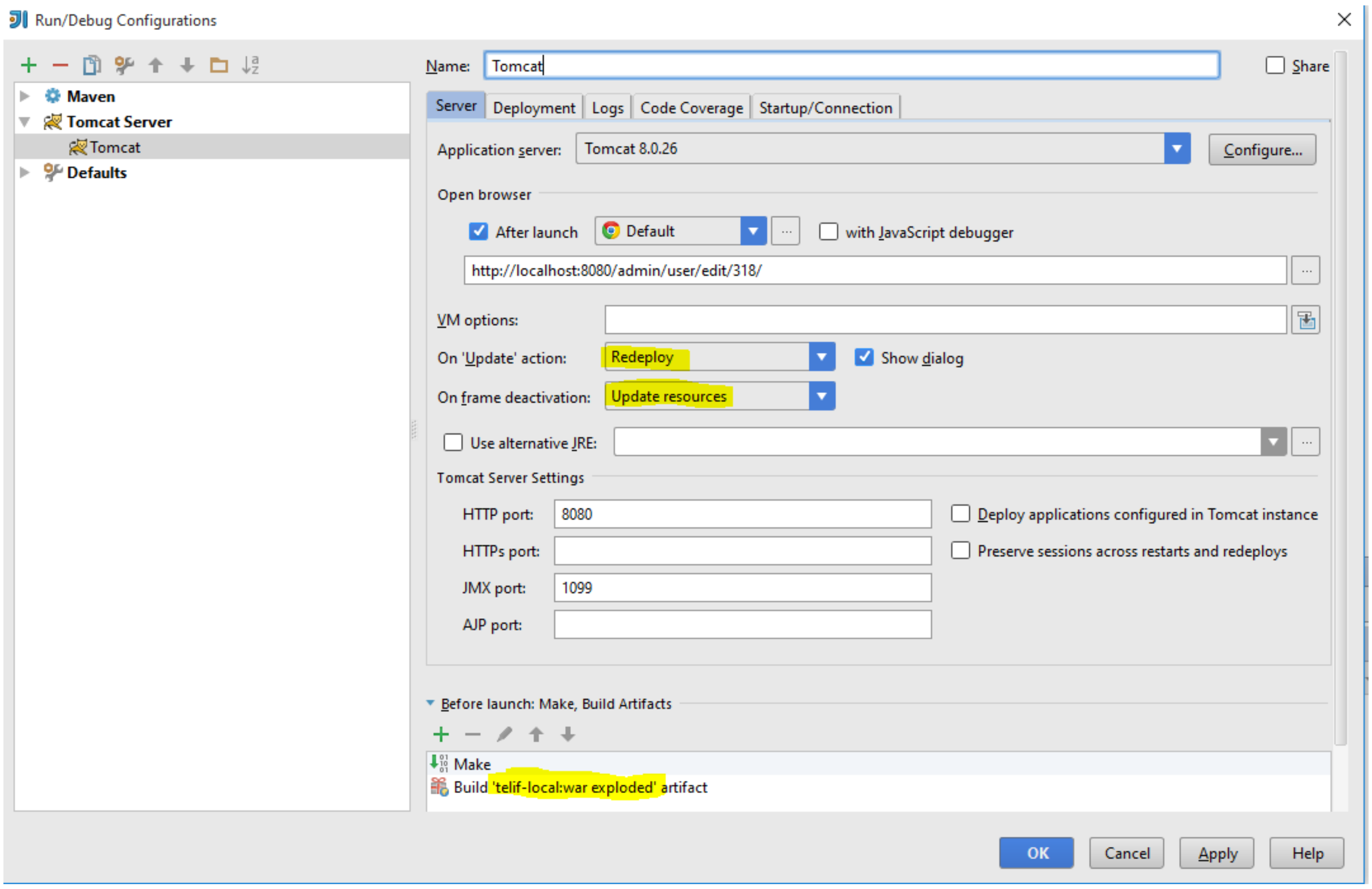
Tomcat Server Configuration in Intellij IDEA

Server Tab



Name: Tomcat

Application server: Tomcat 8.0.26

After launch: Default

http://localhost:8080/admin/user/edit/318/

VM options:

On 'Update' action: Redeploy — Show dialog

On frame deactivation: Update resources

Use alternative JRE:

HTTP port: 8080

HTTPs port:

JMX port: 1099

AJP port:

Before launch: Make, Build Artifacts

Make

Build 'telif-local:war exploded' artifact

As seen, in Server tab click **Configure** button located in Application Server line, then choose **Tomcat Server path**.

