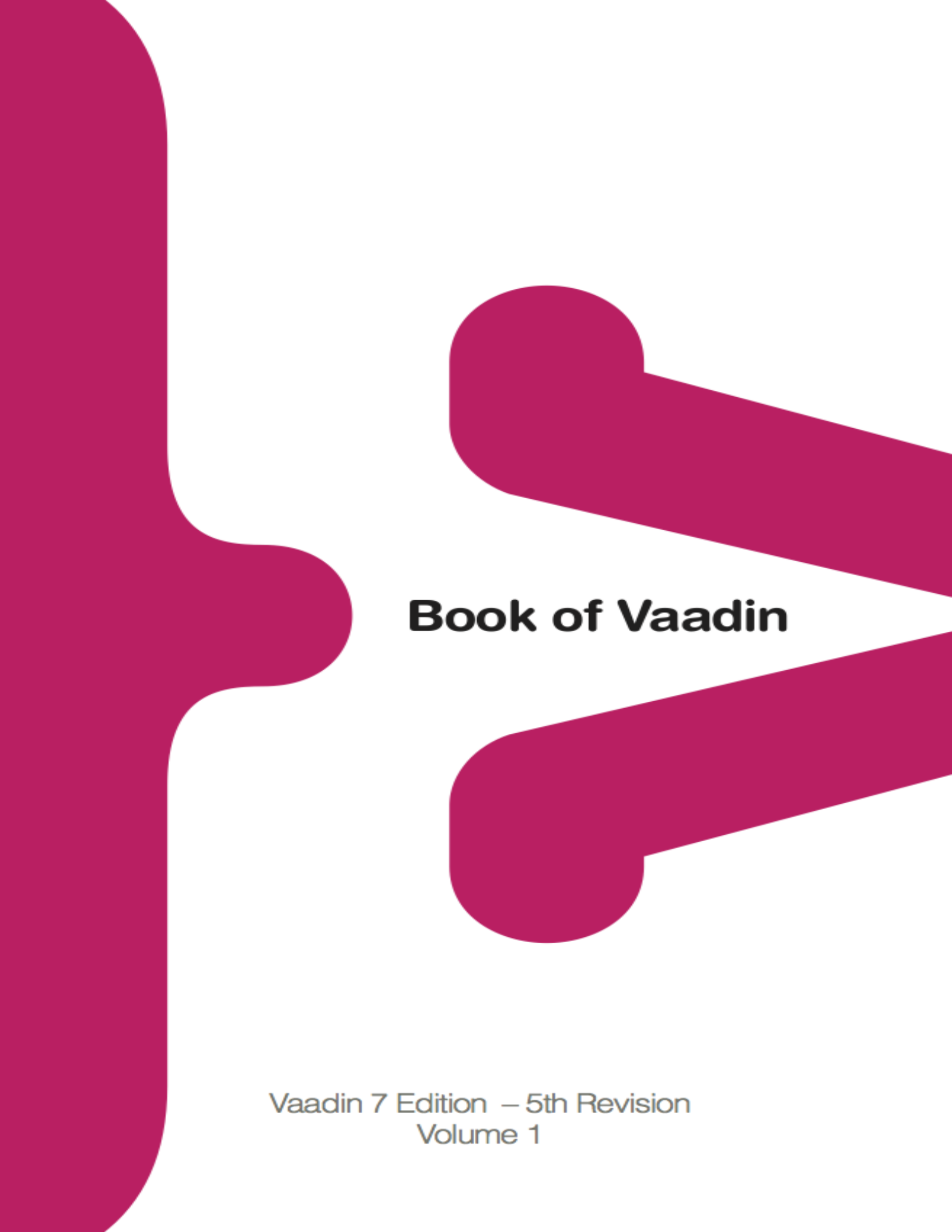


Vaadin 7 Edition – 5th Revision
Volume 1

Book of Vaadin

Marko Grönroos

Volume 1

Table of Contents

Book of Vaadin Marko Grönroos
Vaadin Ltd

Vaadin 7 Edition - 4th Revision
Published 2014-08-25
Vaadin Framewok 7.3

This book can be download for free at http://vaadin.com/book

Published by

Vaadin Ltd
Ruukinkatu 2-4
20540 Turku
Finland

Abstract

Vaadin is an AJAX web application development framework that enables developers to build high-quality user interfaces with Java, both on the server- and client-side. It provides a set of libraries of ready-to-use user interface components and a clean framework for creating your own components. The focus is on ease-of-use, re-usability, extensibility, and meeting the requirements of large enterprise applications.

