

ORACLE®



Java and JSF

JSF Has Never Been Just about Markup

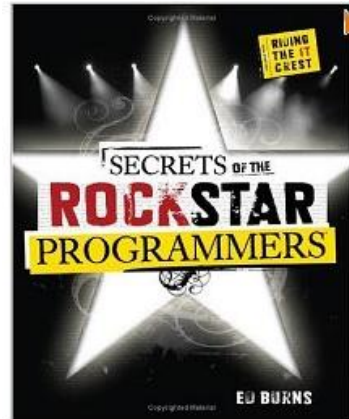
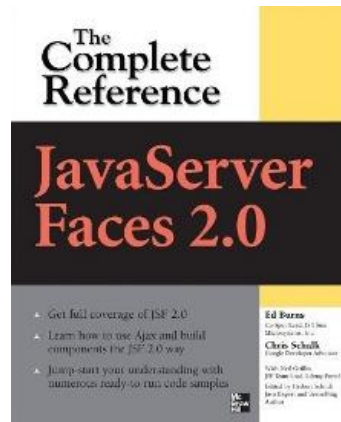
@edburns

JavaEE UI Technical Lead



Speaker Credentials

- JSF Spec Lead since 2002
- Author of two books





Latin America 2011

December 6–8, 2011

Tokyo 2012

April 4–6, 2012

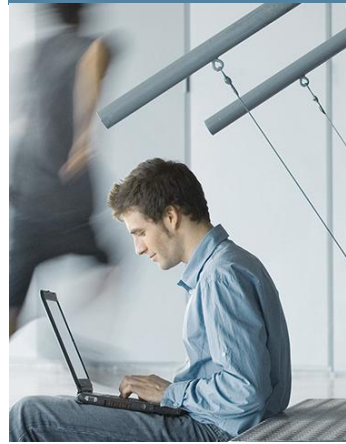
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

The following information is provided for your general product direction. It is intended for informational purposes only, and may not be incorporated into any contract. Oracle has no obligation or commitment to deliver any material, software, or service, and no obligation to update, upgrade, or otherwise modify the information on this page. Oracle shall not be relied upon in making purchasing decisions. Oracle reserves the right to change, modify, or discontinue the products, services, or features at any time without notice, and the timing of any features or products remains at the sole discretion of Oracle.

**ABSOLUTELY
BLEEDING
EDGE**

My Plan For Your Time Investment

- I thought JSF components were just tags?
- The design pedigree of JSF
- The idea behind JavaJSF
- The JavaJSF developer experience
- DEMO
- The afore mentioned JSF Tweaks
- Roadmap



I Thought JSF Components Were Just Tags?

