

- TD 4 -

Interceptors and Timers

1. Profiling

We want to add to the application [SimplestWEA](#) the new logging and profiling features.

1.1. Logging

First, we want to log all method calls in order to identify the methods called most often and the busiest periods. We must provide a mechanism to obtain an output of the form

```
*** LOG: method bonjour in class firststeps.DireBonjourBean
        called Fri Nov 30 16:40:40 CET 2012 ***
```

1.2. Profiling

We have noticed that the methods of the bean [HelloWorldBean](#) are called quite often and we thus want to develop some profiling methods on the bean and, in particular, to analyze their execution time. So, we want to replace the messages for these methods with messages of the form:

```
*** PROFILING: greetingsMessage in class firststeps.HelloWorldBean@1f257fd
        called Fri Nov 30 17:18:51 CET 2012 ***
*** PROFILING: Method greetingsMessage from firststeps.HelloWorldBean@1f257fd
        executed in 2ms ***
```

We want to apply this type of profiling easily on other EJBs like, for example, the EJB [DireBonjourBean](#).

1.3. Verification

Some methods of the EJBs in the application assume that all parameters of type `String` are uppercase (for example, always EN, DE, etc.. for the language) but some customers are not aware of this convention. We should thus provide a mechanism to ensure that these parameters have the correct format, that is, a mechanism that allows to transform in uppercase all parameters of type `String` and which can be easily applied to several methods.

If the profiling mechanism is enabled for some methods then the parameter verification mechanism must operate differently: instead of transforming parameters to uppercase, it must just warn, if necessary, by a message in the console that some parameters are not uppercase. In this case we can obtain a display of the form:

```
*** LOG: bonjour in class class firststeps.DireBonjourBean
        called Sat Dec 01 10:33:42 CET 2012 ***
*** PROFILING: bonjour in class firststeps.DireBonjourBean@1f54a25
        called Sat Dec 01 10:33:42 CET 2012 ***
*** CAPITALIZE: Oups, parameters of bonjour are not in capitals !!!
*** PROFILING: Method bonjour from firststeps.DireBonjourBean@1f54a25
        executed in 1ms ***
```

Adapt the mechanisms previously proposed to account for this behavior. It must not be forgotten that these mechanisms should be enabled/disabled easily.

2. GoodbyeWorld

We want to change the bean `HelloWorldBean` to make it more polite. More precisely, if the method `sayHello` is called with "James Bond" as a parameter then we want to display the message "Hello James Bond" and 1 minute later the message "Goodbye James Bond".

In addition, every morning we want to greet everyone. The weekend we will do it at 10am, 10:20am, 10:40am and on the other days of the week at 8am, 8:20am, 8:40am, 9am, 9:20am, 9:40am. You must obtain messages of the form:

```
Good morning everybody !!!!
      It's Sat Dec 01 10:00:00 CET 2012
Good morning everybody !!!!
      It's Sat Dec 01 10:20:00 CET 2012
...
```

We also want to greet everyone in the evening but it is not that politeness is appreciated by those who like to go to bed early. We must provide a mechanism that allows us to enable and disable greetings interactively:



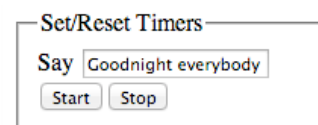
A control panel titled "Set/Reset Timers" containing two buttons: "Start" and "Stop".

If you use the Start button a message of the form

```
Goodnight everybody !!!!
      It's Sat Dec 01 22:00:00 CET 2012
```

is printed regularly until the Stop button is used and in this case messages are no longer displayed.

Eventually, we would like to be able to change the greeting message and therefore have an interface of the form:



A control panel titled "Set/Reset Timers" containing a text input field labeled "Say" with the text "Goodnight everybody" inside, and two buttons: "Start" and "Stop".

Modify existing EJBs and if necessary add more to implement these features.