

**ZK™**

# Ajax Without JavaScript™ Framework



**HENRI CHEN AND ROBBIE CHENG**



**ZK™: Ajax Without JavaScript™ Framework**

**Copyright © 2007 by Potix Corporation**

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-901-3

ISBN-10 (pbk): 1-59059-901-2

Printed and bound in the United States of America (POD)

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Java™ and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the US and other countries. Apress, Inc., is not affiliated with Sun Microsystems, Inc., and this book was written without endorsement from Sun Microsystems, Inc.

ZK™ and all ZK-based marks are trademarks or registered trademarks of Potix Corporation, in the US and other countries.

Lead Editor: Steve Anglin

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jonathan Gennick, Jason Gilmore, Jonathan Hassell, Chris Mills, Matthew Moodie, Jeffrey Pepper, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh, Tony Campbell

Project Manager: Richard Dal Porto

Copy Edit Manager: Nicole Flores

Copy Editor: Heather Lang

Assistant Production Director: Kari Brooks-Copony

Compositor: Richard Ables

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail [info@apress.com](mailto:info@apress.com), or visit <http://www.apress.com>.

The information in this book is distributed on an "as is" basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com> in the Source Code/Download section.

# Contents

About the Authors .....	ix
About the Technical Reviewer .....	xi
Introduction .....	xiii

## Part 1 ■■■ Getting to Know the ZK Ajax Framework

<b>CHAPTER 1</b>	<b>What Is the ZK Ajax Framework? .....</b>	<b>3</b>
	ZK Ajax Framework .....	4
	Ajax: Behind-the-Scenes Technology .....	5
	A Presentation Layer .....	6
	A Server-centric Event-Driven Framework .....	7
	A Component-Based GUI Toolkit .....	7
	Summary .....	7
<b>CHAPTER 2</b>	<b>Getting Started with ZK .....</b>	<b>9</b>
	Installing the Java Runtime Environment .....	9
	Installing the Java Servlet Container .....	10
	Using the Installer .....	10
	Using the ZIP File .....	10
	Deploying and Testing zkdemo-all.war .....	11
	Relative Libraries .....	11
	Configuring web.xml .....	12
	Configuring zk.xml .....	14
	Summary .....	15
<b>CHAPTER 3</b>	<b>Building Your First ZK Application .....</b>	<b>17</b>
	Your First ZK Application: Hello World! .....	17
	The First ZK Component: Window .....	19
	Injecting Ajax to Your ZK Application .....	22
	Summary .....	27
<b>CHAPTER 4</b>	<b>Introducing the Versatile ZK Components .....</b>	<b>29</b>
	Components, Pages, and Desktops .....	29
	The Component Life Cycle .....	30

The Life Cycle of a ZUML Page . . . . .	30
The Page Initialization Phase . . . . .	30
The Component Creation Phase . . . . .	31
The Event Processing Phase . . . . .	31
The Rendering Phase . . . . .	32
Updating Pages . . . . .	32
The Request Processing Phase . . . . .	32
The Event Processing Phase . . . . .	32
The Rendering Phase . . . . .	33
Component Garbage Collection . . . . .	33
Using Component Attributes . . . . .	33
The id Attribute . . . . .	34
The if and unless Attributes . . . . .	36
The forEach Attribute . . . . .	38
The use Attribute . . . . .	38
ID Spaces . . . . .	40
Getting a Fellow that's Inside the ID Space . . . . .	40
Getting a Component that's Outside the ID Space . . . . .	41
Getting a Component from Another Page . . . . .	43
Using zscript and EL expressions . . . . .	43
zscript . . . . .	43
EL Expressions . . . . .	45
Events . . . . .	46
ZUML and XML Namespaces . . . . .	47
Creating a Real Modal Dialog . . . . .	48
Messagebox . . . . .	48
Fileupload . . . . .	49
Implementing Macro Components . . . . .	49
Integrating Mega Components . . . . .	50
Google Maps . . . . .	50
FCKeditor . . . . .	52
Dojo . . . . .	52
Summary . . . . .	54