Printed by Bookwell Oy in Juva, Finland, 2014.

ISBN 978-952-68006-4-6 (print)
ISBN 978-952-68006-5-3 (PDF)
ISBN 978-952-68006-2-2 (volume set)

Preface

This book provides an overview of the Vaadin Framework and covers the most important topics that you might encounter while developing applications with it. A more detailed documentation of the individual classes, interfaces, and methods is given in the Vaadin API Reference.

The book has grown greatly after its first edition, until it became too thick to fit any standard pocket. To accommodate all the content also in the printed edition, it has now been split into two volumes, the first one including the topics that you need to get started with Vaadin.

This edition mostly covers Vaadin 7.3 released in 2014. The most notable feature in the release is the new and highly customizable Valo theme.

In addition to the changes in the core framework, this edition features documentation for the TestBench 4 and TouchKit 4 add-ons, which are not yet released at the time of writing. The updated chapters are based on prerelease versions of the add-ons, so the final releases may include some changes.

Writing this manual is an ongoing work and it is rarely completely up-to-date with the quick-evolving product. Some features may not be included in this book yet. For the most current version, please see the on-line edition available at http://vaadin.com/book. You can also find PDF and EPUB versions of the book there. You may find the other versions more easily searchable than the printed book. The index in the book is incomplete and will be expanded later. The web edition also has some additional technical content, such as some example code and additional sections that you may need when actually doing development. The purpose of the slightly abridged print edition is more to be an introductionary textbook to Vaadin, and still fit in your pocket.

Also, many Vaadin 7 features are showcased as mini-tutorials, which are available in the Vaadin Wiki at https://vaadin.com/wiki/-/wiki/Main/Vaadin+7.

Who is This Book For?

This book is intended for software developers who use, or are considering to use, Vaadin to develop web applications.

The book assumes that you have some experience with programming in Java, but if not, it is at least as easy to begin learning Java with Vaadin as with any other UI framework. No knowledge of AJAX is needed as it is well hidden from the developer.

You may have used some desktop-oriented user interface frameworks for Java, such as AWT, Swing, or SWT, or a library such as Qt for C++. Such knowledge is useful for understanding the scope of Vaadin, the event-driven programming model, and other common concepts of UI frameworks, but not necessary.

If you do not have a web graphics designer at hand, knowing the basics of HTML and CSS can help so

that you can develop presentation themes for your application. A brief introduction to CSS is provided. Knowledge of Google Web Toolkit (GWT) may be useful if you develop or integrate new client-side components.

Organization of This Book

The Book of Vaadin gives an introduction to what Vaadin is and how you use it to develop web applications.

Volume 1

?

The chapter gives an introduction to the application architecture supported by Vaadin, the core design ideas behind the framework, and some historical background.

?

This chapter gives practical instructions for installing Vaadin and the reference toolchain, including the Vaadin Plugin for Eclipse, how to run and debug the demos, and how to create your own application project in the Eclipse IDE.

?

This chapter gives an introduction to the architecture of Vaadin and its major technologies, including AJAX, Google Web Toolkit, and event-driven programming.

?

This chapter gives all the practical knowledge required for creating applications with Vaadin, such as window management, application lifecycle, deployment in a servlet container, and handling events, errors, and resources.

?

This chapter gives the basic usage documentation for all the (non-layout) user interface components in Vaadin and their most significant features. The component sections include examples for using each component, as well as for styling with CSS/Sass.

?

This chapter describes the layout components, which are used for managing the layout of the user interface, just like in any desktop application frameworks.

Volume 2:

?

