Delivering Rich Internet Applications with Ajax4jsf

Modern Web 2.0 applications set a new level of expectations for enterprises on the Web. Developers face heightened requirements for "richer" user interfaces at the same time enterprise Web applications increase in size and complexity. In this paper, we show how the open source Ajax4jsf framework, as part of a component-based approach using JavaServer Faces, solves this problem.

A White Paper from Exadel, Inc.

Exadel, Inc.
1850 Gateway Blvd.
Suite 1080
Concord, CA 94520
925-602-5561
www.exadel.com
info@exadel.com

Table of Contents

The Challenges of the Modern Web

Everybody is now talking about Web 2.0, a Web that is more dynamic, richer, more interactive, and, ultimately – much more exciting than anything we know now. It's just human nature to look for unusual and new stuff. But, what does that mean for business applications?

Let's look at Google Maps. Yes, we are all excited by Google Maps. It looks great. It is very interactive. And, most importantly, it behaves completely differently from what we expect to see in a "normal" browser. But, compared to any realistic business application – say something like trivial Internet banking – Google Maps is nothing. It supports just a few use cases compared to the hundreds or thousands of use cases for a typical business application. How many input fields does it have? How many validation rules? How complex is the business process and page flow? Anybody could build Google Maps using any kind of technology, but business application developers face completely different challenges, challenges that are much more demanding!

What Web 2.0 Mean for Enterprises

Applications have become more and more complex. Any project that is targeting the nominal goal of "retiring" an old outdated system often ends up, not only replacing the old functionality, but actually adding many new functions to the system. Also, businesses like to thrill customers. An Internet bank should be, more dynamic not less dynamic than Google Mail! So, when we start thinking about the challenges facing enterprise developers, we see

that they must deliver cool, rich, dynamic, exciting new functionality in *bulk packaging*. And, they must deliver it yesterday! Did I also mention that the application must be flexible, scalable, secure, and cheap?

Today, enterprise developers must serve two demanding groups of masters at the same time: consumers desiring all of the user interface aspects of Web 2.0 and businesses wanting more and more functionality. Fortunately, there is a solution for this problem.

The Promise of Components

Let's start with components. Developers have been talking about components for ages. It would seem absolutely obvious: you can construct a building from big blocks much faster than from small bricks – and a lot more cheaply. Moving from civil engineering to the IT industry, the same idea is behind the movement from Assembler language (or even pre-Assembler machine code back in prehistoric times, if anybody still remembers) towards high-level and then object-oriented languages.

If we look at the course of Web development from that point of view, the same movement is evident, from plain HTML (Assembler) to JSP (Fortran) to JavaServer Faces (Java for the Web). I put JavaServer Faces at the end of this evolutionary chain, because it is the latest and most advanced web components technology, at least on the Java side of the world.

What is JavaServer Faces (JSF)?

Here is a formal definition that comes from Sun*: "A server side user interface component framework for Java™ technology-based web applications"*. This sums it up perfectly.

- Server Side
    - There are plenty of component frameworks, based on client-side JavaScript (read: in the browser). This is a dead end. The plain truth is that modern enterprise applications will be multi-tier, mostly server-side for the immediate future. Most (but of course, not all) functionality will be implemented on server.
- User Interface
    - Yes, JSF is about the creation of the user interface. Hopefully, Java is doing very well in business functionality implementation, but a UI is a different beast. It requires a special approach, best exemplified in JSF.
- Component framework
    - This is the core of JSF. It's all about how to define, package, distribute and use UI components. It comes with a set of standard ready-to-use components and, much more importantly, it defines an environment for a component ecosystem. Because of this, you can buy or download

somebody's UI components and expect them to fit in and be useful in today's project.

- Java Web applications
    - Is anybody making any other kind of applications today? OK, that's a joke, of course, but the fact is that the vast majority of applications, at least the enterprise type of application, are Web and Java-based.

Why JSF?

JSF is a great framework for a variety of reasons

- First of all, JSF is a great piece of technology.
- It is a clean implementation of the MVC design pattern for Web applications.
- It's easy to use.
- It applies a clean separation of roles for Web applications.
- It is extendable in terms of both component and as a framework.
- It supports multiply delivery channels.

