

Tallinna Ülikool
Informaatika instituut

Google AJAX search API

Seminaritöö

Autor: Rome Tikas
Juhendaja: Jaagup Kippar
Kaitmisele lubatud:

Tallinn 2009

SISUKORD

SISSEJUHATUS.....	3
1.TEOREETILINE TAUST.....	4
1.1 Google (SOAP) Search API.....	4
1.2 Google AJAX search API.....	4
1.3 Ajax API võtme kasulikkus.....	5
1.4 Google otsingud.....	5
1.5 Google AJAX search API viisard.....	7
1.6 Google AJAX Search API't kasutavad saidid.....	8
2.1"Hello, World".....	9
2.1.1 Otsingu script.....	9
2.1.2 Harjutusi õpilasele.....	12
2.1.3 Kokkuvõte „Hello, World“ otsingule.....	13
2.2 Kaardi viisard.....	13
2.2.1 Probleemid kaardi viisardiga.....	13
2.2.2 Otsingu tulemus	13
2.2.3 Kokkuvõte kaardi viisardile.....	14
KOKKUVÕTE.....	15
.....	15
KASUTATUD KIRJANDUS.....	16
LISAD.....	17
Lisa 1. Google poolt loodud USA presidendi kandidaatide leht.....	18
.....	18
Lisa 2. Lähtekood "Hello, World" jaoks.....	19
.....	19
Lisa 3. Kaardi viisardiga saadud lähtekood.....	20

SISSEJUHATUS

Iga veebilehekülg on suuremal või vähemal määral sõltuv külastajate arvust. Mida suurem on külastajate arv, seda rohkem on antud leheküljel võimalus teenida oma eesmärgi. Samas on oluline, et lehekülgedel olev info ei muutu külastaja jaoks tüütuks, mis omakorda sunnib neid muid võimalusi infot leida. Et oma külastajaid hoida on paljud veebilehtede omanikud hakanud kasutama Google toodet Google AJAX search API. Kellele on Google AJAX search API kasulik? Kasu on kolmepoolne. Võidab Google, kui tema otsinguid kasutatakse. Võidab lehe omanik, kui kliendid külastavad nende lehte sageli. Võidab ka klient, kes saab ilma lehelt lahkumata vajaliku info jätte. Samas kui Google AJAX search API on halvasti seadistatud, võib see kahjustada lehekülje mainet ning pärssida populaarsust.

Käesolev töö eesmärgiks on tutvustada õppivale veebiprogrammeerijale, mis on Google AJAX search API ja kuidas seda efektiivselt rakendada.

Töö on üles ehitatud kahe osana, mis jaguneb omakorda alateemadeks. Esimeses osas kirjeldatakse, mis on Google AJAX search API, kuidas on teised veebilehekülgede omanikud seda kasutusele võtnud, antakse ülevaade enimlevinutest otsingu võimalustest. Teises osas tutvustatakse, kuidas on võimalik saada tööle Google AJAX search API, ilma suurema vaevata.

1. TEOREETILINE TAUST

Nii nagu on igal inimesel teatud ainuomane tunnusjoonte komplekt, läbi mille on võimalik teda määratleda ja kirjeldada, on ka igal veebilehel oma ainulaadne identiteet, mis väljendub kõiges, mida leheküljel kuvatakse, mida omatakse ja mida väärtustatakse.

"What's at your fingertips here is a way to build entirely new search environments, and base them on the Google index." [Tõlge: „Sinu näppude all, siin on uus viis ehitada täiesti uusi otsingukeskkondi, mis baseeruvad Google indeksil“ Eric Enge <http://www.stonetemple.com/articles/google-ajax-search-api.shtml>].

See lause võtaks kokku kogu otsingusüsteemi võimsuse, nagu ta praegusel hetkel on.