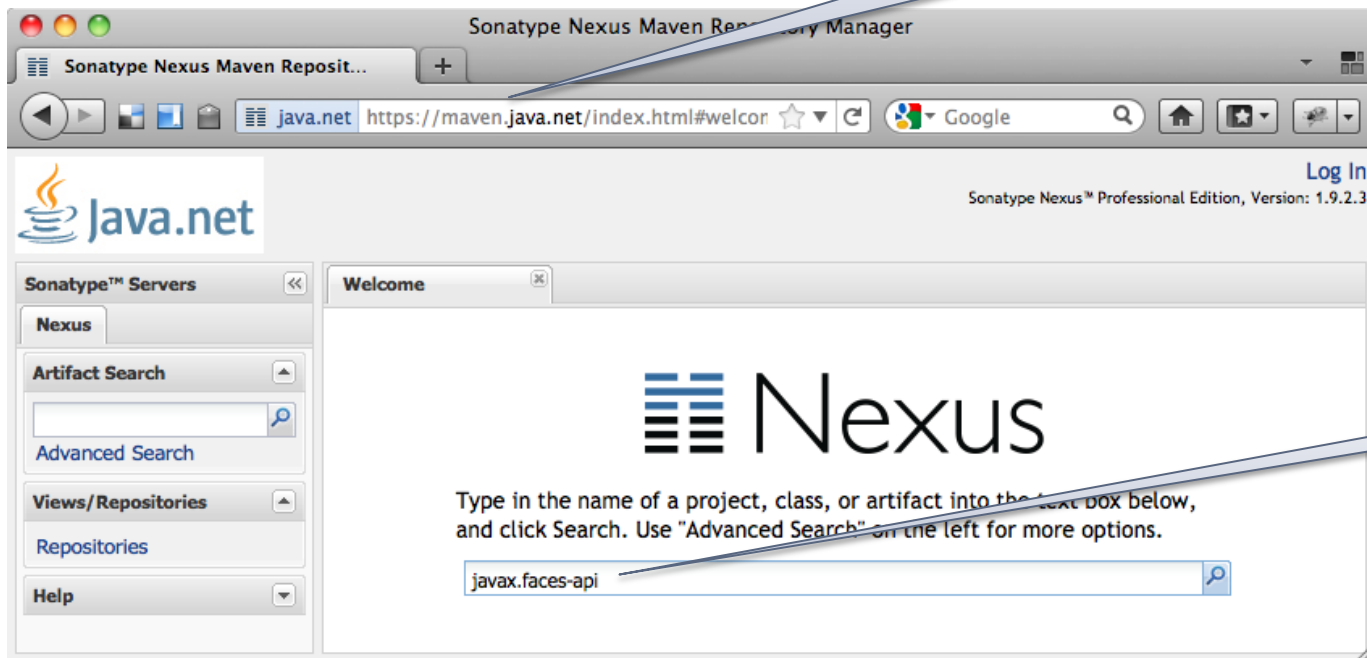
Think again

- The basic artifact in JSF is a Java class `javax.faces.component.UIComponent`
- JSF was designed to allow completely arbitrary ways of expressing how a bunch of things that extend from `UIComponent` make up a user interface
- Two standard View Declaration Languages in JSF: Facelets and JSP

I Thought JSF Components Were Just Tags?

Think again

- Let's take a look at the API <http://maven.java.net>



I Thought JSF Components Were Just Tags?

