

Aalto University
School of Science
!FIXME **Set degree program** FIXME!

Kimmo Puputti

!Fixme **Add English title** Fixme!

!Fixme **Add English subtitle** Fixme!

Master's Thesis
Espoo, !FIXME **Add English date** FIXME!

DRAFT! — Monday 9th January, 2012 — DRAFT!

Supervisor: Professor Petri Vuorimaa, Aalto University
Instructor: Risto Sarvas D.Sc.(Tech.)

Aalto University
School of Science

!FIXME Set degree program FIXME!

ABSTRACT OF
MASTER'S THESIS

Author:	Kimmo Puputti		
Title:	!FIXME Add English title FIXME! !FIXME Add English subtitle FIXME!		
Date:	!FIXME Add English date FIXME!	Pages:	18
Professorship:	Media Technology	Code:	T-110
Supervisor:	Professor Petri Vuorimaa		
Instructor:	Risto Sarvas D.Sc.(Tech.)		
!FIXME Add English abstract FIXME!			
Keywords:	!FIXME Add English keywords FIXME!		
Language:	English		

Aalto-yliopisto
 Perustieteiden korkeakoulu
 Tietotekniikan tutkinto-ohjelma

DIPLOMITYÖN
 TIIVISTELMÄ

Tekijä:	Kimmo Puputti		
Työn nimi:	!FIXME Add Finnish title FIXME! !FIXME Add Finnish subtitle FIXME!		
Päiväys:	!FIXME Add Finnish date FIXME!	Sivumäärä:	18
Professuuri:	Mediatekniikka	Koodi:	T-110
Valvoja:	Professori Petri Vuorimaa		
Ohjaaja:	Tohtori Risto Sarvas		
!FIXME Add Finnish abstract FIXME!			
Asiasanat:	!FIXME Add Finnish keywords FIXME!		
Kieli:	Englanti		

Acknowledgements

!FIXME Add acknowledgements FIXME!

Thank you.

!FIXME Decide city... FIXME!, !FIXME Add English date FIXME!

Kimmo Puputti

Contents

0.1	Thesis Git repository info	8
1	Introduction: Smartphone Market and the Need for Cross-Platform Support	9
1.1	Smartphone Landscape	10
1.2	HTML5	10
1.2.1	History	10
1.2.2	Markup	10
1.2.3	CSS3	10
1.2.4	JavaScript APIs	10
1.2.5	Related APIs	10
1.3	Modern Mobile Web Application Architecture	10
1.3.1	Single-Page applications	10
1.3.1.1	JavaScript MVC Libraries	10
1.3.2	Responsive Design	10
1.3.3	Progressive Enhancement	10
1.3.4	UI Libraries	10
1.3.4.1	jQuery Mobile	10
1.3.4.2	jQTouch	10
1.3.4.3	Sencha Touch	10
1.3.5	Hybrid Applications	10
1.3.6	Wrapping Web Applications Application Stores	10
1.4	Performance Guidelines	10
1.4.1	Make Fewer HTTP Requests	10
1.4.2	Use a Content Delivery Network	10
1.4.3	Add an Expires Header	10
1.4.4	Gzip Components	10
1.4.5	Put Stylesheets at the Top	10
1.4.6	Put Scripts at the Bottom	10
1.4.7	Avoid CSS Expressions	10
1.4.8	Make Javascript and CSS External	10

1.4.9	Reduce DNS Lookups	10
1.4.10	Minify JavaScript	10
1.4.11	Avoid Redirects	10
1.4.12	Remove Duplicate Scripts	10
1.4.13	Configure ETags	10
1.4.14	Make Ajax Cacheable	10
1.4.15	Splitting the Initial Payload	10
1.4.16	Loading Scripts Without Blocking	10
1.4.17	Coupling Asynchronous Scripts	10
1.4.18	Positioning Inline Scripts	10
1.4.19	Writing Efficient JavaScript	10
1.4.20	Scaling with Comet	10
1.4.21	Going Beyond Gzipping	10
1.4.22	Optimizing Images	10
1.4.23	Sharding Dominant Domains	10
1.4.24	Flushing the Document Early	10
1.4.25	Using Iframes Sparingly	10
1.4.26	Simplifying CSS Selectors	10
2	Research Question: HTML5 - Hype versus Realities?	11
3	Methods: Example Application and Library	12
3.1	Qt Developer Days 2011 Conference Schedule Application . . .	12
3.2	JSONCache JavaScript Library	12
4	Results: What Was Good and Where Were the Compromises	14
4.1	Targeting Different Platforms	15
4.1.1	Device Detection	15
4.1.2	Feature Detection	15
4.2	Targeting Different Screens	15
4.3	Handling Mobile Networks	15
4.3.1	Minimizing Data Transfer	15
4.3.2	Caching	15
4.3.3	Preloading	15
4.3.4	Offline Support	15
4.3.5	Handling Interruptions	15
4.4	Graphics and Animations	15
4.5	Performance Analysis	15
4.5.1	YSlow	15
4.5.2	PageSpeed	15