## Part 2 ■■■ Applying Your ZK Knowledge

■ CHAPTER 5	Setting Up the Development Environment . . . . .	57
	Installing the Java SE (JDK) . . . . .	57
	Installing Tomcat . . . . .	58
	Installing the Eclipse IDE SDK . . . . .	58

Downloading Eclipse with WTP . . . . .	58
Specifying the Workspace for Eclipse . . . . .	59
Setting Up Tomcat to Work with Eclipse . . . . .	59
Defining a Server Instance . . . . .	62
Setting Up the .zul File Extension in the XML Editor . . . . .	64
Configuring ZK to Work with Eclipse . . . . .	66
Creating a Dynamic Web Project . . . . .	66
Importing ZK JAR Libraries and Setting Up Web Configurations . . . . .	68
Creating a New index.zul File . . . . .	70
Running Your First Application . . . . .	71
More Convenient Tools . . . . .	73
Importing ZK Source Code into Eclipse . . . . .	73
Integrating the zul.xsd XML Schema . . . . .	75
Creating a Page Using the zul.xsd XML Schema . . . . .	77
Patching a Lost xsd URL in Eclipse . . . . .	83
Summary . . . . .	83
<b>CHAPTER 6</b> <b>Creating a Real WebApplication . . . . .</b>	<b>85</b>
Introducing Java Pet Store 2.0 . . . . .	85
Creating the ZK Pet Shop Application Installing ZK Pet Shop . . . . .	86
Creating the Home Page of ZK Pet Shop . . . . .	87
Building a Fish-Eye List with Dojo . . . . .	87
Listening for an onClick Event . . . . .	89
Listening for an onClick Event . . . . .	89
Including banner.zul in index.zul . . . . .	90
Adding the Parrot Icon . . . . .	91
Adding the Title . . . . .	92
Adding the Menu Bar . . . . .	92
Requirements for the ZK Pet Shop Application . . . . .	93
Interaction Between Viewer and Controller . . . . .	93
Publishing Information About Pets . . . . .	94
Creating seller.zul . . . . .	94
Declaring the Controller for seller.zul . . . . .	95
Creating a Form for Submitting Pet Information . . . . .	95
Adding the Rich Text Editor . . . . .	97
Providing the File Upload Functionality . . . . .	97
Creating a Form for Submitting a Seller's Contact Information . . . . .	99
Adding a CAPTCHA . . . . .	100

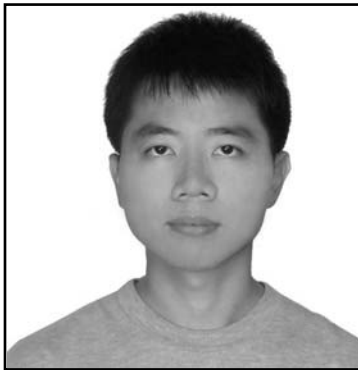
Adding Client-Side Validation Constraints . . . . .	101
Saving the User's Submission in a Database . . . . .	103
Browsing Pet Information and Photos . . . . .	105
Updating catalog.zul . . . . .	105
Creating the Menu Bar . . . . .	106
Creating the Large Photo . . . . .	109
Creating the Gallery of Pet Thumbnail Images . . . . .	109
Creating the Description Pane . . . . .	111
Searching for a Pet's Location with Google Maps . . . . .	111
Creating mapall.zul . . . . .	111
Creating a Form for Submitting an Address as a Query . . . . .	112
Using Google Maps to Show Pet Locations . . . . .	114
Showing the Result with Google Maps . . . . .	115
Showing Pets' Locations on the gmap Component . . . . .	116
Showing Pets' Information in the gr id . . . . .	116
Redirecting this Page to catalog.zul . . . . .	117
Indicating the Pet's Location on the Map . . . . .	117
Customizing the Event Listener . . . . .	118
Showing the Pet's Details and Thumbnail Photo in a Pop-Up Window . . . . .	118
Creating popup.zul . . . . .	120
Summary . . . . .	121
<b>CHAPTER 7</b> Linking the GUI to a Database . . . . .	123
Preparing the Database Environment . . . . .	123
Downloading HSQL Database . . . . .	123
Creating the ZK Pet Shop Tables . . . . .	124
JDBC and Connection Pool . . . . .	128
Making ZK Work with Hibernate . . . . .	130
Prerequisites of Using Hibernate . . . . .	131
Creating the Java Class . . . . .	131
Editing the Hibernate Configuration File . . . . .	132
Mapping Object Resources . . . . .	133
Using aMapping File . . . . .	134
Using JavaAnnotations . . . . .	135
Advanced Features of Hibernate . . . . .	135
Units of Work . . . . .	135
Keeping a Session Open . . . . .	136
Configuring ZK for Use with Sessions . . . . .	137
The ZK Pet Shop Data Model with Annotations . . . . .	137
Table Annotations . . . . .	139

