

## Contents

KLIENDIKIHI OPTIMEERIMINE / GOOGLE PAGESPEED .....	1
<i>Enne:</i> .....	1
Pärast:.....	2
HTML / CSS / JAVASCRIPT .....	2
HTML.....	2
CSS.....	3
JavaScript.....	3
JMeter jõudlustest .....	4
Setup.....	4
Konsooli tulemused.....	4
Seadistus .....	4
Tulemused.....	4

---

## KLIENDIKIHI OPTIMEERIMINE / GOOGLE PAGESPEED

### *Enne:*

#### **Failide laadimine**

avalehe avamisel laetakse 18 .js faili, kusjuures tegelikult laeti 9 erinevat faili, igaüht kaks korda. Lisaks sellele, et faile laeti topelt, oli suur hulk neist tegelikult avalehel veel ebavajalikud.

#### **JavaScripti parsemine**

PageSpeed teatas, et lehe laadimisel parsetakse 825KiB JavaScript koodi. Kusjuures ka siin käiakse kõik js failid läbi kaks korda, kehva rakenduse arhitektuuri tõttu.

#### **Muudatused:**

eemaldatud on paljud <script> tagid index.html-ist mis asjata skripte avalehe laadimisele kaasasid. Ühel juhul liideti kaks .js faili, kuna üks neist koosnes vaid kahest lühidast funktsioonist, ning teine kasutas just neid funktsioone, muud failid neid funktsioone üldse ei kasutanud.

Optimeeritud on ka rakenduse navigatsiooniloogikat, et leiaks aset vähem valel/liiga varajasel hetkel failide laadimist.

## Pärast:

### ***Failide laadimine***

Lehe alglaadimine hõlmab nüüd nelja .js faili laadimist, enam kui 3x vähem varasemast.

### ***JavaScripti parsemine***

Nüüd teatab PageSpeed, et parsetakse 88.2KiB JavaScript'i, mis on pea 10x vähem algseisust.

Järgitud on ka PageSpeed'i soovitusi, lisada staatilistele ressurssidele HTML header Vary: Accept-Encoding.

Lehtedel, mis kasutavad Google Maps API't või Charts API't

PageSpeed soovib näiteks ka minimiseerida koodi, ent järgisime seda vaid väliste teekide puhul, enda kirjutatud koodi jätsime loetavuse huvides minimiseerimata).

Põhiliselt jääb silma PageSpeedi järgmine soovitus: „Leverage browser caching“, ta soovib .js ja .css failidele pikemat „värskuseluiga“ kui 10 minutit. Otsustasime aga mitte seda suurendada, sest rakendus on hetkel veel arenemas, PageSpeedi soovitatud 7 –päevane eluiga ei tundu mõistlik.

Kohalikus serveris FireFox pluginiga mõõdetuna:

Alamleht	Skoor
index	84
Tulemused/piirkonniti	80
Tulemused/parteidekaupa	89
kandideerimine	83
haaletamine	83
Kandidaadid leht, konkreetse querystringiga	88
KESKMINE:	84,5

## HTML / CSS / JAVASCRIPT

### HTML

Puudu olid html img tagides *alt* atribuudid. Leidus ka mõni html lõpetav html-tag, mille jaoks ei leidnud aga alustavat tagi.

Channel.html failis puudus isegi dokumendi tüübi deklaratsioon.

Mainitud puudujääkide lahendus triviaalne.

## CSS

Leidus üks lõpetamata `{ }` plokk.

Veateateid, mida raporteeriti väliste teekide `.css` failides (`jquery-ui.css`), ignoreerisime.

Hoiatuse ...

**Property -ms-box-sizing is an unknown vendor extension**  
**Property -moz-box-sizing is an unknown vendor extension**  
**Property -webkit-box-sizing is an unknown vendor extension**  
**Property -ms-box-sizing is an unknown vendor extension**  
**Property -moz-box-sizing is an unknown vendor extension**  
**Property -webkit-box-sizing is an unknown vendor extension**

... puhul jätsime koodi nii, nagu ta oli, kuna tegemist on alaga CSS'is, kus puudub ühtne „ilus“ brauseritevaheline standard.

## JavaScript

Viisime võrdlusoperaatorid `{!=, ==}` kujule `{===, !==}`.

Uue tühja listi defineerimise viisime kujult `var x = new Array()` kujule `var x = []`;

Lisaks hulganisti tühikute ja treppimise kasutuse viimist ühtsele kujule.