5	Discussion: Bright Future Ahead for HTML5	16
6	L^AT_EXtest	17
6.1	Citing	17

0.1 Thesis Git repository info

Build time: Monday 9th January, 2012 13:56

Git HEAD:

```
commit 1fe89c22f9128dcbe112d16395d7954879ae69b9
Author: Kimmo Puputti <kpuputti@gmail.com>
Date:   Mon Jan 9 13:24:38 2012 +0200
```

Add jsoncache screenshot and abbr command.

Repository status:

```
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
# modified:   methods.tex
# modified:   thesis.pdf
# modified:   thesis.tex
#
no changes added to commit (use "git add" and/or "git commit -a")
```


Chapter 1

Introduction: Smartphone Market and the Need for Cross-Platform Support

1.1 Smartphone Landscape

1.2 HTML5

1.2.1 History

1.2.2 Markup

1.2.3 CSS3

1.2.4 JavaScript APIs

1.2.5 Related APIs

1.3 Modern Mobile Web Application Architecture

1.3.1 Single-Page applications

1.3.1.1 JavaScript MVC Libraries

1.3.2 Responsive Design

1.3.3 Progressive Enhancement

1.3.4 UI Libraries

1.3.4.1 jQuery Mobile

1.3.4.2 jQTouch

1.3.4.3 Sencha Touch

1.3.5 Hybrid Applications

Chapter 2

Research Question: HTML5 - Hype versus Realities?

Chapter 3

Methods: Example Application and Library

3.1 Qt Developer Days 2011 Conference Schedule Application

3.2 JSONCache JavaScript Library

JSONCache is a lightweight JavaScript library for fetching JSON (!FIXME **abbreviation definition** FIXME!) data in flaky networks. The library was designed especially to handle flaky mobile networks with connection problems and short interruptions. The goal is to avoid networking as long as possible and failing gracefully if network connections are not stable.

JSONCache provides two main functionalities: data caching and attempting to fetch the data multiple times.

The caching layer uses the client side `localStorage` (!FIXME **citation needed** FIXME!) cache of HTML5 (!FIXME **abbreviation definition** FIXME!). Data requests can be done using the JSONCache API (!FIXME **abbreviation definition** FIXME!) which always checks the local cache first before opening any network connections. If the data is already in the cache, the cached data is checked for validity and if the data has not been expired, it is returned immediately. If the data is not in the cache or it has been expired, a new network request is made and the received data is cached and returned to the requestor. The expiration time of a data item can be configured in the library settings.

JSONCache also tries to fetch the data multiple times to handle small interruptions in network connection. !FIXME **add example and explain**

that it is very common `FIXME!`. If a data fetch fails, a new fetch is issued after a timeout (defined in the configuration). On subsequent attempts the timeout is increased, and after a defined number of attempts the fetch error is issued to the requestor.

Figure 3.1 shows an interactive demo of the JSONCache library. The demo¹ simulates the caching and fetching functionality of the library by simulating a flaky network according to the configuration.