Think again

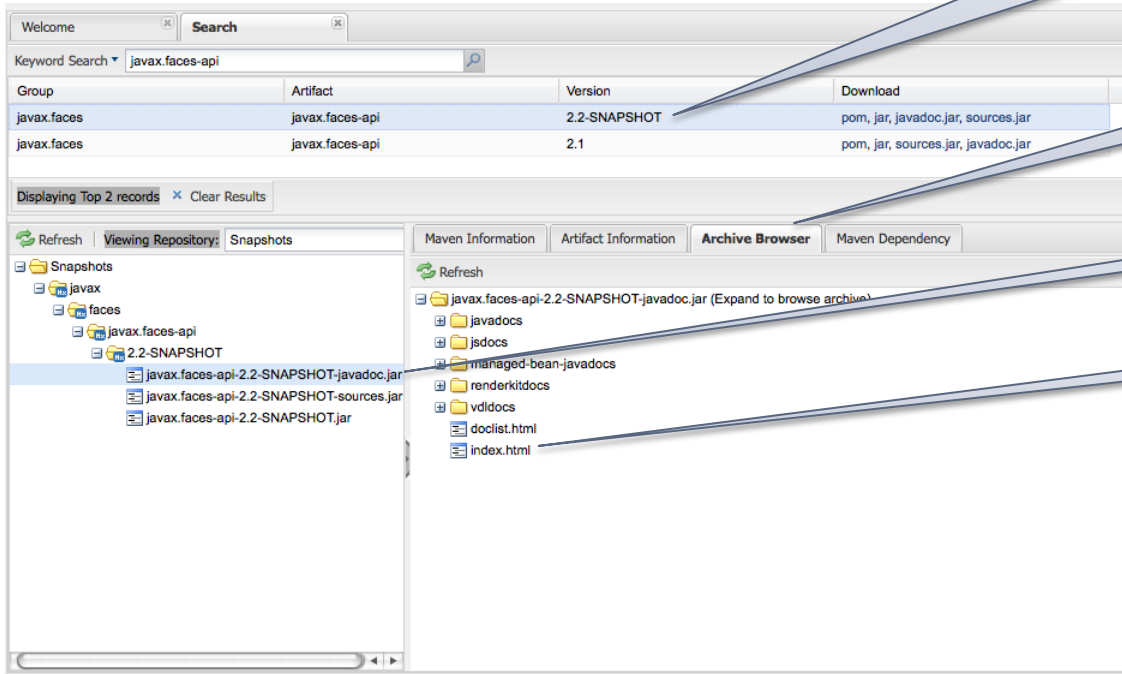
- Let's take a look at the API

2.2-SNAPSHOT

Archive Browser

-javadoc.jar

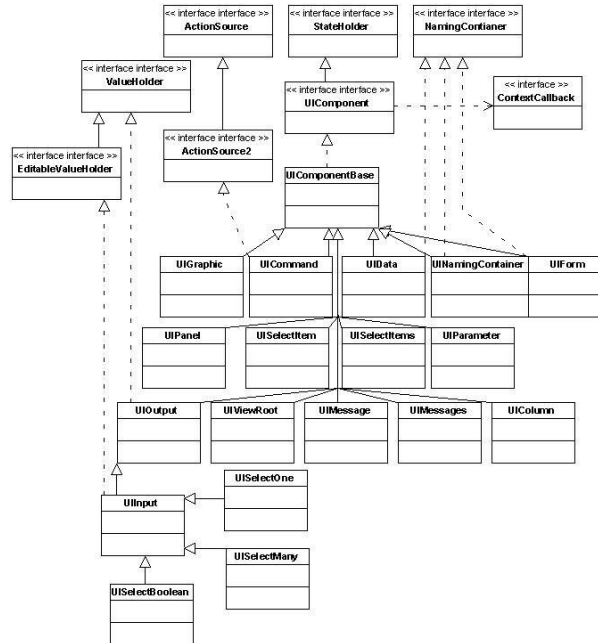
index.html



I Thought JSF Components Were Just Tags?

