

# Write your own decider

## Table of contents

1 Introduction.....	2
2 Why you would need a new decider.....	2
3 Extending the AbstractDeciderImpl class.....	2
3.1 Identify decision parameters.....	2
3.2 Implement "decide" method.....	2
3.3 Adding properties.....	3
4 Configuring Dimensions.....	3

## **1. Introduction**

Dimensions relies on the concept of a "Decider": this is a class that analyzes various sources that identifies the characteristics of the calling client and then takes a "decision", that indicates which is the most appropriate set of parameters among the ones specified in Dimensions configuration file.

The `UserDeviceDecider` is an implementation of this concept, that gives priority to the user, then to the device. And only user roles and device types are taken into consideration.

## **2. Why you would need a new decider**

If you stumbled in this section, maybe a simple decider that considers only a user's role and its calling device is not enough for you. You need to write your own decider if:

- You need some more customization parameters besides the user's role and the device. For example, the locale, or the user's **name**.
- The way the `UserDeviceDecider` takes decision is not appropriate for your needs, for example you will need to give priority to the device instead of the user.
- Many other cases that I am not able to think of, but comments are welcome :-P

## **3. Extending the `AbstractDeciderImpl` class**

If you want to write your own decider, the most easy way is to extend the `com.free2be.dimensions.decider.AbstractDeciderImpl` class. The other option is to implement the "Decider" interface in the same package, but the class above has a simple and useful implementation of common methods.

### **3.1. Identify decision parameters**

First of all, you have to identify all of your decision parameters. In `UserDeviceDecider`, the parameters are "userRole" and "device". For each parameter you have to assign it a meaning, that will never change, because it will be used in configuration files.

### **3.2. Implement "decide" method**

The most important thing to do is implementing the `decide` method. This class has as a parameter an `HttpServletRequest`, so you can access to the application space easily, and take your decision.

Once you reached a decision, you have to return an object of class `Decision`, setting all the needed parameters. The decision can be chosen among those present in `decisionSet`

*Write your own decider*

member. Notice that **if you want to use a default parameter, you must not set it! It is different to set a parameter to null than not to set it at all!**

### 3.3. Adding properties

In most cases you wish to add some customization properties to the decider, that can be configured in Dimensions configuration file. To add configuration properties, add "setX" methods (i.e. bean property mutators) to the class. These properties must be Strings. For example, suppose that you want to add a property called "sampleProperty", you have to add the "setSampleProperty" method.

## 4. Configuring Dimensions

Now you have to configure Dimensions. You have to modify the `dimensions-config.xml` file, to use your own decider.

```
<decider className="foo.bar.MyDecider">
  <set-property name="sampleProperty" value="sample value" />
  <set-property name="samplePropertyTwo" value="sample value two" />
</decider>
```

Remember to use only supported parameters in specified decisions!