Seaside - An Object-Oriented Web Application Development Framework

Seminar
Weiterführende Themen zu Internet- und WWW-Technologien
Sommersemester 2016

Lennard Wolf

Betreuer:
Matthias Bauer, Haojin Yang
Prof. Dr. Christoph Meinel

July 20, 2016

Contents

1 Introduction

Seaside is an open source object-oriented web application development for the Smalltalk programming language. IT DOES THIS AND THAT.... This text aims at a basic understanding of its history and the context in which it was conceived, its concepts, how they differ from other frameworks of its kind, and what working with it looks like.

This work is structured as follows. Section 2 introduces ... and Section 6 concludes this work.

2 History of Seaside

To understand Seaside's origins, knowledge of the technologies currently popular at the beginnings of its development is required first. This Section will thus start out with such an overview, then present the original motivations behind Seaside, followed by a list of the people involved, a development timeline, and finally the current state of both the feature list and the development in general will be examined.

2.1 Historical Context

- Unknown: Server side or client side?
- Javascript was blocked by default Web Apps thus had to be server side
- User-readable URLs were not a must
- Ajax did not exist until 2005

2.2 Motivation

- introduce smalltalk
- Avi Bryant wanted to build interactive Squeak Wiki
- Inspired by Apple's WebObjects (Objective-C)

2.3 Development

- First Release 0.9 in February 2002 by Avi Bryant & Julian Fitzell, Open Source with MIT License
- Quick development, version 2.0 came October 2003
- datepicker was very modern

2.4 Current State

- list of current features:
- Live debugging and code changing while user interacts with web app
- debugger in browser
- no HTML, render page as Morph -> no browser needed
- CSS, Ajax, Comet, & Javascript support

3 Object-Oriented Web Application Development

- list of current features

4 Working with Seaside

- Session for each user

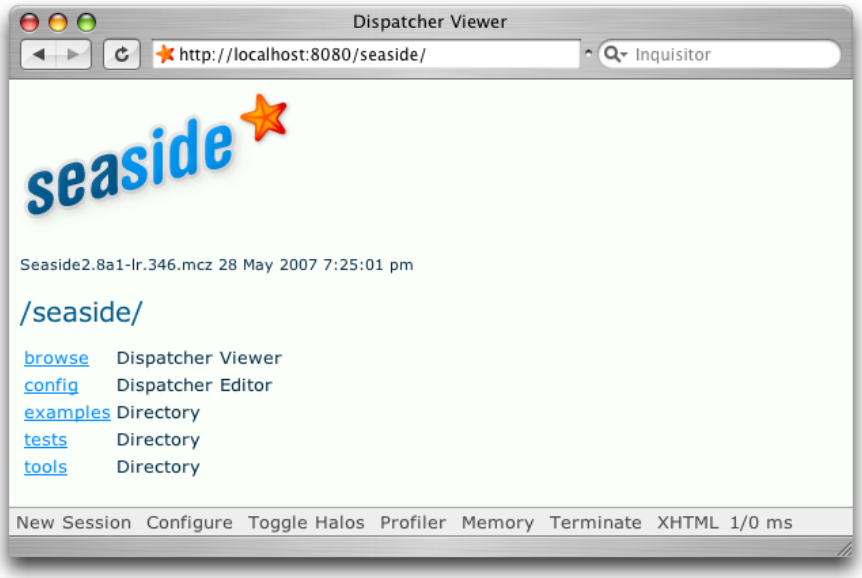

Figure 1: something something

5 Evaluation

- Each session object can be multiple megabytes big If server crashes, all sessions are lost
- Gemstone distributed but expensive
- Javascript not really debuggable
- Very hard to implement human readable URLs
- Not on par with today's demands of a fast dynamic webpage since its server side

6 Conclusion

References

[1] C. Willems and C. Meinel. "Tele-Lab IT-Security: an Architecture for an online virtual IT Security Lab", *International Journal of Online Engineering (iJOE)*, X, 2008.