Think again

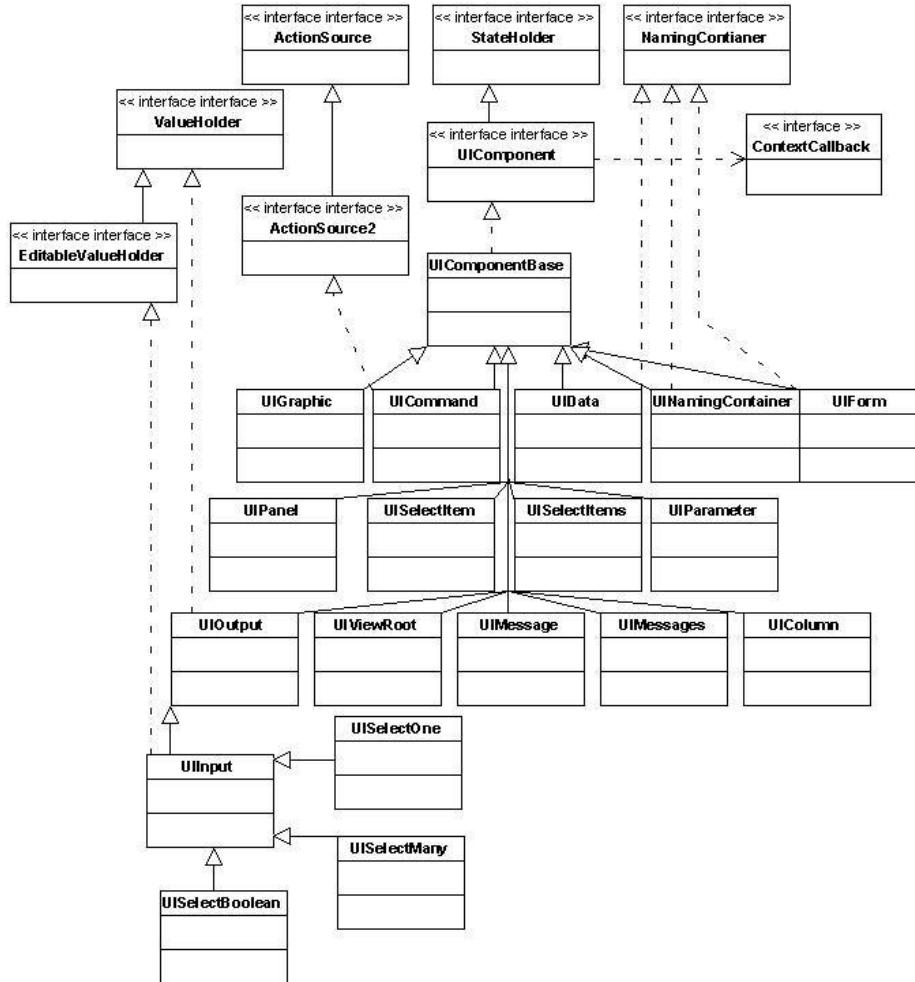
- Let's take a look at the API



I Thought

Think again

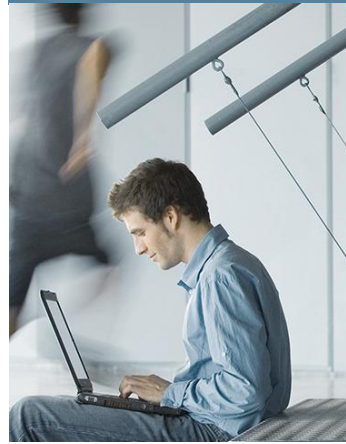
- Let's take a look at



Just Tags?

My Plan For Your Time Investment

- I thought JSF components were just tags?
- The design pedigree of JSF
- The idea behind JavaJSF
- The JavaJSF developer experience
- DEMO
- The afore mentioned JSF Tweaks
- Roadmap



The Design Pedigree of JSF

0xCAFEBAFE

- Amy Fowler



- Hans Bergsten



- In other words, the people that brought you swing

The Design Pedigree of JSF

0xBABB1E

- Craig McClanahan



- In other words, the struts guy

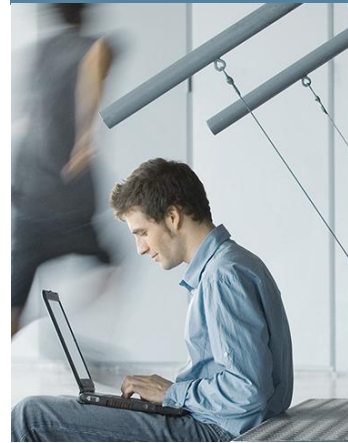
The Design Pedigree of JSF

Break it down

- Swing-like Java API
 - Components
 - Properties
 - Methods
 - Events and Listeners
- Struts like authoring experience
 - For a web framework, this turned out to be more popular.
 - Markup is an easier way to declare a UI

My Plan For Your Time Investment

- I thought JSF components were just tags?
- The design pedigree of JSF
- The idea behind JavaJSF
- The JavaJSF developer experience
- DEMO
- The afore mentioned JSF Tweaks
- Roadmap



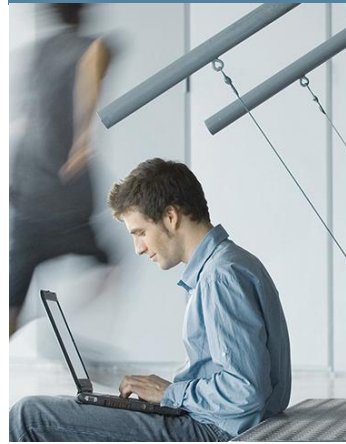
The Idea Behind JavaJSF

Take the popularity of Vaadin and use it to talk about JSF

- This talk is a SNAPSHOT result of an in-progress experiment
- Hypothesis
 - It should be possible to, with just small extensions to core JSF, make a very Vaadin-like authoring experience
 - If true, lots of power could be made available to developers with little new code

My Plan For Your Time Investment

- I thought JSF components were just tags?
- The design pedigree of JSF
- The idea behind JavaJSF
- The JavaJSF developer experience
- DEMO
- The afore mentioned JSF Tweaks
- Roadmap



The JavaJSF Developer Experience

Taken entirely from Vaadin + JSF

- Extend some base “Application” class
- Build the UI in the init() method
- Use event listeners to manipulate the UI dynamically