Property Annotations . . . . .	139
Primary Key Annotations . . . . .	140
Implementing the Façade Class of the ZK Pet Shop . . . . .	142
The CatalogFacade Class . . . . .	142
Implementing the CatalogFacade Class with Hibernate . . . . .	143
SELECT Statement . . . . .	143
INSERT Statement . . . . .	144
User's Activities and the Behind-the-Scene Mechanism . . . . .	145
The User Submits a Pet for Sale . . . . .	145
The User Visits catalog.zul for the First Time . . . . .	146
Browse Pets of a Certain Product Type . . . . .	148
Searching for a Pet's Location with Google Maps . . . . .	150
Summary . . . . .	152
<b>CHAPTER 8 Binding Data Automatically . . . . .</b>	<b>153</b>
Getting Rid of The Plumbing Code . . . . .	153
Using the ZUML Annotations Class . . . . .	154
Formatting Data Binding Annotations . . . . .	154
Using Data Binding with the ZK Annotations Class . . . . .	156
The load-when Tag . . . . .	157
The save-when Tag . . . . .	157
Adding Data Binding to the ZK Pet Shop GUI . . . . .	158
Using Data Binding in the ZK Pet Shop . . . . .	159
Declaring Getter and Setter Methods of the Data Model . . . . .	159
Declaring the ZUML Annotations Namespace . . . . .	161
Defining the Data Binding Annotations . . . . .	162
Specifying Events that Invoke the Data-Binding Manager . . . . .	163
Using Data Binding with Data Collections . . . . .	164
Declaring Getter and Setter Methods of the Data Model . . . . .	165
Defining the Data Binding Annotations . . . . .	166
Specifying the Data Collection in the model Attribute of listBox . . . . .	166
Defining a Template of ZK Components Using the _var Tag . . . . .	166
Summary . . . . .	167

# About the Authors



■ **HENRI CHEN** is the cofounder of the ZK Ajax Framework. He has more than 20 years' programming experience in various areas and programming languages. He has been writing Java programs since the initial release of the Java language in 1995 and ported JavaOS and HotJava to the world's first StrongARM-based network computer, which won the "Best of System" award in Comdex 1996. He is also an expert in thin-client programming, embedded systems, and Ajax web programming.



■ **ROBBIE CHENG** is an engineer on the ZK team. Though his major was English literature in college, he got his MBA degree in management of information systems. He has been an instructor in the use of the ZK Framework and the developer of zk registry, and has published many articles on the ZK web site. In addition to programming, his hobby is watching movies, and his favorite movie is *The Bridges of Madison County*.