1.1 Google (SOAP) Search API

Google (SOAP) Search API, mida peetakse Google AJAX search API eelaseks, loodi arendajatele ja uurijatele, et nad saaksid kasutada Google otsingut oma programmides. Arendajad said kasutada kogu Google andmebaasi, et leida õiget vastust miljardite veebilehtede seest, mis oli struktureeritud. Nad said kasutada Google vahemälu ja kontrollida õigekirja. Et toodet kasutada, oli vaja hankida toote jaoks võti, mis on enamikel Google toodetele vajalik.

1.2 Google AJAX search API

5. detsembril 2006 lõpetas Google võtmete jagamise sellele tootele, siiski kellel oli võti saab seda kasutada mõõndustega, teades, et rohkem uuendusi neile juurde ei tule.

Selle asemel sai täieliku toetuse ennem testimisel olnud Google AJAX search API. Kasutusele tulnud toode oli mõeldud ainult browseri põhiseks JavaScript kasutamiseks.

Oli veel aeg, kui SOAP polnud surnud, kuna paljud ei saanud ega tahtnud kasutada JavaScript põhilist otsingut. Nemad jäid truuks SOAP Search API-le.

Kogu asi hakkas muutuma aprillis 2008, kui Google lisas otsingule RESTful täienduse, et toetada flashi ja teisi mitte JavaScripti keskkondi. Hetkel on need täiendused arendusjärgus ja pea iga päev tuleb uuendusi. Leitakse turvaauke ja samas ka uusi lahendusi otsingu kiiremaks muutmiseks. Aina enam arendajaid kasutavad just uuendatud Google toodet, kuna nad tahavad konkurentsisis püsida.

1.3 Ajax API võtme kasulikkus

Google laseb sul kasutada enda otsingut oma veebilehel, samas olles üks vähestest toodetest Googli poolt, mis ei vaja tegevuseks võtit. Samas kasutades võtit, saad sa oma veebilehele täieliku ülevaate. Võtmega otsingu vea ilmnemisel on sul olemas Google meeskonna toetus, kes püüab sind igati aidata, ilma võtmeta seda ei tehta.

Võtit saab registreerida, selleks on sul vaja minna veebilehele <http://code.google.com/apis/ajaxsearch/signup.html>. Soovitan läbi lugeda Google AJAX search API kasutamise tingimused ja nendega nõustuda. Seejärel tuleb sisestada oma veebilehekülje nimi, vaja on sisestada ainult pea aadress ja Google laseb seda kasutada kõikidel sinu alamlehekülgedel. Ise kasutasin näites <http://www.tlu.ee>. Saadud võti on kõigile sama lehekülje kasutajale sama. Juhul kui oled ennem võtit registreerinud, siis on see tegevus väga lihtne ja võtab aega 5-10 sekundit.

1.4 Google otsingud

Google on praeguseks hetkeks välja töötanud 8 erinevat otsingu kriteeriumit. Ehk kogu, mille järgi on võimalik andmebaasile päringuid esitada. JavaScriptis toimub nendesse kogudesse päringu tegemine väga lihtsalt.

Nendeks kogudeks on

** LocalSearch*

Funktsioon Google MAP'le, saad otsida kaardilt antud kohta. Pean mainima, et otsingule „Tallinn University“ on vastuse leidmine raske, kuna Google pakub vastuseks „Tallinn University of Technology“. Otsingule „Tallinna Ülikool“ saab ilusasti kaardi pealt kõigi Narva mnt majade paiknemise.

** WebSearch*

Kõige tavalisem Google otsing, sama võib leida www.google.com. Siinkohal ei hakka arvustama või kiitma Google otsingu üldist objektiivsust, sellest saab eraldi töö kirjutada.

** VideoSearch*

Saab otsida videoid, peamiselt Google enda video veebilehelt ja samuti ka <http://www.youtube.com>. Otsitakse läbi kogu Google andmebaas videodest ja rõõmustav on, et esimese vastena otsingule „Tallinna Ülikool“ meie enda suhtekorralduse tehtud video „Ülikool on....“

** BlogSearch*

Otsib enamusest maailma juhtivatest veebikolumnidest vasteid sinu otsingule. Tihti võivad otsingutulemused olla mõni päev vanad, kuna maailmas on palju veebikolumne, mida google ei indekseeri iga päev.

