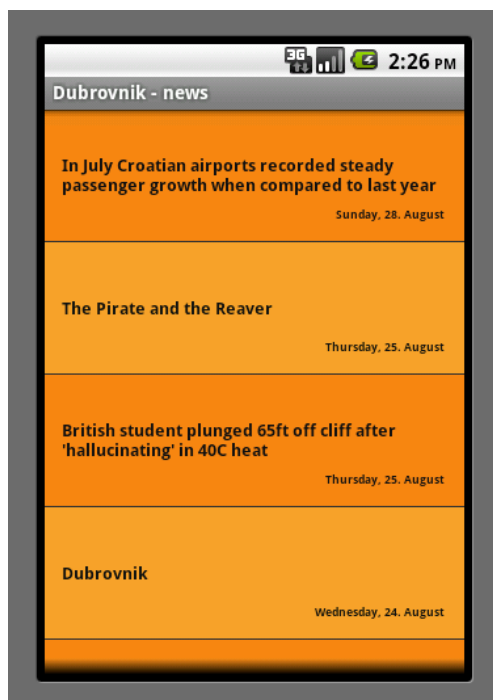
Slika 8.2. CroTouristGuide - primjer Twitter statusa

Slijedi slika na kojoj je predstavljeno sučelje s listom mogućih RSS vijesti. Kao i kod Twitter statusa, svaka od ponuđenih portala sadrži službeni logo.



Slika 8.3. RSS lista

Nakon odabira jedne od opcija, pojavljuje se sučelje s RSS vijestima tog portala zajedno s datumom objave. Primjer RSS vijesti vezanih uz Dubrovnik prikazan je na sljedećoj slici.



Slika 8.4. Dubrovnik News - primjer RSS vijesti

8.2. Kod

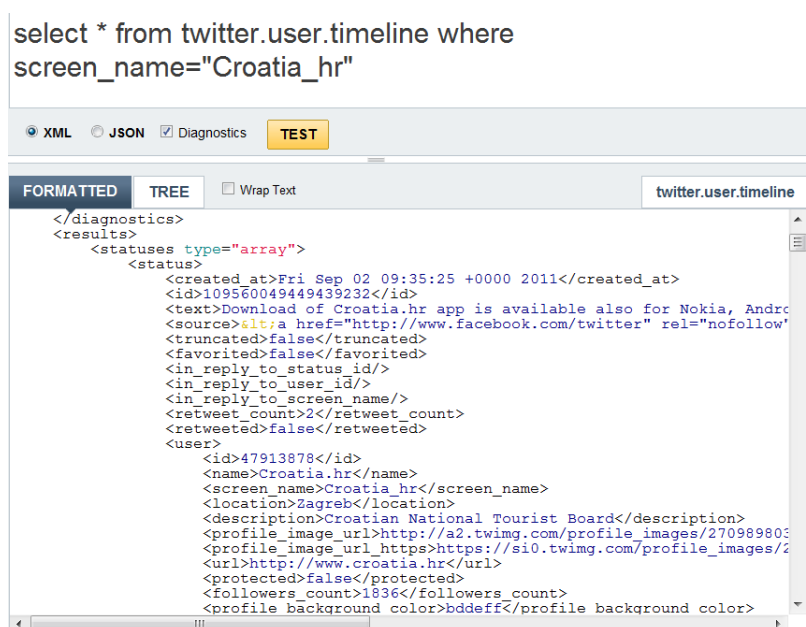
Nakon što je predstavljeno sučelje aplikacije i njezina funkcionalnost, postavlja se pitanje na koji način je ono ostvareno. Kao što je već rečeno, pri izradi aplikacije korišten je Titanium Appcelerator, što znači da su prilikom izrade uglavnom korištene posebne naredbe tog alata poput *Titanium.UI.createTab* te JavaScript jezik. Međutim, pošto je tema ovog rada YQL, naglasak će u ovom poglavlju biti na primjeni YQL u aplikaciji.