# About the Technical Reviewer

■ **FRANK W. ZAMMETTI** is a web architect specialist for a leading worldwide financial company by day and an active open source contributor by night. He is the author of two books—*Practical Ajax Projects with Java Technology* (Apress, 2006) and *Practical JavaScript, DOM Scripting, and Ajax Projects* (Apress, 2007)—and has another on the way. Frank has more than 13 years' of professional development experience under his belt (not to mention a few extra inches) and over 12 more (years, not inches!) of amateur experience before that. Frank lives in the United States with his wife, Traci, and his two kids, Andrew and Ashley. He is also visited nightly by beings from another world, which is fine, except that they won't stay off his grass no matter how nicely he asks, and you wouldn't believe how dry dark-matter-based impulse engines can make a lawn!

# Introduction

In 1994, I developed an infrastructure, inspired by zApp and the Object Window Library (OWL), for developing an accounting system for Windows. In 2000, I developed another infrastructure, inspired by Struts and WebWorks, for developing another accounting system for the J2EE platform. After coaching and watching the development of both systems, I found that not only did the web edition require more advanced programming skills and prerequisites but its total cost was four times more than the client/server's. Worst of all, the user experience harkened back to the age of green terminals, though the look, after decorating with proper images and cascading style sheets, was modern and fresh.

I started wondering whether these problems were intrinsic to the web or if the programming model was simply inadequate. Looking back at the success of desktop applications in the 1990s, the event-driven, component-based programming model played a big role. Blessed by being easy to learn and develop, this model is the standard and best way to handle interactive and responsive user interfaces. Could this model be applied to web applications? After using ZK to develop several commercial projects, I believe I've got the answer—yes!

The reasoning behind that answer is what I want to share with you in this book. How can the ZK Framework make your life easy and your customers happy at the same time? How does ZK help you painlessly write a rich web application? How can you write a rich Ajax web application without learning JavaScript? How you can concentrate on improving your application itself rather than focusing on the plumbing required ensure browser compatibility? You will find your answers in this book.

This book is about how to make Ajax programming simple and easy—the core values of the ZK Framework—as simple as programming desktop applications and as easy as authoring HTML pages. Writing rich Ajax web applications can be very elaborate. On the browser side, you can program user interfaces with HTML, DOM, CSS, and JavaScript. On the server side, you can write business logic and data-access code with another language, such as Java. Then, you have to handle the browser-to-server messages with asynchronous HTTP. Finally, you still have to fight the incompatibility issues and JavaScript bugs across browsers.

This book will introduce you to painlessly programming Ajax applications with the ZK Framework. You are not required to write user interfaces on the browser side. Rather, you construct your applications on the server side with ZK's more than 160 Java components. The complex heterogeneous technologies involved in Ajax programming are automatically handled by ZK behind the scenes.

In this book, I tell you how to install and run ZK programs and how ZK completes its behind-the-scenes jobs, as well as explaining the important ZK components. I then walk you through creating a real web application, where you learn how to design the application screens, access the database, and write control code to coordinate the ZK presentation layer and the data accessing layer.

I sincerely hope this book helps you out of the old, painstaking, and time-consuming way of developing Ajax web applications. So read on to see how ZK makes your life easy and your customers happy at the same time. Enjoy your Ajax web programming experience.

## Who This Book Is For

This book is especially for those who are interested in Ajax but don't want to learn to use JavaScript, CSS, and DOM, and who prefer not to deal with the incapability among browsers. To read this book, you should have basic knowledge of Java and HTML for building a web application with ZK.

## How This Book Is Structured

This book is divided into two parts: the first part, “Getting to Know the ZK Ajax Framework” contains Chapters 1 through 4 and introduces you to the framework and the development environment setup:

- Chapter 1, “What Is the ZK Ajax Framework?” includes a basic introduction to ZK, which is a server-centric framework requiring little programming to use. Also, the architecture of ZK and how ZK realizes the idea of Ajax are explained.
- Chapter 2, “Getting Started with ZK,” tells you how to set up the environment and to deploy the web application for running the ZK Framework.
- Chapter 3, “Building Your First ZK Application,” demonstrates how to build the famous Hello World web application with ZK, along with providing some experience about how ZK components interact with each other.
- Chapter 4, “Introducing the Versatile ZK Components,” introduces more concepts about using components and facilities provided by ZK to build a web application.