** NewsSearch*

Otsib uudiseid Google enda uudiste portaalist <http://news.google.com/>. Mis omakorda monitoorib uudiste portaale pidevalt. Seega otsingust leiab suhteliselt uued uudised kogu aeg.

** ImageSearch*

Kasutab piltide otsinguks täpselt sama süsteemi, mis <http://images.google.ee/>. Pildid kuvatakse ilusti ka otsingu aknasse.

** BookSearch*

Selle otsingu tulemus suunab ümber kasutaja lehele <http://books.google.com/>. Jaanuar 2009 seisuga oli sinna skanneeritud üle 7 miljoni raamatu. Hetkel on seal saadaval enamasti inglise keelsed raamatud, samas on olemas enamike raamatute andmed. Näitena kasutasin A.Hitleri „Mein Kampf“, üllatav oli, et seal oli isegi väike ülevaade raamatust, siiski ei olnud tegemist skanneeritud variandiga.

** PatentSearch*

Järjekordselt üks Ameerikast lähtuv Google rakendus. Nimelt otsib see Ameerika Patendiametis (United States Patent and Trademark Office) registreeritud tooteid. Kahju tõdeda, et Tallinna Ülikoolil ei ole seal midagi patenteeritud.

1.5 Google AJAX search API viisard

Google on teinud veebilehtede omanikele elu suhteliselt kergeks. Nimelt on nad valmistanud 7 viisardit enim kasutatavatele otsingutele.

Kaardi otsingu viisard - Laseb lõppkasutajal otsida Google Mapsist ilma, et leheküljelt lahkuksid

Video viisard - Google videode kogum(omaniku poolt pre-otsitud), mida kasutatakse enamjaolt blogides

Video otsingu viisard - Laseb otsida Google videodest ja neid vaadata, ilma leheküljelt lahkumata

Uudiste viisard (horisontaalselt) – Annab ülevaate hetke uudistest antud otsinguga, kuvab neid üksikult

Uudiste viisard (vertikaalselt) – Annab ülevaate uudistes kas paremal või vasakul servas, ülevalt alla. Kuvab rohkem kui ühte tulemust (juhul kui on).

Blogi uudiste viisard – Kuvab blogi uuemaid uudiseid.

Raamatu viisard - Kuvab raamatud veebilehelt <http://books.google.com/>.

Kõik viisardid on üsna lihtsad ja mugavad. Neid võib enda nägemise järgi ka muuta, kui oled piisavalt pädev. Hiljem näitan ka kuidas ühte viisardit seadistada, mis on selle head ja vead.

1.6 Google AJAX Search API't kasutavad saidid

Esiteks veebileht, mis asub aadressil <http://ajaxsearch.typepad.com/>. Seal lehel on olemas kolm Google AJAX search API otsingut : tavaline otsing, Google video otsing ja Google kaardi otsing. Õpetatakse seal just kuidas teha blogijatele erinevaid otsinguid.

Miks just see lehekülg?

Kuna minu arvates see on üks parimaid näiteid, kui võimas on see Google funktsioon. Videoid saab otsida mitmete parameetrite järgi. Kaardile saab suunata blogi teksti seest asuvast viitest ja mis peamine otsing töötab kiiresti ja täpselt.

Kaardile suunamise funktsioonist ka lähemalt.

Tuleb luua hüperlink, mis on JavaScriptina ja seal tuleb kasutada käsku `doMapSearch('otsingu tulemus');`. Kui sa otsid näiteks Tartut, siis sinu kood peaks välja nägema nii `Tartu`.

Töötab lihtsalt ja on võimalus teha väga palju huvitavat.

