Crear un blog Acceder

# Software Development

Tuesday, August 21, 2012

# Increase your JSF application performance (Part 1 - Environment & Configuration)

- Always use the newest MyFaces version (As you can read here: Blog Entry)
- If you just need a plain servlet container, use Tomcat instead of Jetty
- Use JUEL as EL implementation
  - Add the newest JUEL API + implementation as depedency
  - Configure MyFaces to use JUEL:

- Increase expression cache in JUEL
  - Create src/main/resources/el.properties
  - Add property *javax.el.cacheSize* with a custom size. The default size is *1000*. In my application i use a size of *3000*.
- If you use CDI, consider to use OpenWebBeans as implementation and configure this in your web.xml:

- Enable MyFaces EL caching as described in MyFaces Wiki
- Disable JSP support in MyFaces:

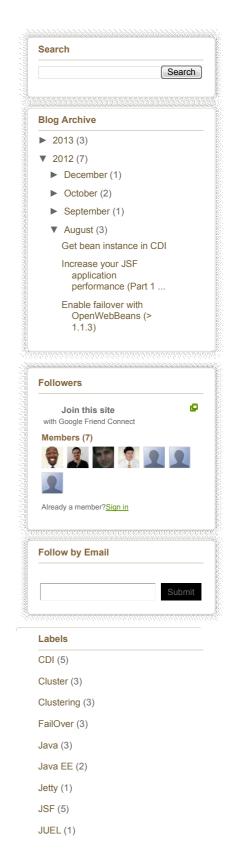
• Other params to increase performance:

- Configure state management as described in MyFaces Wiki
- Use a custom ServletFilter to set the correct expires/cache headers of your resources (images, stylesheets, javascripts)
- Compress and optimize your Javascripts in your build process. If you use maven, try primefaces-extensions' closure compiler maven plugin

Posted by Thomas Andraschko at 2:03 PM

+1 Recommend this on Google

Labels: CDI, JSF, JUEL, MyFaces, OpenWebBeans, OWB, Performance, Tomcat



## 8 comments:



#### smithh032772 November 6, 2012 at 8:43 AM

Thomas, thanks for this blog. I recognized it while reading one of your posts on PrimeFaces Core forum. I have taken most of your advice when it comes to JSF/Performance, I've already migrated to MyFaces Core, JUEL, I have added myfaces context-parameters in web.xml for performance reasons, I have read up on OpenWebBeans in last 12 hours, stayed up all night last night 'trying' to migrate from Glassfish 3.1.2.2 to Apache Tomcat 7.0.32, since I wanted to use PrimeFaces Push (atmosphere/websocket), but decided to postpone (for the 4th or 5th time after failure) using PrimeFaces Push with Glassfish 3.1.2.2, AND I decided to start migrating my app to CDI, initially the RI (weld), but OpenWebBeans (CDI) is in my sights. Reading about it now in Java EE 6 tutorial, and I think Ronald (in PrimeFaces Core forum) recommended someone else's blogs as well for using/coding CDI.

Reply

## Replies



#### smithh032772 January 4, 2013 at 9:55 AM

UPDATE: As you may already know, Thomas, some time within the last month, I completed migration from Glassfish 3.1.2.2 and JSF-managed-beans to TomEE1.5.1+ and CDImanaged-beans, and the web app is running well in production, on 6-to-10-year old hardware and software (Windows Server 2003 32-bit 4GB RAM). So, I'm using OpenWebBeans, and myself and the endusers are loving the web app.

So, thank you thank you thank you Thomas for any/all advice that you have offered!

Reply



## smithh032772 February 17, 2013 at 5:17 AM

Thomas,

Of course, I am revisiting this, because the only outstanding thing I left undone was the following:

Increase expression cache in JUEL

Create src/main/resources/el.properties

Add property javax.el.cacheSize with a custom size. The default size is 1000. In my application i use a size of 3000.

- 1. Can you update this blog and include code-tagged example of el.properties file  $\,$
- 2. Also, I had to re-read JUEL documentation and learned that I can place el.properties anywhere on my classpath; i didn't understand what you meant by src/main/resources/..., as that folder path does not exist in my project (using netbeans).

Reply



### Thomas Andraschko February 17, 2013 at 6:02 AM

Whats the problem? just add the property file with javax.el.cacheSize as key and the amount as value.

src/main/resources will be moved to WEB-INF/classes in the WAR file.

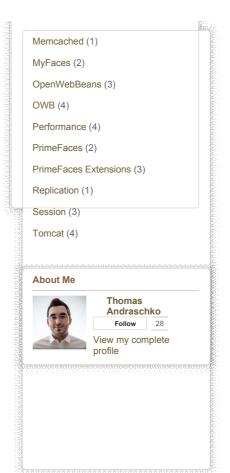
Reply

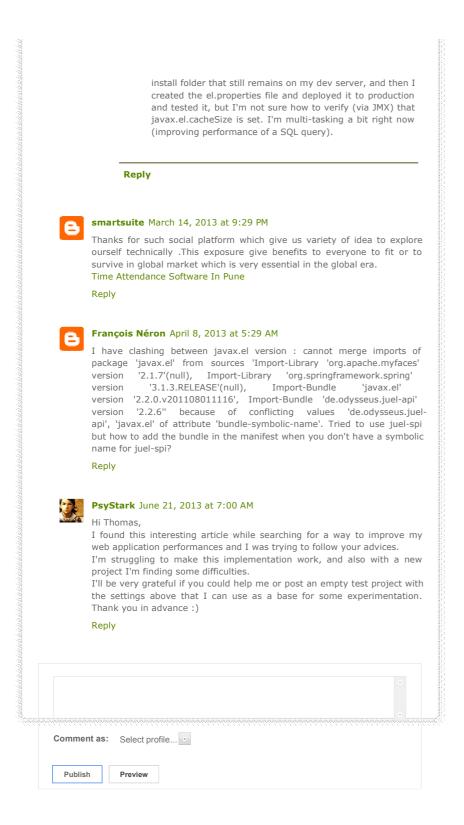
### Replies



## smithh032772 February 17, 2013 at 7:03 AM

Okay. I had to refer to a properties file I saw in glassfish  $% \left( 1\right) =\left( 1\right) \left( 1\right)$ 





- Mariana		
Newer Post	Home	Older Post
Newel Fost	HOHIC	Oluci i Ost

Subscribe to: Post Comments (Atom)

Software Development: Increase your JSF application performance (Part 1 - Enviro... Página 4 de 5

Powered by Blogger.