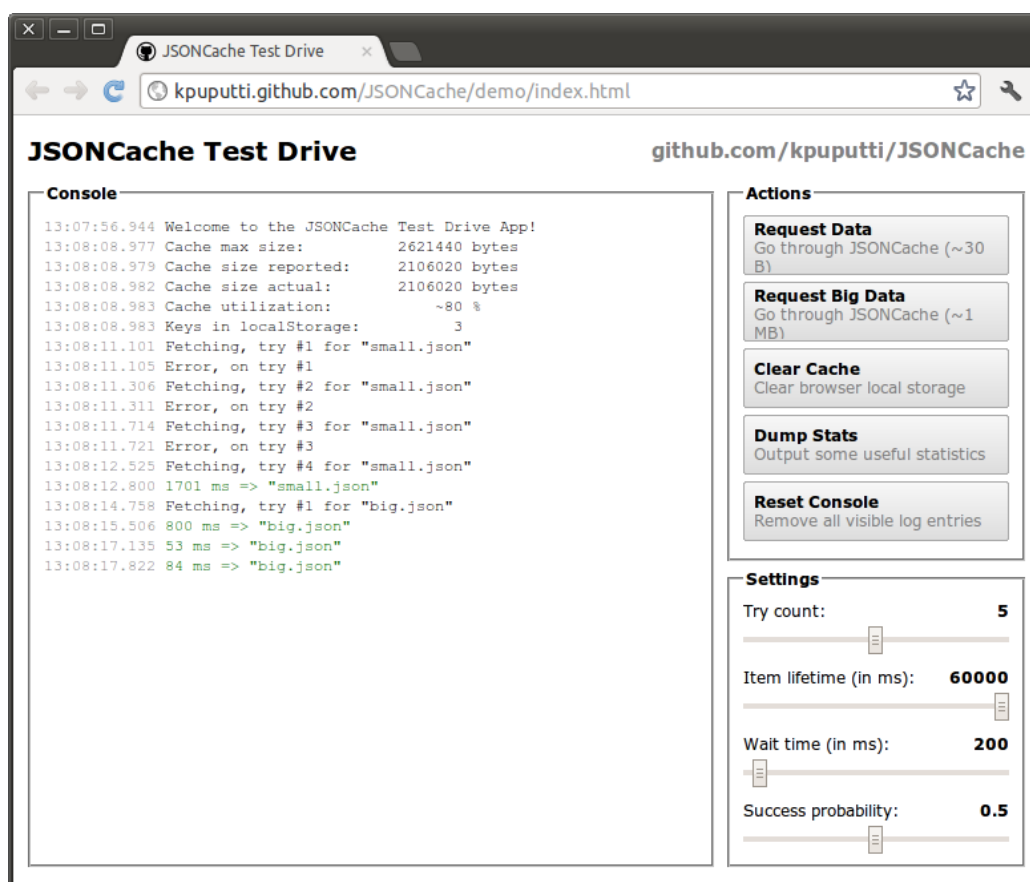


Figure 3.1: Interactive JSONCache demo.

¹<http://kpuputti.github.com/JSONCache/demo/index.html>

Chapter 4

Results: What Was Good and Where Were the Compromises

4.1 Targeting Different Platforms

4.1.1 Device Detection

4.1.2 Feature Detection

4.2 Targeting Different Screens

4.3 Handling Mobile Networks

4.3.1 Minimizing Data Transfer

4.3.2 Caching

4.3.3 Preloading

4.3.4 Offline Support

4.3.5 Handling Interruptions

4.4 Graphics and Animations

4.5 Performance Analysis

4.5.1 YSlow

4.5.2 PageSpeed

Chapter 5

Discussion: Bright Future Ahead for HTML5

Chapter 6

L^AT_EXtest

6.1 Citing

- Berners-Lee [1]
- Mikkonen & Taivalsaari [4]
- Taivalsaari & Mikkonen [7]
- Pilgrim [5]
- Crockford [2]
- Souders [6]
- Garrett [3]
- Zakas [8]

Bibliography

- [1] BERNERS-LEE, T. Long live the web. *Scientific American* 303, 6 (2010), 80–85.
- [2] CROCKFORD, D. *JavaScript: The Good Parts*. O'Reilly Media / Yahoo Press, 2008.
- [3] GARRETT, J. J. Ajax: A new approach to web applications. *Adaptive path* 18 (2005). Available at: <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>. Accessed 5-January-2012.
- [4] MIKKONEN, T., AND TAIVALSAARI, A. Apps vs. Open Web: The Battle of the Decade. In *2nd Annual Workshop on Software Engineering for Mobile Application Development* (2011).
- [5] PILGRIM, M. *HTML5: Up And Running*. O'Reilly Media, 2010.
- [6] SOUDERS, S. *High Performance Web Sites*. O'Reilly Media, 2007.
- [7] TAIVALSAARI, A., AND MIKKONEN, T. The Web as an Application Platform: The Saga Continues. In *Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on* (2011), IEEE, pp. 170–174.
- [8] ZAKAS, N. C. *High Performance JavaScript*. O'Reilly Media / Yahoo Press, 2010.