The JavaJSF Developer Experience

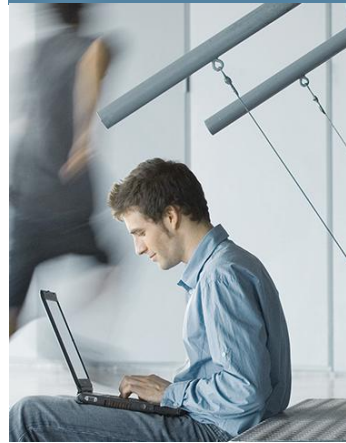
Getting the software

- Install Mojarra 2.2.0-SNAPSHOT into GlassFish 3.1 or later
- Use this archetype

```
mvn archetype:generate -DarchetypeGroupId=com.sun.faces -DarchetypeArtifactId=hellojavajsf-archetype  
-DarchetypeVersion=0.1-SNAPSHOT  
-DarchetypeRepository=https://maven.java.net/content/repositories/snapshots/
```

My Plan For Your Time Investment

- I thought JSF components were just tags?
- The design pedigree of JSF
- The idea behind JavaJSF
- The JavaJSF developer experience
- DEMO
- The afore mentioned JSF Tweaks
- Roadmap

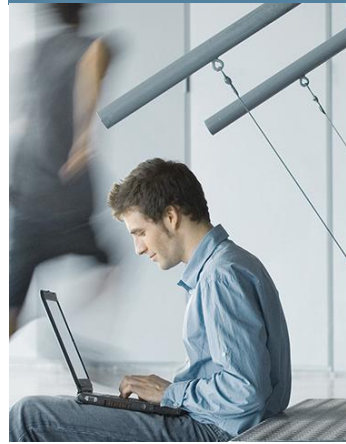


DEMO



My Plan For Your Time Investment

- I thought JSF components were just tags?
- The design pedigree of JSF
- The idea behind JavaJSF
- The JavaJSF developer experience
- DEMO
- The afore mentioned JSF Tweaks
- Roadmap



The Afore Mentioned JSF Tweaks

Look at my recent commits to Mojarra

- With `<https://svn.java.net/svn/mojarra~svn/trunk>` do
- `svn log -r {2011-09-15}:{2011-10-07} | egrep -A 2 'r[0-9]+ \\\ edburns \\\'`
 - r9381 Expose FaceletFactory and a wrapper for same
http://java.net/jira/browse/JAVASERVERFACES_SPEC_PUBLIC-611
 - r9398 Make it easier to instantiate a composite component from java
http://java.net/jira/browse/JAVASERVERFACES_SPEC_PUBLIC-599
 - r9400 Annotation for ResourceResolver
http://java.net/jira/browse/JAVASERVERFACES_SPEC_PUBLIC-1038

The Afore Mentioned JSF Tweaks

Oh, and the existing `com.sun.faces.spi`

- Annotation scanner SPI in GlassFish 3.0, 3.1, 3.1.1
- maven.java.net
 - `<dependency>`
 - `<groupId>com.sun.faces</groupId>`
 - `<artifactId>jsf-spi</artifactId>`
 - `<version>1.0-SNAPSHOT</version>`
 - `</dependency>`
- `com.sun.faces.spi.InjectionProvider`

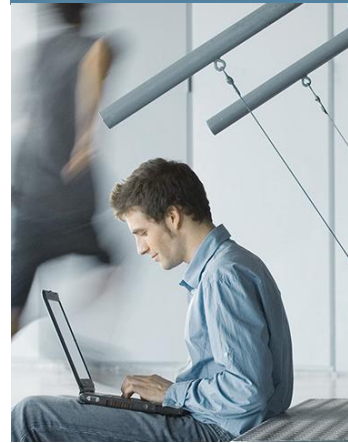
The Afore Mentioned JSF Tweaks

Key enablers already in JSF

- Facelets
- Composite Components
- Facets
 - Maps to things like `Window.getContent()`
- Existing Third Party Component Ecosystem

My Plan For Your Time Investment

- I thought JSF components were just tags?
- The design pedigree of JSF
- The idea behind JavaJSF
- The JavaJSF developer experience
- DEMO
- The afore mentioned JSF Tweaks
- Roadmap



Roadmap for JavaJSF



- Uses JSF 2.2 APIs
<http://weblogs.java.net/blog/edburns/archive/2011/09/26/try-out-mojarra-220-snapshot>
- Entirely depends on interest
- Source being developed at
<https://svn.java.net/svn/jsf-extensions~svn/trunk/code/run-time/javajsf>
- 0.1-SNAPSHOT out now
- Including Maven Archetype

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Q&A

ORACLE®