This chapter gives an introduction to Cascading Style Sheets (CSS) and Sass and explains how you can use them to build custom visual themes for your application.

?

This chapter gives an overview of the built-in data model of Vaadin, consisting of properties, items, and containers.

?

This chapter gives documentation for the SQLContainer, which allows binding Vaadin components to SQL queries.

?

This chapter gives instructions for using the visual editor for Eclipse, which is included in the Vaadin Plugin for the Eclipse IDE.

?

This chapter provides many special topics that are commonly needed in applications, such as opening new browser windows, embedding applications in regular web pages, low-level management of resources, shortcut keys, debugging, etc.

?

This chapter describes the development of Vaadin applications as portlets which you can deploy to any portal supporting Java Portlet API 2.0 (JSR-286). The chapter also describes the special support for Liferay and the Control Panel, IPC, and WSRP add-ons.

?

This chapter gives an introduction to creating and developing client-side applications and widgets, including installation, compilation, and debugging.

?

This chapter describes how to develop client-side applications and how to integrate them with a back-end service.

?

This chapter describes the built-in widgets (client-side components) available for client-side development. The built-in widgets include Google Web Toolkit widgets as well as Vaadin widgets.

?

This chapter describes how to integrate client-side widgets with their server-side counterparts for the purpose of creating new server-side components. The chapter also covers integrating JavaScript components.

?

This chapter gives instructions for downloading and installing add-on components from the Vaadin Directory.

?

This chapter documents the use of the Vaadin Charts add-on component for interactive charting with many diagram types. The add-on includes the Chart and Timeline components.

?

This chapter gives documentation of the JPAContainer add-on, which allows binding Vaadin components directly to relational and other databases using Java Persistence API (JPA).

?

This chapter gives examples and reference documentation for using the Vaadin TouchKit add-on for developing mobile applications.

?

This chapter gives the complete documentation of using the Vaadin TestBench tool for recording and executing user interface regression tests of Vaadin applications.

Supplementary Material

The Vaadin websites offer plenty of material that can help you understand what Vaadin is, what you can do with it, and how you can do it.

Demo Applications

The most important demo application for Vaadin is the Sampler, which demonstrates the use of all basic components and features. You can run it on-line at http://demo.vaadin.com/ or download it as a WAR from the Vaadin download page.

Most of the code examples in this book and many others can be found online at http://demo.vaadin.com/book-examples-vaadin7/book/.

Cheat Sheet

The two-page cheat sheet illustrates the basic relationship hierarchy of the user interface and data binding classes and interfaces. You can download it at http://vaadin.com/book.

Refcard

The six-page DZone Refcard gives an overview to application development with Vaadin. It includes a diagram of the user interface and data binding classes and interfaces. You can find more information about it at https://vaadin.com/refcard.

Address Book Tutorial

The Address Book is a sample application accompanied with a tutorial that gives detailed step-by-step instructions for creating a real-life web application with Vaadin. You can find the tutorial from the product website.

Developer's Website

Vaadin Developer's Site at http://dev.vaadin.com/ provides various online resources, such as the

ticket system, a development wiki, source repositories, activity timeline, development milestones, and so on.

The wiki provides instructions for developers, especially for those who wish to check-out and compile Vaadin itself from the source repository. The technical articles deal with integration of Vaadin applications with various systems, such as JSP, Maven, Spring, Hibernate, and portals. The wiki also provides answers to Frequently Asked Questions.

Online Documentation

You can read this book online at http://vaadin.com/book. Lots of additional material, including technical HOWTOs, answers to Frequently Asked Questions and other documentation is also available on Vaadin web-site.

Support

Stuck with a problem? No need to lose your hair over it, the Vaadin Framework developer community and the Vaadin company offer support to all of your needs.

Community Support Forum

You can find the user and developer community forum at http://vaadin.com/forum. Please use the forum to discuss any problems you might encounter, wishes for features, and so on. The answer to your problems may already lie in the forum archives, so searching the discussions is always the best way to begin.

