Dimensions Tutorial

Table of contents

Copyright © 2002-2005 Free2Be, Aaron Roller, Antonio Petrelli All rights reserved.

1. Quickstart

Installing and using Dimensions with basic features is pretty simple. Follow these steps to create a Dimensions-based application. Here we assume that you know how to create a Struts+Tiles based application.

1.1. Configuring Tiles

First of all you have to configure Tiles for the use of Dimensions. Simply cut and paste this piece of code instead of your inclusion of Tiles plug-in in `struts-config.xml`:

```
  <plug-in className="org.apache.struts.tiles.TilesPlugin">
    <set-property property="definitions-config"
value="/WEB-INF/dimensions-config.xml"/>
    <set-property property="moduleAware" value="true"/>
    <set-property property="definitions-parser-validate" value="false"/>
    <set-property property="definitions-factory-class"
      value="com.free2be.dimensions.tiles.ConfigurableFactorySet"/>
  </plug-in>
```

1.2. Decide your user-device matrix

Now this is a design step. You have to think about your user roles and device types.

For instance, suppose that you are building an eCommerce site, and you have the following user roles:

- Visitor, or the one who enters the site only to visit it.
- Customer, or the one who buys.
- The sales employee, i.e. the one who puts the prices on.

Now you have to identify all supported devices. For example:

- An HTML browser on a PC.
- An HTML browser on PDA.
- A WAP telephone, that supports WML.

The resulting matrix is a 3 x 3 table, maybe you wish only to cover some points in this table. For example, the sales employee could use a PC or a PDA, but not a WAP phone. Obviously this is up to you.

Suppose then that we have such a table.

*	PC	PDA	Phone
Visitor	X		X

Copyright © 2002-2005 Free2Be, Aaron Roller, Antonio Petrelli All rights reserved.

Customer	X	X	X
Employee	X	X	

1.3. Assigning Tiles configuration files

After making the table above, you should make another table, where you assign a Tiles configuration file to each user-device pair. For example:

*	PC	PDA	Phone
Visitor	tiles-defs.xml	N/A	tiles-defs_wml.xml
Customer	tiles-defs_customer.xml	tiles-defs_customer_html_pda	tiles-defs_customer_wml.xml
Employee	tiles-defs_employee.xml	tiles-defs_employee_html_pda	N/A

1.4. Specifying user identification

Now a question: how are you going to identify the user role in your application? It is really common that you have a bean put in session scope that contains information about the user (if it is logged in). And it is common that you have a property that identifies the role. Dimensions supports using this property, but it must be a string. So, if you don't have a bean in session scope that identifies the user, create it. If you have this bean but you don't have a property to identify the user's role, create it, and it must be a string.

1.5. Finding out default user role and device

A very important step is to identify your default user role and your default device. In most cases, the default user role is the one who does not need a login phase (in our example, the Visitor), while usually the default device is a HTML PC browser.

1.6. Configuring Dimensions

Finally, you can configure Dimensions. Dimensions uses a concept called "decision": the engine "decides" which is the most proper user-device pair that can be used by the user. You have to write a decision for each supported user-device pair

But let's see the final configuration file, that in our example it is `dimensions-config.xml`, we'll comment it below (we are assuming that you are going to put the configuration files in `/WEB-INF/config` directory).

```
<?xml version="1.0" encoding="UTF-8"?>
```


Copyright © 2002-2005 Free2Be, Aaron Roller, Antonio Petrelli All rights reserved.

```
<dimensions-config>
    <!-- Here we choose the "decider", i.e. the class that "decides" the
most
      correct user-device pair. -->
    <decider className="com.free2be.dimensions.decider.UserDeviceDecider">
        <!-- The bean that identifies a user is "loginInfo", in scope
"session",
         and the property that identifies the user's role is "role". -->
        <set-property name="userBeanName" value="loginInfo" />
        <set-property name="userBeanScope" value="session" />
        <set-property name="userBeanRoleProperty" value="role" />
    </decider>

    <!-- This is the class that resolves inheritances between Tiles
definitions files -->
    <inheritance-resolver
className="com.free2be.dimensions.tiles.inheritance.UserDeviceInheritanceResolver">
    </inheritance-resolver>

    <decisions>
        <decision>
            <!-- Default decision -->
            <definitions-config path="/WEB-INF/config/tiles-defs.xml" />
        </decision>
        <decision>
            <!-- Default user (visitor) with WAP phone. -->
            <parameter name="device" value="wml" />
            <definitions-config path="/WEB-INF/config/tiles-defs_wml.xml"
/>
        </decision>
        <decision>
            <!-- Customer with default device (HTML PC) -->
            <parameter name="userRole" value="customer" />
            <definitions-config
path="/WEB-INF/config/tiles-defs_customer.xml" />
        </decision>
        <decision>
            <!-- Customer on PDA -->
            <parameter name="userRole" value="customer" />
            <parameter name="device" value="html_pda" />
            <definitions-config
path="/WEB-INF/config/tiles-defs_customer_html_pda.xml" />
        </decision>
        <decision>
            <!-- Customer on WAP phone -->
            <parameter name="userRole" value="customer" />
            <parameter name="device" value="wml" />
            <definitions-config
path="/WEB-INF/config/tiles-defs_customer_wml.xml" />
        </decision>
        <decision>
            <!-- Employee with default device (HTML PC) -->
            <parameter name="userRole" value="employee" />
            <definitions-config
```