This book's second part, “Applying Your ZK Knowledge,” explains how to build a real application with ZK:

- Chapter 5, “Setting Up the Development Environment,” shows you how to set up Eclipse as the development environment.
- Chapter 6, “Creating a Real Web Application,” provides a step-by-step explanation for using the ZK framework to implement the GUI of a ZK Pet Shop application, based on the famous Java Pet Store reference application.
- Chapter 7, “Linking the GUI to a Database,” explores how to build the persistent layer with Hibernate and how the behind-the-scenes technology of ZK is implemented.
- Chapter 8, “Binding Data Automatically,” introduces how to automatically move data between the GUI and controllers with ZK's data binding mechanism.

## Downloading the Code

The source code for this book is available to readers at [http://sourceforge.net/project/showfiles.php?group\\_id=156140](http://sourceforge.net/project/showfiles.php?group_id=156140) and at <http://www.apress.com> in the Source Code/Download section.

## Contacting the Authors

You can contact Henri Chen at [henrichen@zkoss.org](mailto:henrichen@zkoss.org) and Robbie Cheng at [robbiecheng@zkoss.org](mailto:robbiecheng@zkoss.org). Also, check out the ZK web site at <http://www.zkoss.org>.

PART 1



# Getting to Know the ZK Ajax Framework

*ZK is an open source Ajax web framework that enables a rich user interface for web applications with no JavaScript and little programming.*

—ZK web site

**In this part, we will discuss the various aspects of the ZK Ajax Framework and how Ajax is implemented without programming JavaScript.**

## CHAPTER 1

# What Is the ZK Ajax Framework?

Over a decade, web applications have evolved from static HTML pages, to dynamic HTML (DHTML) pages, to pages using applets and Flash, and finally, to those incorporating Ajax (Asynchronous JavaScript and XML) technologies. Two great examples of Ajax are Google Maps and Google Suggest. Ajax breathes new life into web applications by delivering the same level of interactivity and responsiveness as desktop applications. However, unlike applets or Flash, Ajax is based on the standard browser and JavaScript, and no proprietary plug-in is required.

Ajax is a kind of next-generation DHTML; hence, it relies heavily on JavaScript to listen to events triggered by user activity and manipulates the visual representation of a page (that is, the document object model, or DOM) in the browser dynamically.

So, how can you easily incorporate Ajax into your web pages? Use the ZK Framework. Unlike most other Ajax frameworks, ZK does not require you to have any knowledge of JavaScript to develop Ajax-based web applications, since the ZK engine auto-generates the JavaScript code. To develop a web application with ZK, you need to know only a little about HTML. To simplify development of web application, the ZK team has also defined the ZK User Interface Markup Language (ZUML) to provide an intuitive way to create ZK components by simply declaring an enclosing tag, which is similar in format to an HTML tag.

Let's look at ZK's Live Demo web page to experience the magic of ZK (<http://www.zkoss.org/zkdemo/userguide>). This live demonstration explores various examples of ZK components that are built with Ajax and provides an online programming experience. You can modify the source code of each example online and see the results immediately.

Figure 1-1 shows demonstrations of two ZK components: Chart and Drag and Drop. Look at the Chart example shown on the left in Figure 1-1; the distribution of the pie chart will be updated automatically when you change the values of the programming categories. In the Drag and Drop component example, shown on the right in Figure 1-1, the sequence of each row will be reordered after dropping the dragged row onto the other one.