In the middle of that tab, change the "On frame Deactivation" setting to either "Update resources" or "Update Classes and Resources"

> Update resources: All changed resources (that is, all application components other than the classes) will be updated.
>
> Update classes: and resources. All changed resources will be updated; changed classes will be recompiled.

Note that whether the actual classes will be updated depends on the capabilities web server. If I recall, Tomcat will reload html/xhtml and jsp files, but not Servlets or classes that JSPs or Servlets use. You need to modified Tomcat to use a dynamic classloader for that.

You can also set the "On 'update' action as well.

> This determined what happens when you hit the Update Update icon icon (or Ctrl+F10) in the Run window.
>
> the "Show dialog" determines if IDEA prompts you each time you hit the update icon

Now anytime you make a change, IDEA will redeploy the changed file(s) when IDEA's frame is deactivated (i.e. loses focus). It does take a second or two, You'll see it in the lower status bar in IDEA. Obviously. you'll still need to refresh your web browser so it fetches the new file (unless of course if the page has an auto refresh of ajax like fetch).

Note: A good combination with Tomcat is to set "On frame deactivation" to "Update Resources" and the "On 'update' action to either "Redeploy" or "Restart Server". That allows static pages to be quickly updated via frame deactivation, and class updated via the 'update' action.

Note: A company named ZeroTurnaround sells JRebel which is a dynamic classloader solution. They also have a five part series on the subject that's very good.

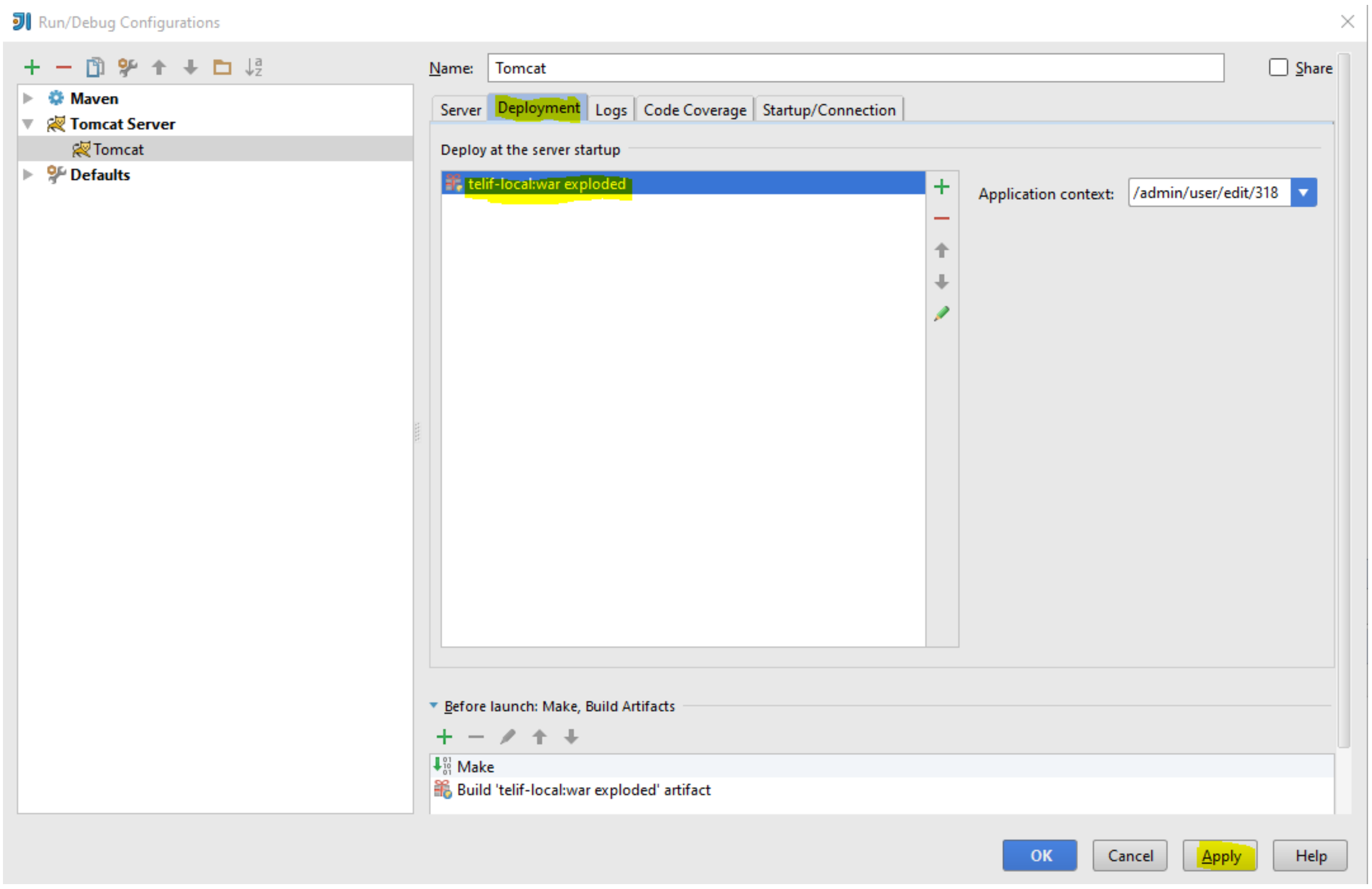
Note: If you want to use free classloader solution, please look at HotswapAgent