But all of this is less important than the plain fact that JSF is the only **standard** Web **component** technology for Java. JSP, portlets, and servlets are not component-based. Struts (Do we need to mention Struts anymore?) is neither component-based nor standard. Did we mention huge industry support? Certainly nobody wants to stay with a big important application based on a funky open source framework that was invented and developed by one clever programmer! This is why IBM, Oracle, Sun and others are cultivating JSF, carefully selecting the best of the ideas for Web development, and paying big money to the best architects and developers to improve JSF for the rest of us.

Its position as a standard is doubly important because it allows JSF to fully realize its promise as a component-based framework. The chances are much better that components from different vendors will coexist in the same application when the technology is more standardized – a great boon for all enterprise applications developers!

The JSF Answer for the Web 2.0 Challenge

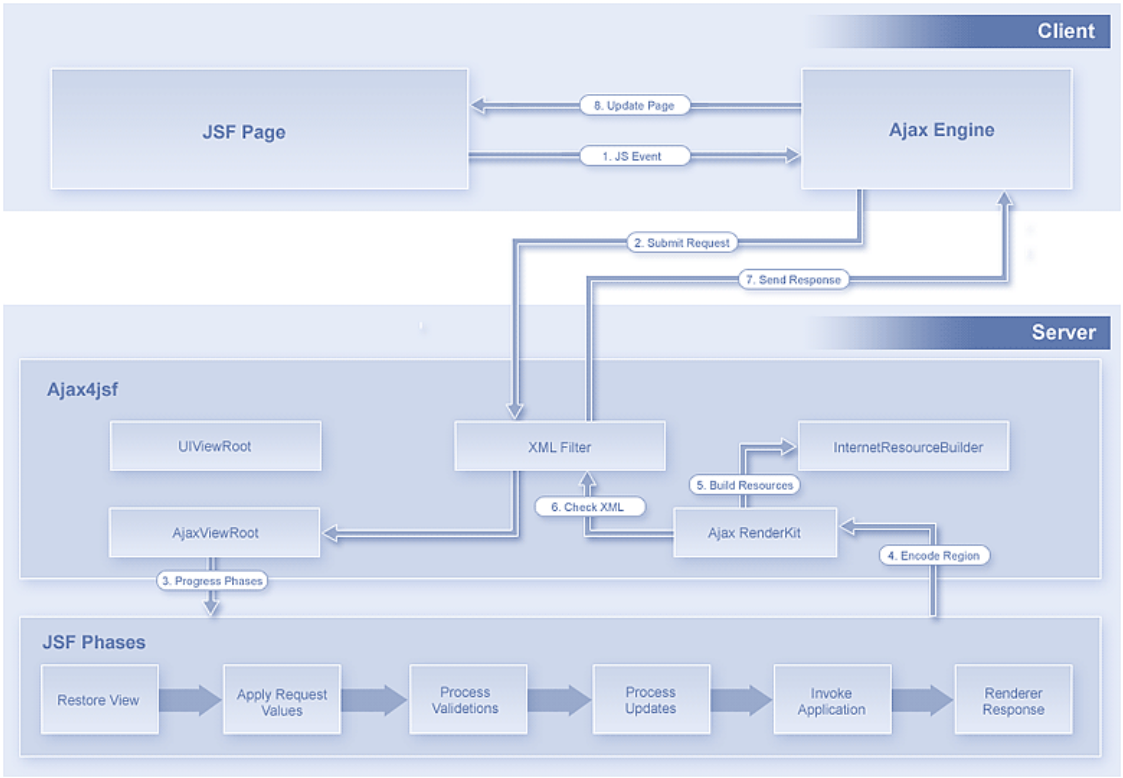
Despite the flood of "new" ways to solve modern challenges, at Exadel, we believe that JSF is the best answer to these challenges. Rich and dynamic should not equate to being less functional and too simple. When you deal with a real-world application, you must be ready for complicated page flows, data model, and business logic. To handle this complexity, JSF puts all of the needed tools and capabilities into the hands of developers. What JSF still needs is a little bit of "richness" to complete it. And, that is exactly what Ajax4jsf does.