Vastavalt Jslint soovitudele muutsime `map.js` failis JavaScript objektidega tegelevas koodis kantsulu notatsiooni punkti-notatsioonile.

Mõnes `.js` failis leidus üleliigseid `$(document).ready()` plokke, need said eemaldatud.

Hoiatusi selle koodi osas, mis on kopeeritud face booki sisselogimise implementeerimiste juhistes, jätsime muutmata. Lisaks ignoreerisime ka „' \_\_\_\_ ' was used before it was defined“ hoiatusi, kuna see reaalselt probleeme ei tekita.

# JMeter jõudlustest.

## Setup

Kasutatakse jõudlustestide läbiviimiseks Apache JMeterit. Selleks loome uue JMeteri Test Plani, kuhu tekitasime Thread Groupi. Kasutades HTTP Proxy serverit, salvestasime kasutaja poolt tehtavad erinevad seisundeid mitte muutvad tegevused. Varieeruvuseks tegime Microsoft Exceliga kaks CSV Data Seti, mida kasutasime Kandidaadi otsingul erinevate argumenti andmiseks. Tulemuste jälgimiseks tekitasime Summary Reporti. Konsooli tulemuste kuvamiseks tekitasime ka Generate Summary Results-i.

## Konsooli tulemused

Konsoolis tulemuste saamiseks tuleb minna JMeter kausta (cd ROOT/e-valimised/JMeter), kus leidub evalimised.bat fail ja see konsoolis avada. Konsoolis mitte avades (double clickides evalimised.bat) tehakse test küll ära kuid jmeter sulgeb konsooli akna kohe testi lõppu ning pole võimalik tulemusi jälgida.

## Seadistus

Nagu oli ka tahetud, kasutasime 20 kasutajat, kes iga 10 sekundi tagant midagi muudavad. Tsükleid tegime 5, kuna sellest rohkem võtab ülearu kaua aega.

Number of Threads(users): 20

Ramp-Up Period(in seconds): 10

Loop Count: 5

## Tulemused



Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:

Remove

Log/Display Chk: ☐ Errors ☐ Successes

Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Aug. Bytes
Cache.manifest	133	154	141	318	27.01	0.00%	5.7/sec	8.51	1519
Avogadroite "kandidaadi" lehek	133	177	154	272	25.72	0.00%	5.3/sec	2.13	389
Kandidaadi strong 4ms parameetrite	133	173	157	281	25.49	0.00%	5.7/sec	3.85	651
Kandidaadi strong maastik	133	179	154	305	22.82	0.00%	5.7/sec	1.75	313
Kandidaadi strong erastik	133	177	158	429	38.94	0.00%	5.8/sec	2.02	384
Kandidaadi strong maastik ja erastik	133	167	154	388	28.92	0.00%	5.8/sec	1.95	242
Täpsemad info loomine	200	179	153	407	28.95	0.00%	11.4/sec	3.32	300
Google maps: paratide loomine	133	61	48	109	20.09	0.00%	5.8/sec	13.48	5959
Tulemusedpariti lehek	133	167	154	321	23.43	0.00%	5.7/sec	1.58	300
Koord loomine	133	168	153	280	20.77	0.00%	5.7/sec	2.30	410
TOTAL	1133	163	48	429	42.11	0.00%	57.3/sec	54.60	975

Kasutajad läbisid alguses esilehe, kust mingi kandidaadi otsingutesse. Otsiti erinevatest era kondadest ja/või piirkondadest kandidaate. Seejärel liiguti läbi tulemuste lehe, kus vaadati graafikult parteide lõikes tulemusi ning peale seda ka google mapsiga lahendatud piirkonna lõikes tulemusi. Siis lõpetati veebilehe surfimine.

Keskmiselt võtab laadimine aega 160 millisekundit ehk 0.16 sekundit, mis on üpriski mõistlik. Kõige rohkem aega võtsid kandidaadi otsingud 0.31 - 0.43 sekundit. Kõige vähem võttis aega google pakketide laadimine, mis võttis vaid 49 millisekundit. Vigade esinemise protsent päringutel on 0.00%. Seega ei esinenud ühtegi viga. Läbilase oli kokku 57.3 request/sec ehk päringut sekundis ja 54.5 KB/sec Keskmine näidise vastuse suurus baitides on 975,5 byte.

Seega on ka suurema koormuse all veebileht vägagi kiire ning ilma vigadeta.