Note: If you want Tomcat to recognize changed files automatically, choose **war exploded** option.

In the "Deployment" section, remove (if present) the artifact my-webapp-name:war.

Deploy Tab

Click the add icon, select 'artifact' and then select my-webapp-name:war exploded


Run/Debug Configurations

Name: Tomcat

- Maven
- Tomcat Server
  - Tomcat
- Defaults

Server | Deployment | Logs | Code Coverage | Startup/Connection

Deploy at the server startup

telif-local:war exploded

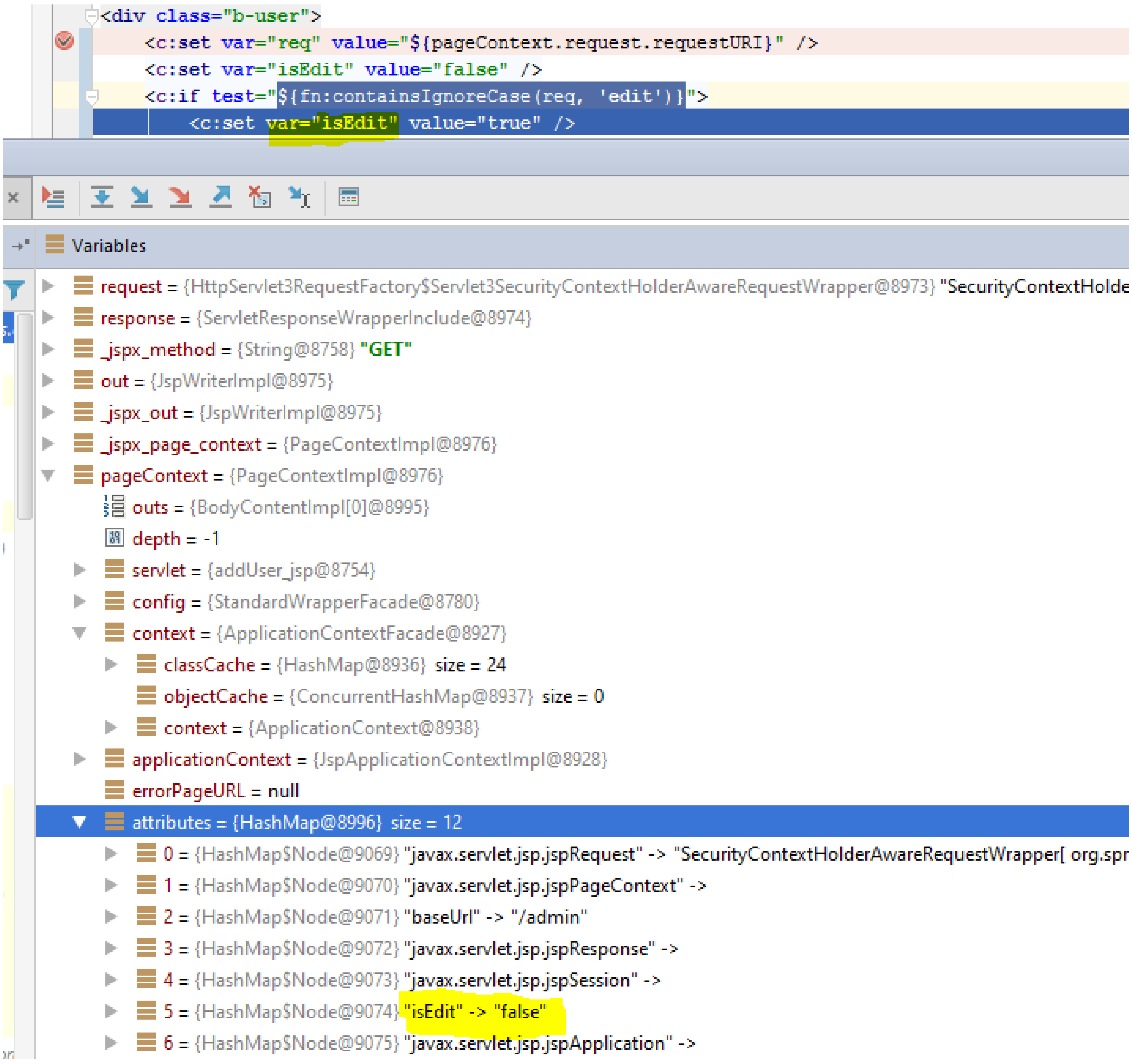
Application context: /admin/user/edit/318