Report Bugs

If you have found a possible bug in Vaadin, the demo applications, or the documentation, please report it by filing a ticket at the Vaadin developer's site at http://dev.vaadin.com/. You may want to check the existing tickets before filing a new one. You can make a ticket to make a request for a new feature as well, or to suggest modifications to an existing feature.

Commercial Support

Vaadin offers full commercial support and training services for the Vaadin Framework and related products. Read more about the commercial products at http://vaadin.com/pro for details.

About the Author

Marko Grönroos is a professional writer and software developer working at Vaadin Ltd in Turku, Finland. He has been involved in web application development since 1994 and has worked on several application development frameworks in C, C++, and Java. He has been active in many open source software projects and holds an M.Sc. degree in Computer Science from the University of Turku.

Acknowledgements

Much of the book is the result of close work within the development team at Vaadin Ltd. Joonas Lehtinen, CEO of Vaadin Ltd, wrote the first outline of the book, which became the basis for the first two chapters. Since then, Marko Grönroos has become the primary author and editor. The development team has contributed several passages, answered numerous technical questions, reviewed the manual, and made many corrections.

The contributors are (in rough chronological order):

Joonas Lehtinen
Jani Laakso
Marko Grönroos
Jouni Koivuviita
Matti Tahvonen
Artur Signell
Marc Englund
Henri Sara
Jonatan Kronqvist
Mikael Grankvist (TestBench)
Teppo Kurki (SQLContainer)
Tomi Virtanen (Calendar)
Risto Yrjänä (Calendar)
John Ahlroos (Timeline)
Petter Holmström (JPAContainer)
Leif Åstrand

About Vaadin Ltd

Vaadin Ltd is a Finnish software company specializing in the design and development of Rich Internet Applications. The company offers planning, implementation, and support services for the software projects of its customers, as well as sub-contract software development. Vaadin Framework, previously known as IT Mill Toolkit, is the flagship open source product of the company, for which it provides commercial development and support services.

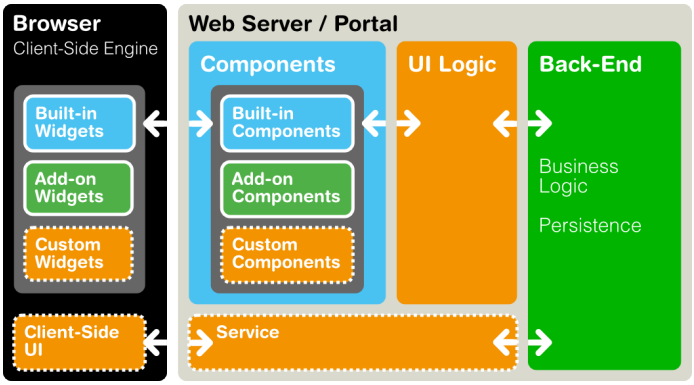
Chapter 1. Introduction

This chapter gives a brief introduction to software development with Vaadin. We also try to give some insight about the design philosophy behind Vaadin and its history.

1.1. Overview

Vaadin Framework is a Java web application development framework that is designed to make creation and maintenance of high quality web-based user interfaces easy. Vaadin supports two different programming models: server-side and client-side. The server-driven programming model is the more powerful one. It lets you forget the web and program user interfaces much like you would program a desktop application with conventional Java toolkits such as AWT, Swing, or SWT. But easier.

While traditional web programming is a fun way to spend your time learning new web technologies, you probably want to be productive and concentrate on the application logic. The server-side Vaadin framework takes care of managing the user interface in the browser and the AJAX communications between the browser and the server. With the Vaadin approach, you do not need to learn and deal directly with browser technologies, such as HTML or JavaScript.



? illustrates the basic architectures of web applications made with Vaadin. The server-side application architecture consists of the *server-side framework* and a *client-side engine*. The engine runs in the browser as JavaScript code, rendering the user interface, and delivering user interaction to the server. The UI logic of an application runs as a Java Servlet in a Java application server.

As the client-side engine is executed as JavaScript in the browser, no browser plugins are needed for using applications made with Vaadin. This gives it an edge over frameworks based on Flash, Java Applets, or other plugins. Vaadin relies on the support of Google Web Toolkit for a wide range of browsers, so that the developer does not need to worry about browser support.