Teise näitena toon <http://www.google.com/2008election/> (Lisa 1.praeguseks hetkeks on see lehekülg suletud). Kunagi ennem, kui alustasin seminaritöö kirjutamist sattusin sinna lehele ja imestasin, kuidas saab sellist lehekülge üldse teha. Kogu veebileht on minimalistlik ja paigutatud lehe keskele. Samas on olemas enamus kandidaate ja info nende kohta on alati uuendatud. Toona polnud mul aimugi mis on Google AJAX search API. Nüüd kui tean juba kuidas see toimib, siis tean ka, et seal lehel ei ole midagi võimatud vaid väga targalt kasutatud just Google oma enese toodangu vilju. Rohkem võite selle veebilehe tegemisest lugeda ja kuna originaal lehte enam ei eksisteeri siis ka sellel lehel olevaid funktsioone proovida <http://googleajaxsearchapi.blogspot.com/2008/08/howd-they-do-that.html>.

2. PRAKTILINE TEOSTUS

Kirjeldan antud peatükis läbiviidud praktilise teostuse aluseid, metoodikat, väljundite ülevaadet. Näitena on toodud kahe otsingu kood. Üks on lihtne „Hello, World“ näide ja teine on kaardi viisard.

2.1 "Hello, World"

Lihtsaim viis alustada õpingutega on kõigile tuttav "Hello, World". Ka siin saab samasuguseid paralleele tuua, aga siiski see on natuke keerulisem. Otsingu lähtekoodi (Lisa 2.) sain koos võtmega, natuke on seda modifitseeritud. Otsinguks sõnaks kasutasin "Tallinn University".

2.1.1 Otsingu script

Kogu script on ehitatud *Hypertext Markup Language* (HTML) ja JavaScript koodis. Javascripti osa ehk otsingu mootor on ülesse ehitatud HEAD-i ja väljund harjumuspäraselt BODY-sse.

Alguses määrame ära dokumendi tüübi:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Järgmisena tasub välja tuua, et kasutame xhtml'i:

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

Edasi liigume HEAD'i, kus on nagu mainitud on üles ehitatud otsingumootori tuum.

Kirjutame <head> ja </head>, nende vahele hakkame siis JavaScripti ehitama.

Enne veel tuleb määrata leheküljele tiitel ja HTML standard:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
```

```
<title>Hello World - Google AJAX Search API Näidis</title>
```

Nüüd alustame siis põhilisega, ehk otsingumootoriga:

```
<script src="http://www.google.com/jsapi" type="text/javascript"></script>
```

Lihnte tekst see, aga väga oluline. Nimelt kaasasime just Google AJAX seach API lähtekoodi.

Lähtekood olemas, lähme edasi ja lisame rea:

```
google.load('search', '1');
```

See selgitab Google serverile millist tema API-t kasutada tahame, hetkel on selleks „*search*“. Teine parameeter '1' on seal selle API versiooni number. Kusjuures iga Google API juures tuleb määrata ära ka tema versioon. Piisab teadmisest, et meie otsingu API on versioon 1, sest Google automaatselt pakub uusima 1 versiooni. Näiteks kui me paneksime versiooniks 1.1, siis Google annaks meile 1.1 kõige uuema versiooni. Vahel on sellest kasu, kui on teada, et piisab ainult mõnest teatud versioonist.

Teades mida me soovime, tahab server ka teada, mis otsinguid me serverist teha tahame.

Selleks käivitame funktsiooni:

```
function OnLoad() { }
```

Funktsiooni sees esimeseks loome otsingu kontrolli:

```
var searchControl = new google.search.SearchControl();
```

Edasi lisame just need otsingu tulemused, mida soovime kuvada. Vaata „Google otsingud“:

```
searchControl.addSearcher(new google.search.WebSearch());
```

Antud otsing teostab siis *WebSearch*'i, selle asemele või samasugune uus rida luues, saab kasutada teisi otsinguid. Minu näites on kasutatud veel peale selle otsingu *VideoSearch*, *BlogSearch*, *NewsSearch*, *ImageSearch*.

Järgnev rida ütleb otsingule, et ka ennast kuvab ja määrab ära tema ID, kuna tegemist on erinevate otsingutega, siis on valitud ka ID'ks „searchcontrol“. Kui lehel on rohkem elemente, siis tasub ka hoolikamalt mõelda, mida just teatud asja ID'ks määrata.