The Birth of Ajax4jsf

Exadel is a very well known player in the JSF arena. Here, at Exadel, we believe in the future of JSF and we have provided our customers with the best tools for JSF from the very early days of JSF as a technology. We also strongly believe in Open Source. These two tendencies came together in Ajax4jsf. Exadel contibuted Ajax4jsf as an open source extension to JSF and continues to support and promote the Ajax4jsf project. It's true open source distributed under the Common Development and Distribution License (CDDL).

- Ajax4jsf is a 100% JSF-compliant library that adds a modern twist of AJAX to the power of JSF. The major goals for Ajax4jsf are:
- To combine two great forces JSF and AJAX, into one powerful package
- To leverage the full set of JSF benefits, including powerful server-side component-based functionality, for developing modern, interactive, rich, Web 2.0 applications
- To embrace openness and the true spirit of JSF to achieve the widest possible compatibility with different JSF implementations and third-party component libraries
- To add native support to great technologies complement JSF, like Facelets and JSF Extensions, to create a frameworks ecosystem around the JSF core

The picture below gives you a glimpse "under the hood" of how JSF and AJAX work together in Ajax4jsf:



What Can Ajax4jsf Do for You?

Ajax4jsf is a rich component framework that has been created exclusively to bring rich user interface functionality to the JavaServer Faces world. You can use Ajax4jsf to:

- Leverage the whole set of JSF benefits while working with AJAX
- Add AJAX capability to existing JSF applications
- Write your own custom rich components with built-in AJAX support
- Package resources with the application's Java classes
- Create an exciting look for the applications on-the-fly
- Create a modern rich user interface look-and-feel with skins-based technology
- Test components, actions, listeners, and pages as you create them

Leverage the Whole Set of JSF Benefits while Working with AJAX

Ajax4jsf is fully integrated into the JSF lifecycle. While other frameworks only give you access to the managed bean facility, Ajax4jsf leverages the action and value change listeners and invokes server-side validators and converters during the AJAX request-response cycle.

Add AJAX Capability to Existing JSF Applications

The framework is implemented using a component library that adds AJAX capability to your existing pages without needing to write any JavaScript code or needing to replace existing components with new AJAX widgets. Ajax4jsf enables page-wide AJAX support instead of the traditional component-wide support. This means you can define the event on the page that invokes an AJAX request along with the areas of the page that should be synchronized with the JSF Component Tree after the AJAX request changes the data on the server according to the events fired on the client.

Write Your Own Custom Rich Components with Built-in AJAX Support

An important part of Ajax4jsf is the Component Development Kit (CDK). The CDK includes both a code-generation facility and a templating facility using a JSP-like syntax.. These capabilities remove the need for much of the routine work involved in the component creation process. The component factory work like a well-oiled machine to allow the creation of first-class rich components with built-in AJAX functionality even more easily than the creation of simple components using the traditional coding approach.

Package Resources with the Application's Java Classes

In addition to its core AJAX functionality, Ajax4jsf also has advanced support for the management of resources like pictures, JavaScript code, and CSS stylesheets. The resource

framework makes it possible to easily pack such resources into Jar files along with the code of your custom components.

Create an Exciting Look for Applications On-the-fly

As yet another extra feature, the resource framework has a facility for generating images on-the-fly. With this feature, you can create images using the familiar approach of the Java graphic2D library. The function allows the creation of great looking, "slick" components whose look and feel can still be easily adjusted.