Before launch: Make, Build Artifacts

Make
Build 'telif-local:war exploded' artifact

Show JSP Variables

To show JSP variables, click **pageContext** variable in the Debug frame, then click **attributes** variable.



```
<div class="b-user">
    <c:set var="req" value="${pageContext.request.requestURI}" />
    <c:set var="isEdit" value="false" />
    <c:if test="${fn:containsIgnoreCase(req, 'edit')}">
        <c:set var="isEdit" value="true" />
```

```
▶ request = {HttpServlet3RequestFactory$Servlet3SecurityContextHolderAwareRequestWrapper@8973} "SecurityContextHolde
▶ response = {ServletResponseWrapperInclude@8974}
▶ _jspx_method = {String@8758} "GET"
▶ out = {JspWriterImpl@8975}
▶ _jspx_out = {JspWriterImpl@8975}
▶ _jspx_page_context = {PageContextImpl@8976}
▼ pageContext = {PageContextImpl@8976}
      outs = {BodyContentImpl[0]@8995}
      depth = -1
   ▶ servlet = {addUser_jsp@8754}
   ▶ config = {StandardWrapperFacade@8780}
   ▼ context = {ApplicationContextFacade@8927}
      ▶ classCache = {HashMap@8936} size = 24
         objectCache = {ConcurrentHashMap@8937} size = 0
      ▶ context = {ApplicationContext@8938}
   ▶ applicationContext = {JspApplicationContextImpl@8928}
      errorPageURL = null
   ▼ attributes = {HashMap@8996} size = 12
      ▶ 0 = {HashMap$Node@9069} "javax.servlet.jsp.jspRequest" -> "SecurityContextHolderAwareRequestWrapper[ org.spr
      ▶ 1 = {HashMap$Node@9070} "javax.servlet.jsp.jspPageContext" ->
      ▶ 2 = {HashMap$Node@9071} "baseUrl" -> "/admin"
      ▶ 3 = {HashMap$Node@9072} "javax.servlet.jsp.jspResponse" ->
      ▶ 4 = {HashMap$Node@9073} "javax.servlet.jsp.jspSession" ->
      ▶ 5 = {HashMap$Node@9074} "isEdit" -> "false"
      ▶ 6 = {HashMap$Node@9075} "javax.servlet.jsp.jspApplication" ->
```