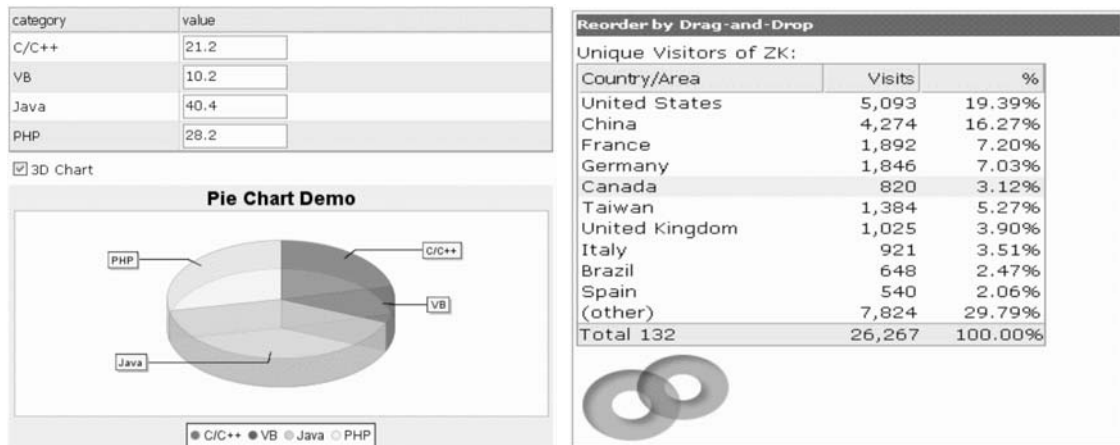


Figure 1-1. Live Demo shows how easily you can manipulate ZK components like these two.

## ZK Ajax Framework

ZK is an event-driven, component-based framework to enable rich user interfaces for web applications. ZK includes an Ajax-based event-driven engine, a rich set of XML User Interface Language (XUL) and XHTML components, and a markup language called ZK User Interface Markup Language (ZUML).

With ZK, you represent your application in feature-rich XUL and XHTML components, and you manipulate them based on events triggered by user activity, as you have probably done for years in desktop applications. Unlike in most other frameworks, Ajax is a behind-the-scenes technology in ZK. ZK simplifies the development of rich Ajax web applications in the following ways:

- The event-driven engine brings the intuitive desktop programming model to web developers.
- The XUL and XHTML components enrich web applications by using off-the-shelf building blocks.
- The ZUML markup language makes the design of rich user interfaces as simple as authoring HTML pages.

In this chapter, I will explain how Ajax is accomplished by behind-the-scene mechanism of ZK. And, of course, the three most important characteristics of the ZK Framework will be introduced:

- It's a presentation layer tool.
- It's a server-centric framework.
- It's has a component-based GUI.

## Ajax: Behind-the-Scenes Technology

The Ajax-based mechanism of ZK is realized by three important parts, as depicted in Figure 1-2: the ZK loader, ZK AU (asynchronous update) engine, and ZK client engine. The ZK loader and ZK AU engine are each composed of a set of Java servlets, and the ZK Client Engine is composed of JavaScript codes. Figure 1-2 illustrates the mechanism when the ZK loader receives a URL request at the first time.