Najvažniji dio aplikacije je bio kreiranje upita koji će dohvatiti željene podatke, tj. Twitter statuse i RSS vijesti. U YQL konzoli je pronađena lista Twitter tablica koja sadrži statuse korisnika, a njezino ime je *twitter.user.timeline*. Nad tom tablicom je provedena SELECT naredba te je dodan filter s imenom korisnika od kojega dohvaćamo statuse. Primjer jednog upita za dohvaćanje Twitter statusa izgleda ovako:

```
query = 'select * from twitter.user.timeline where screen_name="Croatia_hr"'
```

Slika 8.5. Primjer Twitter upita

Rezultat upita možemo testirati u YQL konzoli te proučiti strukturu dobivene tablice, kako bi znali na koji način možemo pristupiti željenim podacima.



Slika 8.6. Rezultat Twitter upita

Nakon što smo dobili upit, potrebno ga je iskoristiti u aplikaciji. Za izvršavanje YQL upita Titanium Appcelerator posjeduje posebnu funkciju *Titanium.Yahoo.yql* kojoj se proslijeđuje upit te se kao rezultat dobije XML dokument u obliku JavaScript objekta. Način korištenja te funkcije može se vidjeti na sljedećoj slici:

```
Titanium.Yahoo.yql(query, function(e) {  
    var data = e.data,
```

Slika 8.6. Titanium.Yahoo.yql funkcija

Na temelju gornjeg koda može se vidjeti da varijabla *data* sadrži JavaScript objekt te se dohvaćenim Twitter podacima može pristupiti pomoću Dot sintakse. Primjer pristupa tekstu Twitter statusa može se vidjeti na sljedećoj slici (opcija *text*).

```
var tweetText = Ti.UI.createLabel({  
    text:data.statuses.status[i].text,  
    left:54,  
    top:2,  
    bottom:0,  
    height:'auto',  
    width:236,  
    textAlign:'left',  
    color:'#111',  
  
    font:{fontSize:13}  
});
```

Slika 8.7. Pristup podacima

Gornja slika, osim što prikazuje način na koji se pristupa podacima dohvaćenim YQL upitom, prikazuje i dio dizajniranja sučelja. Može se vidjeti kako se pozicionira mjesto ispisa podataka (opcije *left*, *top*, *bottom*, *width*) te kako se navode opcije izgleda teksta (*height*, *textAlign*, *color*, *font*).

Gornji primjeri reprezentativno prikazuju korištenje YQL u aplikaciji. YQL se koristi na još mnogo mjesta, ali to se uglavnom svodi na primjere vrlo slične navedenima. Ipak, potrebno je još istaknuti i primjer upita koji je korišten za dohvaćanje RSS vijesti, a koji je kasnije pretvoren u JavaScript objekt te mu je pristupano na jednak način kao što je već objašnjeno. Upit glasi:

```
query = "SELECT * FROM rss WHERE url = 'http://www.topix.com/rss/hr/dubrovnik'"
```

Slika 8.8. Primjer RSS upita

Kod ovog upita bitno je naglasiti da se ne navodi ime tablice kojoj pristupamo, već se nakon FROM argumenta navodi da se sadržaj nalazi u RSS formatu. Nakon WHERE argumenta se navodi URL na kojem se nalaze RSS vijesti.

Obrada dobivenih rezultata se odvija na jednak način kao i Twitter statusa. Dobiveni XML dokument se pretvori u JavaScript objekt, a do podataka u njemu se dolazi pomoću Dot sintakse. Kako se dva puta ne bi objašnjavala ista stvar, ovdje će se preskočiti taj dio. Kao prilog ovom radu će biti priložen CD s izvornim kodom aplikacije i izvršnom datotekom te će se tamo moći pogledati cjelokupni kod.

9. Kritički osvrt

Nakon velikog broja sati potrošenih na proučavanje YQL jezika i izradu aplikacije nije lako svoje dojmove sažeti u nekoliko rečenica. Što reći nego da nisam ni slutio kakva je moć YQL jezika. Gotovo je moguće dohvatiti bilo kakav sadržaj koji se nalazi bilo gdje na Internetu, čak i sadržaj društvenih mreža poput Twittera ili Facebooka, a sve to koristeći sintaksu slični SQL sintaksi.