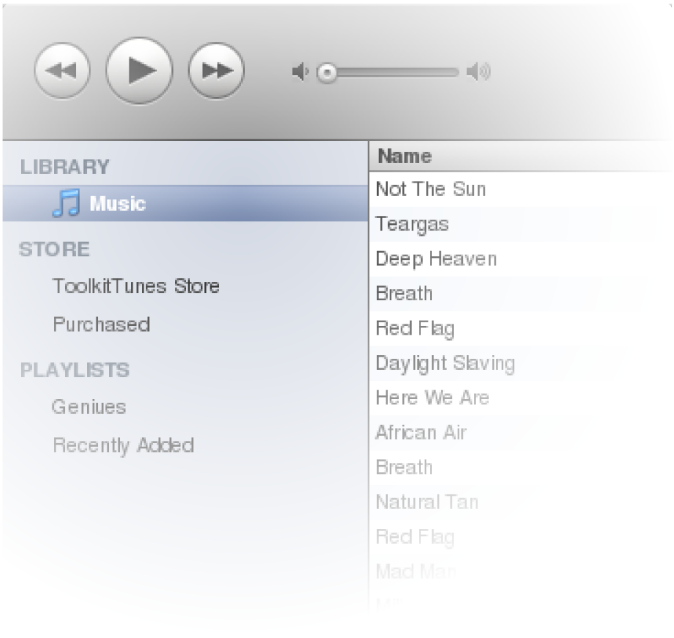
Because HTML, JavaScript, and other browser technologies are essentially invisible to the application logic, you can think of the web browser as only a thin client platform. A thin client displays the user interface and communicates user events to the server at a low level. The control logic of the user interface runs on a Java-based web server, together with your business logic. By contrast, a normal client-server architecture with a dedicated client application would include a lot of application specific communications between the client and the server. Essentially removing the user interface tier from

the application architecture makes our approach a very effective one.

Behind the server-driven development model, Vaadin makes the best use of AJAX (Asynchronous JavaScript and XML, see ? for a description) techniques that make it possible to create Rich Internet Applications (RIA) that are as responsive and interactive as desktop applications.

In addition to the server-side Java application development, you can develop on the client-side by making new widgets in Java, and even pure client-side applications that run solely in the browser. The Vaadin client-side framework includes Google Web Toolkit (GWT), which provides a compiler from Java to the JavaScript that runs in the browser, as well a full-featured user interface framework. With this approach, Vaadin is pure Java on both sides. Google Web Toolkit

Vaadin uses a client-side engine for rendering the user interface of a server-side application in the browser. All the client-server communications are hidden well under the hood. JavaScript Vaadin is designed to be extensible, and you can indeed use any 3rd-party widgets easily, in addition to the component repertoire offered in Vaadin. In fact, you can find hundreds of add-ons in the Vaadin Directory.



Vaadin allows flexible separation between the appearance, structure, and interaction logic of the user interface. You can design the layouts either programmatically or declaratively, at the level of your choosing. The final appearance is defined in themes in CSS or Sass, as described in ?.

We hope that this is enough about the basic architecture and features of Vaadin for now. You can read more about it later in ?, or jump straight to more practical things in ?.

1.2. Example Application Walkthrough

Let us follow the long tradition of first saying "Hello World!" when learning a new programming framework. First, using the primary server-side API.

HelloWorld.java

```
import com.vaadin.server.VaadinRequest;
import com.vaadin.ui.Label;
import com.vaadin.ui.UI;

@Title("My UI")
@Theme("valo")
public class HelloWorld extends UI {
    @Override
    protected void init(VaadinRequest request) {
        // Create the content root layout for the UI
        VerticalLayout content = new VerticalLayout();
        setContent(content);

        // Display the greeting
        content.addComponent(new Label("Hello World!"));

        // Have a clickable button
        content.addComponent(new Button("Push Me!",
            new ClickListener() {
                @Override
                public void buttonClick(ClickEvent e) {
                    Notification.show("Pushed!");
                }
            }));
    }
}
```