Copyright © 2002-2005 Free2Be, Aaron Roller, Antonio Petrelli All rights reserved.

```
path="/WEB-INF/config/tiles-defs_employee.xml" />
        </decision>
        <decision>
            <!-- Employee on PDA -->
            <parameter name="userRole" value="employee" />
            <parameter name="device" value="html_pda" />
            <definitions-config
path="/WEB-INF/config/tiles-defs_employee_html_pda.xml" />
        </decision>
    </decisions>
</dimensions-config>
```

The `<decider>` identifies the class that takes decision. In this case you don't have to change it, anyway see "Deciders" page to customize the way you take decisions. In section "Understanding UserDeviceDecider" we explain how it works.

This decider, analyzer the bean representing the user and the HTTP request, to identify the user's role and the calling device. The caracteristics of this bean are specified by `<set-property>` elements, where the properties mean:

- **userBeanName**: the name of the bean to use;
- **userBeanScope**: the scope in which the engine should search. It can be "application", "session" or "request". If it is not specified, the bean is searched across all scopes.
- **userBeanRoleProperty**: the name of the property of the bean that specifies the user's role (as a string).

The `<inheritance-resolver>` element specifies the class that is going to resolve inheritances between Tiles definitions files. In this case you don't have to change it, anyway see "Inheritance Resolvers" page to create your custom inheritance resolver.

A `<decision>` represents a decision. Each decision is characterized by parameters. In this case, there can be two parameters:

- **userRole**: the user's role;
- **device**: a string representing the device. For string representations of devices, see the section "Device string representations".

If a parameter in a decision is not specified, that it is intended the "default" parameter.

The `<definitions-config>` element inside a decision connects the decision itself to a Tiles definitions file. So that, each time that decision is taken, the specified Tiles definitions file is used.

1.7. Writing Tiles definitions files

Well, this section is up to you. I think a tutorial is useful, but an example is better, so download the simple example and test it.

Copyright © 2002-2005 Free2Be, Aaron Roller, Antonio Petrelli All rights reserved.

2. Understanding UserDeviceDecider

The "decider" is a class that analyzes something (for example the HTTP request or some bean values) and takes a decision, that contains a certain amount of parameter values.

The "UserDeviceDecider" is an implementation of such decider. The steps that it follows are:

- Find out the bean that represents the user and check which is the user's role, and if it can't, the "default" role is assumed.
- Analyze the header of the HTTP request and identify the most probable client device.
- Start from this "high" decision and descend it, to find out the best decision among the ones that are defined in `dimensions-config.xml` file.

But, how does UserDeviceDecider "descend" the decision stairs?

1. If the user is not the default one, go on. Otherwise, jump to point 3.
2. Starting from the most detailed string representation of the device, see if a decision with matching "userRole" and "device" parameters. If none is found, degrade the detail level of the device step by step. If it arrives at default device and there's still no decision matching, degrade to default user and go on. Otherwise, end the algorithm and take the decision.
3. Trying with a default user, the same way as it is done at point 2, until it arrives to default decision.

As you can see, the user's role has the first priority in taking the decision, simply because it is critical that a user sees the correct interface. So it degrades to default user only if all the device possibilities, even the default device, are tried.

3. Understanding UserDeviceInheritanceResolver

An "inheritance resolver" in Dimensions is a class that resolves inheritances between Tiles definitions described in different Tiles configuration files. It is used to avoid code repetitions, i.e. avoiding rewriting Tiles definitions in multiple files.

The class `UserDeviceInheritanceResolver` resolves inheritances giving priority to user's role, than it evaluates the device type. In other terms:

1. Starting from a Tiles definitions file, with a non-default user's role and a non-default device, to resolve inheritances it "degrades" the detail of the device, until it arrives to the default device, then it degrades the user's role, thus arriving to the default Tiles definitions file.
2. With the default user's role, it degrades the device, until it arrives to the default device, thus arriving to the default Tiles definitions file.

Copyright © 2002-2005 Free2Be, Aaron Roller, Antonio Petrelli All rights reserved.