Create a Modern Rich User Interface Look-and-Feel with Skins-based Technology

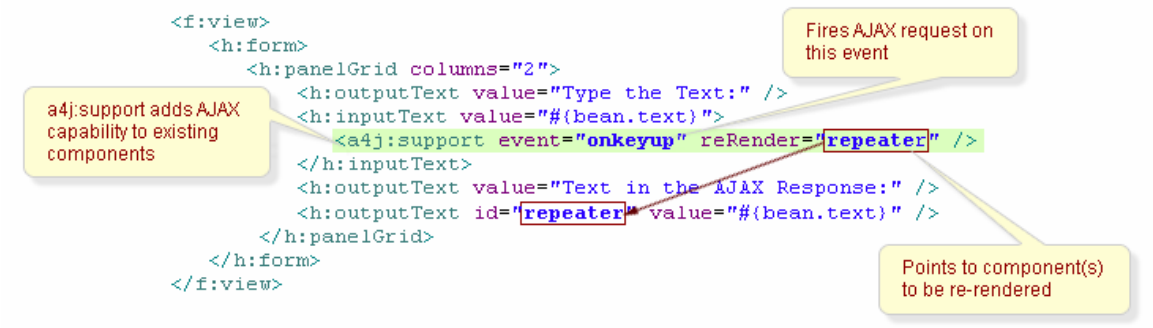
Ajax4jsf provides a skinability feature. This feature allows the easy definition and management of different color schemes and other characteristics of the UI through named skin parameters. You can access the skin parameters from both the JSP code and the Java code (for example, to adjust generated on-the-fly images based on the text parts of the UI). Note, though, that skinability is not a complete replacement for traditional CSS, but complementary to it

Test Components, Actions, Listeners, and Pages as You Create Them

An automated testing facility is in our roadmap for the near future. It will generate test cases for your component as you develop the components. The testing framework will not just test the components, but also any other server-side or client-side functionality including JavaScript code. As an additional convenience, it will do all of this without the need to separately deploy the test application into the servlet container.

Use Ajax4jsf Components

In addition to all of these great features, Ajax4jsf contains a set of JSF components that are essential for creating modern Web 2.0 applications. These components works right out of the box and allows dynamic "AJAX"-style interactions between the client and server sides of an application. Here is an example in code of using the Ajax4jsf "support" component to add AJAX behavior to the standard JSF "inputText" component:



Why Do You Need Ajax4jsf?

The answer to that question is simple (although it is more than one answer):

- If you need to develop modern, rich, dynamic, Web 2.0 applications using JSF, you need Ajax4jsf!
- No other solution combines such powerful functionality in a single package to add AJAX to your applications.
- No other solution allows you to combine the full power of JSF with a modern Web 2.0 interface.
- No other solution has such openness, such full compatibility with all standard JSF implementations and component libraries.
- No other solution gives you such flexibility in look and feel.
- No other solution gives you this level of power to deliver results.

Conclusion

A components-based approach using JSF and Ajax4jsf is the best way for enterprises to address the Web 2.0 challenge. To learn more about Ajax4jsf, go to Web site for the Ajax4jsf project site at https://ajax4jsf.dev.java.net or contact sales@exadel.com . On the project site, you can see examples of development scenarios and applications designed with Ajax4jsf technology.

About Exadel, Inc.

For more information on Exadel, product demos, and case studies, check out our Web site at www.exadel.com or email us at sales@exadel.com.

Founded in 1998, Exadel is a US-based company headquartered in Concord, CA. With more than 120 employees, 700 customers, 150,000 registered users of our products, and international offices in 3 countries, Exadel is one of the leading companies in products and implementations for enterprise, Web 2.0, JSF, AJAX, and open source-based applications.

Our Philosophy

Exadel's success in serving our customers is based on a few strongly held beliefs. These ideas have driven us from the very beginning of Exadel:

- **Be Customer-Driven:** Listen to your customer, and your customer will listen to you. All our products are based on our experience in real-life applications that we deliver to our customers. Our philosophy is to apply the knowledge and experience from our implementations to our products for the greater good of all our customers.
- **Be Open:** Be fair, and the customer will understand you. Our products do not have any hidden sticky spots and traps. Use it, if you love it. This has been our philosophy from the very beginning, and we will continue to stick to it.
- **Be Focused:** Nourish the fields that feed you. Our focus on JSF is an important part of our philosophy. Exadel is the best JSF company in the world, and now we add back exciting JSF products.
- **Be Committed:** Play hard. Everything that we do, we follow through on until the end. Our products must be the best, and we will not stop until they are the best. When we have won, we then choose another target and start a new game. Before it was Struts Studio, then JSF Studio, and now it is Ajax4jsf.