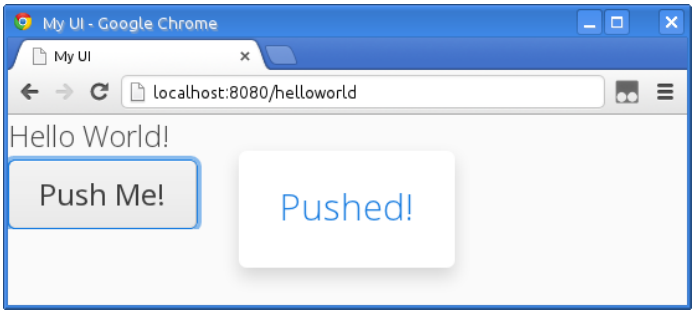
A Vaadin application has one or more *UI*s that extend the `com.vaadin.ui.UI` class. A UI is a part of the web page in which the Vaadin application runs. An application can have multiple UIs in the same page, especially in portals, or in different windows or tabs. A UI is associated with a user session, and a session is created for each user who uses the application. In the context of our Hello World UI, it is sufficient to know that the underlying session is created when the user first accesses the application by opening the page, and the init() method is invoked at that time.

The page title, which is shown in the caption of the browser window or tab, is defined with an annotation. The example uses a layout component as the root content of the UI, as that is the case with most Vaadin applications, which normally have more than one component. It then creates a new `Label` user interface component, which displays simple text, and sets the text to "Hello World!". The label is added to the layout.

The example also shows how to create a button and handle button click events. Event handling is described in ? and on the practical side in ?. In addition to listeners, in Java 8 you can handle events with lambda expressions, which simplifies the handler code significantly.

```
content.addComponent(new Button("Push Me!",
    event -> Notification.show("Pushed!")));
```

The result of the Hello World application, when opened in a browser, is shown in ?.



To run a program, you need to package it as a web application WAR package and deploy it to a server, as explained in ?. During development, you typically deploy to an application server integrated with the IDE.

Developing a pure client-side application, you could write a Hello World just as easily, and also in Java:

HelloWorld.java

```
public class HelloWorld implements EntryPoint {
    @Override
    public void onModuleLoad() {
        RootPanel.get().add(new Label("Hello, world!"));
    }
}
```

We do not set the title here, because it is usually defined in the HTML page in which the code is executed. The application would be compiled into JavaScript with the Vaadin Client Compiler (or GWT Compiler). It is more typical, however, to write client-side widgets, which you can then use from a server-side Vaadin application. For more information regarding client-side development, see ?.

1.3. Support for the Eclipse IDE

While Vaadin is not bound to any specific IDE, and you can in fact easily use it without any IDE altogether, we provide special support for the Eclipse IDE, which has become the most used environment for Java development. The support is provided in the Vaadin Plugin for Eclipse, which helps you in:

- Creating new Vaadin projects
- Creating custom themes
- Creating custom widgets

- Creating composite components with a visual editor
- Easily upgrading to a newer version of the Vaadin library

Using the Vaadin Plugin for Eclipse is the recommended way of installing Vaadin for development. Downloading the installation package that contains the JARs or defining Vaadin as a Maven dependency is also possible.

Installing and updating the Eclipse plugin is covered in ? and the creation of a new Vaadin project using the plugin in ?. See ?, ?, and ? for instructions on using the different features of the plugin.

1.4. Goals and Philosophy

Simply put, Vaadin's ambition is to be the best possible tool when it comes to creating web user interfaces for business applications. It is easy to adopt, as it is designed to support both entry-level and advanced programmers, as well as usability experts and graphic designers.

When designing Vaadin, we have followed the philosophy inscribed in the following rules.

Because our goals are high, the focus must be clear. Vaadin is designed for creating web applications. It is not designed for creating websites or advertisement demos. You may find, for example, JSP/JSF or Flash more suitable for such purposes.

We have chosen to emphasize robustness, simplicity, and maintainability. This involves following the well-established best practices in user interface frameworks and ensuring that our implementation represents an ideal solution for its purpose without clutter or bloat.