Izenadila me i raširenost jezika iako postoji tek nekoliko godina. Pojedinci kreiraju vlastite tablice s podacima i dijele ih s ostalim korisnicima, što ubrzava izradu aplikacija. Također, iznenadila me i podrška u alatima jer već postoje posebne funkcije koje procesiraju YQL upite, a omogućeno je korištenje YQL upita i kod mobilnih aplikacija.

Naglasio bih i YQL konzolu pomoću koje je moguće kreirati svoje YQL upite i saznati strukturu dohvaćenog dokumenta, što se kasnije može iskoristiti u vlastitim aplikacijama. Također, istaknuo bih da je korisnička podrška vrlo dobra jer postoji službeni vodič⁴¹ u kojem su objašnjeni svi dijelovi YQL jezika, službeni blog⁴² koji donosi novosti vezane uz jezik te forum⁴³ na kojem se mogu pronaći odgovori na različita pitanja vezana uz YQL jezik.

Na kraju mogu reći da sam uživao istraživajući YQL jezik. Shvatio sam njegov potencijal i definitivno ću ga koristiti u svojim budućim aplikacijama, kao i u razvoju aplikacije izrađene za ovaj rad. Svoja iskustva ću prenijeti i svojim kolegama jer je šteta što se tijekom preddiplomskog studija nismo susreli s ovim jezikom.

⁴¹ <http://developer.yahoo.com/yql/>

⁴² <http://www.yqlblog.net/>

⁴³ <http://developer.yahoo.com/forum/YQL>

10. Zaključak

U ovom završnom radu predstavljen je Yahoo Query Language s naglaskom na njegovu upotrebu prilikom pretraživanja društvenog Weba. Navedene su prednosti YQL-a i njegove moguće primjene. Objašnjena je sintaksa YQL jezika i način na koji se filtriraju podaci u upitima. Također, objašnjeni su načini pokretanja YQL naredbi GET metodom i pomoću YQL konzole. Zatim su navedene vrste autorizacije prilikom dohvaćanja privatnih podataka te je prikazan način pretraživanja i dohvaćanja HTML sadržaja YQL upitima.

Gore navedeni pojmovi čine najvažniji dio razumijevanja YQL jezika, ali nikako ne i čitav. Postoji još nekolicina detalja koji nisu spomenuti u ovom radu, a koji bi mogli biti korisni u slučaju da neka osoba profesionalno počne koristiti YQL jezik. Zbog ograničenog prostora takvi detalji nisu pronašli svoje mjesto u ovom radu, ali oni znatiželjniji mogu ih pronaći na službenim stranicama YQL jezika ⁴⁴.

Pošto je YQL vrlo mlad jezik, u budućnosti se očekuje sve veća upotreba i daljnji razvoj jezika. S obzirom na njegov potencijal, smatram da će vrlo brzo veliki broj razvojnih programera shvatiti njegove mogućnosti i početi ih uvelike iskorištavati u svojim aplikacijama. Osobno, spoznao sam njegovu moć prilikom izrade aplikacije te ću i u budućnosti nastaviti raditi na njoj i razvijati ju pomoću YQL upita.