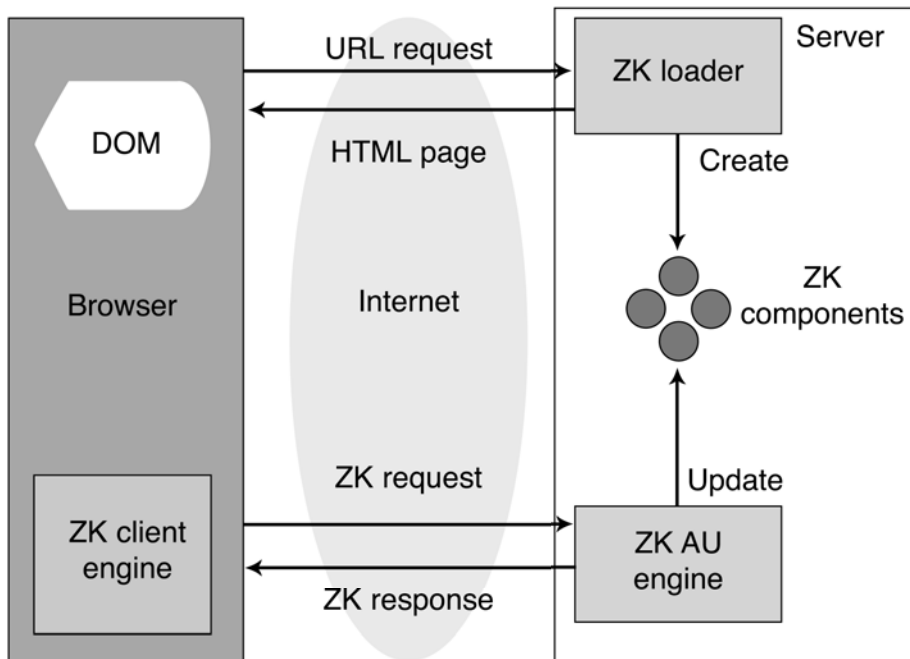


Figure 1-2. The ZK loader, the ZK AU engine, and the ZK client engine at work



The mechanism works like this:

1. The ZK Loader interprets an incoming URL request and generates a corresponding HTML page, including standard HTML, CSS, and JavaScript code, and ZK components at the server side.
2. The ZK loader sends the HTML page to the client and the ZK client engine. The ZK client engine resides on the client side for monitoring JavaScript events queued in the browser.

---

Note ➔ The ZK client engine is composed of a lot of JavaScript that is responsible for receiving events and updating the content of web pages.

---

3. If any JavaScript events are triggered, the ZK client engine will send those events (that is, those Ajax requests) back to ZK AU engine on the server side.
4. The ZK AU engine receives the Ajax requests, updates the properties of ZK components, and sends an Ajax response back to the client side.
5. The ZK client engine receives the response and updates the corresponding content in the browser's Document Object Model (DOM) tree.

This process is constantly repeated until the URL is no longer referenced by the user.

## **A Presentation Layer**

ZK is designed to be as thin as possible, so it focuses only on the presentation tier. It does not require any other back-end technologies, and all of your favorite middleware, such as Java Database Connectivity (JDBC), Hibernate, Java Mail, Enterprise Java Beans (EJBs), and Java Message Service (JMS), works appropriately with ZK. Thus, you can build your web application with familiar technologies without learning new ones.

## **A Server-centric Event-Driven Framework**

With most Ajax frameworks, the role of server is passive, since it is responsible only for providing and accepting data after receiving requests from the client side. The communication between components is quite complex and requires a lot of JavaScript programming, not to mention the problem of incompatibility among JavaScript and browsers.

By contrast, in ZK's solution, all the components are created on the server side, which makes communication between components easier since you can access these components directly on the server side. Moreover, the way components communicate with each other is event driven, which means interaction can be triggered by a user's activities on the client side or events sent from other components. In short, ZK mirrors the simplicity of developing desktop applications in the development of web applications and gives users more interactivity and more responsiveness.

## **A Component-Based GUI Toolkit**

ZK is a component-based graphical user interface (GUI) toolkit. ZK provides more than 70 XUL-based and 80 XHTML-based components, and it provides the ZUML markup language for designing user interfaces. Programmers design their application pages in feature-rich XUL/XHTML components, which manipulate applications upon events triggered by the end user's activity. It is similar to the programming model found in desktop GUI-based applications.

## **Summary**

Unlike other Ajax frameworks, which require a lot of knowledge about CSS, JavaScript, and DOM, ZK provides a shortcut for you to create Ajax-based web applications without learning other technologies, and developing ZK applications with ZUML requires you to know only a small amount about HTML. The ZK engine will handle the rest of the plumbing—generating HTML, CSS, and JavaScript code.

In addition, ZK makes it possible to develop web applications in an environment similar to desktop applications', since all components are also created on the server side. In other words, the relationship among components on the client side and components on server side