The Web is inherently document-centered and very much bound to the declarative presentation of user interfaces. While Vaadin allows for declarative designs of views, layouts, and even entire UIs, the programmatic approach by building the UIs from Java components frees the programmer from these limitations. To create highly dynamic views, it is more natural to create them by programming.

There should not be any limits on what you can do with the framework: if for some reason the user interface components do not support what you need to achieve, it must be easy to add new ones to your application. When you need to create new components, the role of the framework is critical: it makes it easy to create re-usable components that are easy to maintain.

1.5. Background

The Vaadin Framework was not written overnight. After working with web user interfaces since the beginning of the Web, a group of developers got together in 2000 to form IT Mill. The team had a desire to develop a new programming paradigm that would support the creation of real user interfaces for real applications using a real programming language.

The library was originally called Millstone Library. The first version was used in a large production application that IT Mill designed and implemented for an international pharmaceutical company. IT

Mill made the application already in the year 2001 and it is still in use. Since then, the company has produced dozens of large business applications with the library and it has proven its ability to solve hard problems easily.

The next generation of the library, IT Mill Toolkit Release 4, was released in 2006. It introduced an entirely new AJAX-based presentation engine. This allowed the development of AJAX applications without the need to worry about communications between the client and the server.

IT Mill Toolkit 5, released initially at the end of 2007, took a significant step further into AJAX. The client-side rendering of the user interface was completely rewritten using GWT, the Google Web Toolkit. Google Web Toolkit

IT Mill Toolkit 5 introduced many significant improvements both in the server-side API and in the functionality. Rewriting the Client-Side Engine with GWT allowed the use of Java both on the client and the server-side. The transition from JavaScript to GWT made the development and integration of custom components and customization of existing components much easier than before, and it also allows easy integration of existing GWT components. The adoption of GWT on the client-side did not, by itself, cause any changes in the server-side API, because GWT is a browser technology that is hidden well behind the API. Also theming was completely revised in IT Mill Toolkit 5.

The Release 5 was published under the Apache License 2, an unrestrictive open source license, to create faster expansion of the user base and to make the formation of a developer community possible.

IT Mill Toolkit was renamed as *Vaadin Framework*, or Vaadin in short, in spring 2009. Later IT Mill, the company, was also renamed as Vaadin Ltd. Vaadin means an adult female semi-domesticated mountain reindeer in Finnish.

With Vaadin 6, the number of developers using the framework exploded. Together with the release, the Vaadin Plugin for Eclipse was released, helping the creation of Vaadin projects. The introduction of Vaadin Directory in early 2010 gave it a further boost, as the number of available components multiplied almost overnight. Many of the originally experimental components have since then matured and are now used by thousands of developers. In 2013, we are seeing tremendous growth in the ecosystem around Vaadin. The size of the user community, at least if measured by forum activity, has already gone past the competing server-side frameworks and even GWT.

Vaadin 7 is a major revision that changes the Vaadin API much more than Vaadin 6 did. It is certainly more web-oriented than Vaadin 6 was. We are doing everything we can to help Vaadin rise high in the web universe. Some of this work is easy and almost routine - fixing bugs and implementing features. But going higher also requires standing firmer. That was one of the aims of Vaadin 7 - redesigning the product so that the new architecture enables Vaadin to reach over many long-standing challenges. Many of the changes required breaking API compatibility with Vaadin 6, especially in the client-side, but they are made with a strong desire to avoid carrying unnecessary legacy burden far into the future. Vaadin 7 includes a compatibility layer for making adoption of Vaadin 7 in existing applications easier.

Inclusion of the Google Web Toolkit in Vaadin 7 is a significant development, as it means that we now

provide support for GWT as well. When Google opened the GWT development in summer 2012, Vaadin (the company) joined the new GWT steering committee. As a member of the committee, Vaadin can work towards the success of GWT as a foundation of the Java web development community.

Chapter 2. Architecture

In ?, we gave a short introduction to the general architecture of Vaadin. This chapter looks deeper into the architecture at a more technical level.