```
searchControl.draw(document.getElementById("searchcontrol"));
```

Lõpuni on veel natukene jäänud. Tuleb veel lisadada otsingu parameeter, ehk mida me otsida tahame:

```
searchControl.execute("");
```

Kui veebilehe omanik tahab võib ta teha ka eelotsingu kasutaja jaoks. Ütleme, et tegemist on mingi mängu fänni-leheküljega, siis võib jutumärkide vahele kirjutada selle mängu nime ja otsing kuvab külastajale kohe info Googlest selle mängu kohta.

Aeg lõpetada ka script, lisame rea:

```
</script>
```

Olles JavaScriptiga valmis, lähme HEAD'ist välja ja loome BODY:

```
<body>... </body>
```

Punktiiri asemele kirjutatakse veebilehtedel üldjuhul veebi kujundus. Seda teeme meiegi, lisades ennem ära märgistatud elemendi „searchcontrol“:

```
<div id="searchcontrol">Loading</div>
```

„Loading“ on seal näitena toodud, juhul kui interneti ühendus on aeglane või serveriga on mingi probleem. Põhimõtteliselt võib sinna teksti, pildi või mõne muu vajaliku rakenduse lisada.

Lõpetuseks peame sulgema HTML'i, selleks lisame viimase rea:

```
</html>
```

Nii ongi meie väike otsingumootor olemas (joonis 1).

Töötav lähtekood on nähtav (vt Lisa 2).

Tallinn University Search
powered by Google™

▼ Web 

Tallinn University

36th EUCEN Conference “University Lifelong Learning: Synergy between partners”.

06-11-2008 ... Tallinn University, Narva mnt 25, 10120 Tallinn, ph 6409 101, ...

www.tlu.ee

▶ Video (4) 

▶ Blog (4) 

▶ News (4) 

▼ Image 



Tallinn University
180 x 244

www.euroeducatio



Tango Tallinn
682 x 512

www.tango-
.

Joonis 1. „Hello, World“ otsingu „Tallinn University“ tulemuse kuvand

2.1.2 Harjutusi õpilasele

1. Loo otsingud kõigist Google otsingutest.
2. Otsi ainult pilte
3. „Loading“ teksti asemel lisa Google logo
4. Tee veebileht, kus kasutajad saavad ise valida, mis otsinguid nad kasutada tahavad

2.1.3 Kokkuvõte „Hello, World“ otsingule

Üldiselt on sellist otsingut väga lihtne seda otsingut oma veebilehele sobitada. Väikese vaeva juures saab juba arvestatava tööriista. Tuleks kindlasti lugeda veel natuke manuaale ja õpetusi ja selle otsinguga saab teha imesid.

2.2 Kaardi viisard

Järgminena vaatleme, kuidas saada tööle viisard. (vt Lisa 3).

Viisardiks on valitud kaardi viisard. Selle võib leida aadressilt: <http://www.google.com/uds/solutions/wizards/mapsearch.html>.

2.2.1 Probleemid kaardi viisardiga

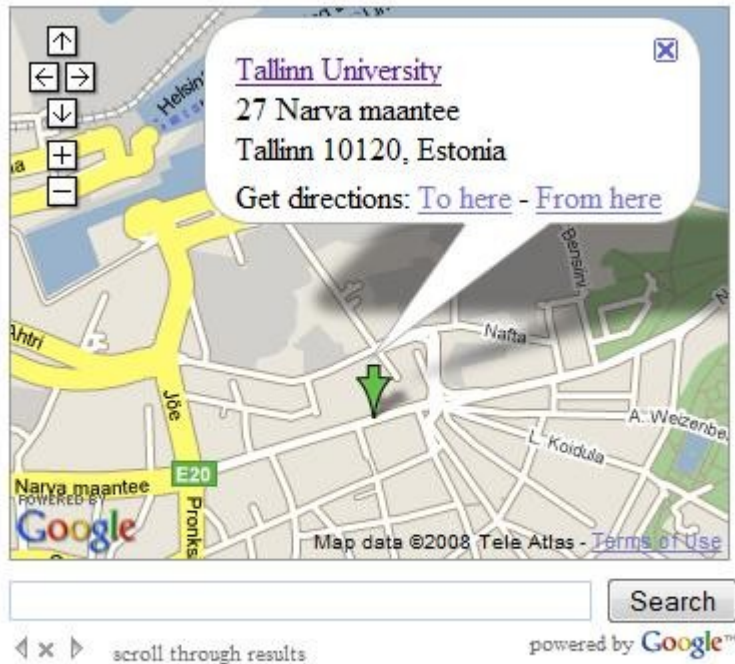
Alguses tundub kõik väga kena. Saab valida kaardi kõrgust ja laiust ja samuti kui kõrgelt soovitakse kaarti näha. "*Center location name*" tuleb panna koha nimi, mida soovid kaardil kuvada, kusjuures sinna võib panna suvalise nime. "*Center location link URL*" tekib esimene raskus, nimelt peab url täielikult välja kirjutama www.tlu.ee ei tööta, õige on <http://www.tlu.ee> "*Center location address*" tekib teine probleem. Kuna Google on enamjaolt USA põhine, siis ka siin tuleb kirjutada kaardilt otsitav aadress nagu USA-s. Et < kaardil Tallinna Ülikooli peamaja on vaja kirjutada sinna kasti " 27 Narva maantee, Tallinn" muud moodi soovitud tulemust ei suutnud leida.