⁴⁴ <http://developer.yahoo.com/yql/>

Literatura

1. Choi J.: YQL: INSERT INTO internet, [Internet], <http://developer.yahoo.com/blogs/ydn/posts/2009/07/yql_insert/>, učitano: 21.08.2011.
2. Firebug, [Internet], <<http://getfirebug.com/>>, učitano 21.08.2011.
3. Net.hr, [Internet], <<http://www.net.hr/>>, učitano 21.08.2011.
4. Smith M.: Two-legged OAuth Example for YQL with PHP, [Internet], <<http://mattsmithblog.wordpress.com/2010/06/15/two-legged-oauth-example-for-yql-with-php/>>, učitano 21.08.2011.
5. Stojanović D.: Yahoo! PULSE, [Internet], <<http://pulse.yahoo.com/y>>, učitano 21.08.2011.
6. Trevor J.: YQL Launched, [Internet], <<http://pcworld.about.net/od/software/services/Yahoo-Extends-YQL-Web-Data-Que.htm>>, učitano: 21.08.2011.
7. W3.org, XML Path Language (XPath) 2.0 (Second Edition), [Internet], <<http://www.w3.org/TR/xpath20/>>, učitano 21.08.2011.
8. W3schools.com, XPath Introduction, [Internet], <http://www.w3schools.com/xpath/xpath_intro.asp>, učitano 21.08.2011.
9. W3schools.com, XPath Syntax, [Internet], <http://www.w3schools.com/xpath/xpath_syntax.asp>, učitano 21.08.2011.
10. Willison S.: Simon Willison's Weblog, [Internet], <<http://simonwillison.net/2009/Jul/8/yql/>>, učitano: 21.08.2011.
11. Yahoo! Developer Network, Yahoo! Query Language (YQL) Guide, [Internet], <http://openhacklondon.pbworks.com/f/yql_guide.pdf>, učitano 21.08.2011.
12. Yahoo! Inc.: YQL Console, [Internet], <<http://developer.yahoo.com/yql/console/#h=show%20tables>>, učitano 21.08.2011.
13. Yahoo! Inc.: YQL Console, [Internet], <http://developer.yahoo.com/yql/console/#h=select%20*%20from%20flickr.photos.info%20where%20photo_id%3D%220123456789%22>, učitano 21.08.2011.
14. Yahoo! Inc.: YQL Console, [Internet], <http://developer.yahoo.com/yql/console/#h=select%20*%20from%20answers.search%20where%20query%3D%22computer%22%20AND%20NumAnswers%20%3E%20%221%22%20OR%20query%3D%22airplane%22%20AND%20NumComments%20%3D%220%22>, učitano 21.08.2011.

15. Yahoo! Inc.: YQL Console, [Internet],
<http://developer.yahoo.com/yql/console/#h=select%20*%20from%20music.track.popular%20where%20releaseYear%3E%3D%222009%22%20LIMIT%203%20OFFSET%205>, učitano 21.08.2011.
16. Yahoo! Inc.: YQL Console, [Internet], <<http://developer.yahoo.com/yql/console/>>, učitano 21.08.2011.
17. Yahoo! Inc.: YQL Console, [Internet], <<http://query.yahooapis.com/v1/public/yql?q=show%20tables&diagnostics=true>>, učitano 21.08.2011.
18. Yahoo! Inc.: YQL Console, [Internet],
<<https://developer.yahoo.com/yql/console/?q=show%20tables#h=desc%20social.profile.status>>, učitano 21.08.2011.
19. Yahoo! Inc.: YQL Console, [Internet], http://developer.yahoo.com/yql/console/#h=select%20*%20from%20music.track.popular%20where%20releaseYear%3E%3D%222009%22>, učitano 21.08.2011.
20. Yahoo! Inc.: YQL Guide: Filtering Query Results (WHERE), [Internet],
<<http://developer.yahoo.com/yql/guide/filters.html>>, učitano 21.08.2011.
21. Yahoo! Inc.: YQL Guide: Filtering Query Results (WHERE), [Internet],
<<http://developer.yahoo.com/yql/guide/filters.html>>, učitano 21.08.2011.
22. Yahoo! Inc.: YQL Guide: Sort and Other Functions, [Internet],
<<http://developer.yahoo.com/yql/guide/sorting.html>>, učitano: 21.08.2011.
23. Yahoo! Inc.: YQL Guide: Syntax of I/U/D, [Internet],
<http://developer.yahoo.com/yql/guide/iud-syntax.html>
24. Yahoo! Inc.: YQL Guide: Updates Tables, [Internet],
<http://developer.yahoo.com/social/rest_api_guide/updates_table.html>, učitano 21.08.2011.
25. Yahoo! Inc.: YQL Guide: Introduction, [Internet],
<<http://developer.yahoo.com/oauth/guide/about.html>>, učitano 21.08.2011.
26. Yahoo! Inc.: YQL Guide: OAuth Authorization Flow, [Internet],
<<http://developer.yahoo.com/oauth/guide/oauth-auth-flow.html>>, učitano 21.08.2011.
27. Yahoo! Inc.: YQL Guide: Private Data v. Public Data, [Internet],
<http://developer.yahoo.com/oauth/guide/oauth-private_public_data.html#>, učitano 21.08.2011.
28. Yahoo! Inc.: YQL Social, [Internet],
<<https://developer.apps.yahoo.com/projects/JeHM4n6s>>, učitano 21.08.2011.