Siis tekib järsku veel probleeme, nimelt kasutab see viisard peale Google AJAX search API ka Google maps API-t. Ja need mõlemad vajavad tasuta API võtit, mis on ühendatud Google Kontoga ja sinu lehekülje URL-ga ja kasutades neid API-si pead sa nõustuma mõlema kasutustingimustega.

Järjekordselt kasutan meie enda ülikooli veebilehte võtme saamiseks <http://www.tlu.ee>.

2.2.2 Otsingu tulemus

Saan otsingu JavaScripti ja väikese lõigu HTML koodi, ehk siis väljundi otsingule. Seal on hästi ära selgitatud, mis kus kasutada. Samuti jääb silma, et otsingule kui ka Mapsile on oma võti genereeritud. Nagu ennem mainitud on siis ka www.tlu.ee võti sama, mis eelnevates kohtades saadud.



Joonis 2. Kaardi viisardi tulemuse kuvand

2.2.3 Kokkuvõtte kaardi viisardile

Viisard on hea ju üsnagi lihtne, samas tekitab ta probleeme oma MAP-si osaga, kuna tegemist on teistmoodi kultuurilise otsinguga. Seadistamisega sain hakkama 2 minutiga (kogenenumad kasutajad teevad seda kindlasti kiiremini).

Kokkuvõte

Siinse töö eesmärgiks oli õppivale veebiehitajale uute ideede juurutamine nii, et just nende veebilehed oleksid külastajate seas populaarsed.

Tahtes muuta veebilehti efektiivsemaks, on vaja aru saada külastajate vajadustest kaasaja ühiskonnas. Vajadustele järgneb alati motiiv, stiimul, mis ergutab käitumist. Nii on veebilehtede omanikel võimalus jälgida hetkel trende ja kuvada just seda oma veebilehel läbi Google AJAX search API, mis just külastajatele sobib (videod, kaardid, uudised). Saab olla suhteliselt kindel, et kui külastajat huvitavad uued naljakad videod näiteks www.youtube.com. siis lehekülg, mis neid pakub on potentsiaalseks sihtmärgiks.

Teatud lehekülgede pidevast külastamisest võib järeldada külastaja positiivset hoiakut antud lehekülje suhtes. Sellest tulenevalt võib eelada, et ka sarnaste sisuelementidega teised leheküljed vastavad külastajate ootustele. Tänu tekkinud hoiakutele on võimalik ka kergema vaevaga kaardistada külastajate vajadused ja neid ka muuta.

Soovitan kõigil veebiprogrammeerijatel, kes tahavad oma veebilehel kasutada mingit otsingut mujalt veebilehekülgedelt kasutada just Google AJAX search API-t. Tema kasutamine on lihtne ja uusi võimalusi lisandub pea iga nädal.

Kasutatud kirjandus

1. Google AJAX Search API. 14.02.2009 <http://code.google.com/apis/ajaxsearch/>
2. Google AJAX Search API BLOG 25.10.2008 <http://code.google.com/apis/ajaxsearch/signup.html>
3. Google SOAP Search API (Beta) 15.02.2009
<http://code.google.com/apis/soapsearch/>
4. Google AJAX Search API Playground 16.02.2009 <http://ajaxsearch.typepad.com/>
5. <http://groups.google.com/group/Google-AJAX-Search-API>
6. Querycat 15.02.2009
<http://www.querycat.com/faq/b05c497857f53e3c3d7acc022386e459>
7. Software As She's Developed 16.02.2009 <http://softwareas.com/google-gadget-api-for-ajax-developers>
8. Eric Enge Stonetemple 19.10.2008 <http://www.stonetemple.com/articles/google-ajax-search-api.shtml>

LISAD

Lisa 1. Google poolt loodud USA presidendi kandidaatide leht

Baldwin	Barr	McCain	McKinney	Nader	Obama
----------------	-------------	---------------	-----------------	--------------	--------------



Chuck Baldwin

baldwin'08 - news - youtube - facebook

Election News	YouTube	Blog Posts	Quotes	News by State
----------------------	----------------	-------------------	---------------	----------------------

[Time and place set for debate between Bob Barr, Chuck Baldwin and ...](#)
Independent Political Report, 12 hours ago
On October 30, Ralph Nader, Bob Barr, and Chuck Baldwin will debate each other at the Cleveland (Ohio) City Club, between 4:30 pm and 5:30 pm. ...

[LETTER: Baldwin is a third party choice](#)
Superior Telegram, Oct 29, 2008
... for President and head of the new Campaign for Liberty, endorses Constitution Party candidate Chuck Baldwin, Baptist Pastor, on a Baldwin/Castle ticket. ...

[Staff photos/Katie Yantis](#)
The Evening Leader, 13 hours ago
Students also learned about the lesser known presidential candidates and their running mates, including Chuck Baldwin with the Constitution Party. ...

[I Would Vote for Chuck Baldwin, Before I'd Vote for Barack Obama](#)
OpEdNews, Oct 28, 2008
Chuck Baldwin would be better than Barack Obama for the future of the country. Hopefully he wouldn't be able to get the worst parts of his agenda ...

1 [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [More News »](#)

[+ Add to Google](#)

Lisa 2. Lähtekood "Hello, World" jaoks

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
6     <title>Hello World - Google AJAX Search API Näidis</title>
7     <script src="http://www.google.com/jsapi"
8       type="text/javascript"></script>
9     <script language="JavaScript" type="text/javascript">
10    <![CDATA[
11    google.load('search', '1');
12
13    function OnLoad() {
14      // Loob otsingu kontrolli
15      var searchControl = new google.search.SearchControl();
16
17      // Lisab otsingu tulemused
18
19      searchControl.addSearcher(new google.search.WebSearch());
20      searchControl.addSearcher(new google.search.VideoSearch());
21      searchControl.addSearcher(new google.search.BlogSearch());
22      searchControl.addSearcher(new google.search.NewsSearch());
23      searchControl.addSearcher(new google.search.ImageSearch());
24
25
26
27      // Käseb otsingul ennast kuvada ja kuhu ennast kinnitada
28
29      searchControl.draw(document.getElementById("searchcontrol"));
30
31      // Mida otsida
32      searchControl.execute("");
33    }
34    google.setOnLoadCallback(OnLoad);
35
36    </script>
37  </head>
38  <body>
39    <div id="searchcontrol">Loading</div>
40  </body>
41 </html>
```

Lisa 3. Kaardi viisardiga saadud lähtekood

```
1 <!-- ++Begin Map Search Control Wizard Generated Code++ -->
2 <!--
3 // Created with a Google AJAX Search Wizard
4 // http://code.google.com/apis/ajaxsearch/wizards.html
5 -->
6
7 <!--
8 // The Following div element will end up holding the map search control.
9 // You can place this anywhere on your page
10 -->
11 <div id="mapsearch">
12   <span style="color:#676767;font-size:11px;margin:10px;padding:4px;">Loading...</span>
13 </div>
14 <!-- Maps Api, Ajax Search Api and Stylesheet
15 // Note: If you are already using the Maps API then do not include it again
16 //       If you are already using the AJAX Search API, then do not include it
17 //       or its stylesheet again
18 //
19 // The Key Embedded in the following script tags is designed to work with
20 // the following site:
21 // http://www.tlu.ee
22 -->
23 <script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAATiIh3QRbRYw9dELv_
24   HB3MhRxedIj_hRAULd6lsTtNfCc75PKerQpYkvklquvGcfvDpVjd6dldLme9A"
25   type="text/javascript"></script>
26 <script src="http://www.google.com/uds/api?file=uds.js&v=1.0&source=uds-msw&key=ABQIAAAATiIh3QRbRYw9dELv
27   _HB3MhRxedIj_hRAULd6lsTtNfCc75PKerQpYkvklquvGcfvDpVjd6dldLme9A"
28   type="text/javascript"></script>
29 <style type="text/css">
30   @import url("http://www.google.com/uds/css/gsearch.css");
31 </style>
32
33 <!-- Map Search Control and Stylesheet -->
34 <script type="text/javascript">
35   window._uds_msw_dontorepair = true;
36 </script>
37 <script src="http://www.google.com/uds/solutions/mapsearch/gsmsearch.js?mode=new"
38   type="text/javascript"></script>
39 <style type="text/css">
40   @import url("http://www.google.com/uds/solutions/mapsearch/gsmsearch.css");
41 </style>
42
43 <style type="text/css">
44   .gmsc-mapDiv {
45     height : 275px;
```

```
46     }
47
48     .gsmc-idleMapDiv {
49         height : 275px;
50     }
51
52     #mapsearch {
53         width : 365px;
54         margin: 10px;
55         padding: 4px;
56     }
57 </style>
58 <script type="text/javascript">
59     function LoadMapSearchControl() {
60
61         var options = {
62             zoomControl : GSmapSearchControl.ZOOM_CONTROL_ENABLE_ALL,
63             title : "Tallinn University",
64             url : "http://www.tlu.ee",
65             idleMapZoom : GSmapSearchControl.ACTIVE_MAP_ZOOM,
66             activeMapZoom : GSmapSearchControl.ACTIVE_MAP_ZOOM
67         }
68
69         new GSmapSearchControl(
70             document.getElementById("mapsearch"),
71             "27 Narva maantee, Tallinn",
72             options
73         );
74
75     }
76     // arrange for this function to be called during body.onload
77     // event processing
78     GSearch.setOnLoadCallback(LoadMapSearchControl);
79 </script>
80 <!-- ++End Map Search Control Wizard Generated Code++ -->
81
```