- Pour Commencer
- Week 0: Introduction to Network and Service Management
- Week 1: Key Concepts with SNMP
- Week 2: Monitoring with Nagios
- **Week 3: Instrumentation with JMX**

Overview of the Content

Lecture 1: Key Concepts and Architecture
Lesson_Quiz

Lecture 2: Basic Instrumentation
Lesson_Quiz

Lecture 3: Support Services
Lesson_Quiz

Practical Exercise 1: JMX and JConsole
Practical_Exercise_Quiz

Practical Exercise 2: Standard MBean
Practical_Exercise_Quiz

Practical Exercise 3: Dynamic MBean

Attention : les exercices pratiques de cette semaine nécessitent la connaissance du langage Java (qui fait partie des pré-requis du MOOC). Si vous n'êtes pas familiers avec ce langage, vous pouvez simplement passer cette partie. N'oubliez pas cependant de répondre au quizz de fin de semaine.

PRACTICAL EXERCISE 2 (W3_PE2): STANDARD MBEAN

In this second practical exercice, you will extend the TimeServer project with a standard MBean that exposes the following attributes and operations:

- Number of available cities in the database of the server,
- Size of the threads pool used by the server,
- Set the size of the threads pool used by the server,
- Number of served requests,
- Number of unknown cities,
- Stop the timer server.

The Java interface of the required standard MBean is as following:

```
package fun.mooc.management.jmx.mbeans;

public interface TimeServerBaseMOMBean {
    // Number of available cities
    public int getNumberOfCities();
    // Size of the threads pool
    public int getSizeOfThreadsPool();
    // Set the size of threads pool
    public void setSizeOfThreadsPool(int size);
    // get the number of serverd requests
    public int getNumberOfRequests();
    // get the number of unknonw cities
    public int getNumberOfUnknownCities();
    // stop the timeserver
    public void stopServer();
}
```

- You have to create in the time server project a package named *fun.mooc.management.jmx.mbeans* where you put inside it the interface and the implementation classes of the MBean. For validation

4: Adding a Notification
Practical_Exercise_Quiz

Evaluations
Week_Evaluation
Echéance le avril 10, 2022 at 22:00 UTC

Aidez-nous à améliorer ce MOOC

- Week 4: Next-Generation Management Protocols
- Votre avis nous intéresse

- A template of the implementation class of the Standard MBean *TimeServerBaseMO* follows:

```
package fun.mooc.management.jmx.mbeans;

import javax.management.AttributeChangeNotification;
import javax.management.MBeanNotificationInfo;
import javax.management.NotificationBroadcasterSupport;
import fun.mooc.management.jmx.timeserver.ThreadPoolServer;

public class TimeServerBaseMO implements TimeServerBaseMOMBean{
        private static ThreadPoolServer server;

        public TimeServerBaseMO(ThreadPoolServer server) {
                this.server = server;
        }
        @Override
        public int getNumberOfCities() {
                return server.getNumberOfCities();
        }
        @Override
        public int getSizeOfThreadsPool() {
                /* To be completed ... */
        }
        @Override
        public void setSizeOfThreadsPool(int size) {
                int oldSize = server.getPoolSize();
                server.setSizeOfThreadsPool(size);
        }
        @Override
        public int getNumberOfRequests() {
                /* To be completed ... */
        }
        @Override
        public int getNumberOfUnknownCities() {
                /* To be completed ... */
        }
        @Override
        public void stopServer() {
                /* To be completed ... */
        }
}
```

- To register the standard MBean in the TimeServer.java you have to create an ObjectName and an instantiate your standard MBean and use the method registerMBean, as shown in the following code:

```
    stdName = new
ObjectName("fun.mooc.management.jmx.mbeans:type=TimeServerBa
seMOMBean");
    TimeServerBaseMO sMbean= new TimeServerBaseMO(server);
    mbs.registerMBean(sMbean, stdName);
} catch (InstanceAlreadyExistsException |
MBeanRegistrationException| NotCompliantMBeanException |
MalformedObjectNameException e) {
                e.printStackTrace();
}
```

- After coding your standard MBean which implements the Java interface (see above), you have to test your implementation using the script *validate.py* available in the folder */home/user/jmx/validation*.

Please note, that you have to execute the command: *source /usr/local/bin/set-jmx-lab-env.sh* in each new terminal to set correctly the lab environment variables.

Firstly, you have to run your modified version of the TimeServer project. Then, you have to execute the script in a terminal by executing the validation script with the exercise number: ***python validate.py 1***

The script will execute the TimeClient program 10 times and it will check if your standard MBean exposes correctly the required attributes and operations.

If the script returns the **code 200** as shown below, congratulation your implementation is correct.

```
The current time in New York is Sunday 10 September 2017, 10:59:38 AM
The current time in Villers-les-nancy is Sunday 10 September 2017, 16:59:38 PM
The current time in vandoeuvre-les-nancy is Sunday 10 September 2017, 16:59:38 PM
The current time in tatooine is Unknown city!
The current time in Java is Sunday 10 September 2017, 18:59:39 PM
The current time in moose is Unknown city!
The current time in Madrid is Sunday 10 September 2017, 16:59:39 PM
The current time in Los Angeles is Sunday 10 September 2017, 07:59:39 AM
The current time in toronto is Sunday 10 September 2017, 10:59:39 AM
OK. Your standard MBean is available.
OK. Expected attributes are available in func.mooc.management.jmx.mbeans:type=TimeServerBaseMOMBean
.
OK. Attributes have expected values.
OK. attribute SizeOfThreadsPool is writable.
Code 200: congratulation, well done!
```

If the script returns the **code 404**, it means that one or many attributes or operations are missing. If the returned code is **500**, it means that your MBean is badly implemented, typically you have a naming problem. If the returned code is **501,** it means that your TimeServer is not running.

Important: you should restart the time server in Eclipse before each execution of the validation script.

QUESTION W3.PE2.1 (1/1 point)

If your implementation is valid after running the script, you should obtain a 8 digits validation token, that you need to copy and paste in the answer box. What is the value of the validation token that you obtained?

72377361 ✔

72377361

Vous avez utilisé 1 essais sur 3

SOLUTION: TIMESERVERBASEMO.JAVA

```
package fun.mooc.management.jmx.mbeans;

import fun.mooc.management.jmx.timeserver.ThreadPoolServer;
```

```
private static ThreadPoolServer server;

public TimeServerBaseMO(ThreadPoolServer server) {

 this.server = server;

}


@Override

public int getNumberOfCities() {

 return server.getNumberOfCities();

}

@Override

public int getSizeOfThreadsPool() {

 return server.getPoolSize();

}

@Override

public void setSizeOfThreadsPool(int size) {

 int oldSize = server.getPoolSize();

 server.setSizeOfThreadsPool(size);

}

@Override

public int getNumberOfRequests() {

 return server.getNumberOfrequests();

}
```

```
 public int getNumberOfUnkownCities() {
  return server.getNumberOfUnkownCities();
 }
 @Override
 public void stopServer() {
  server.stop();
 }
}
```

SOLUTION: TIMESERVER.JAVA

```
package fun.mooc.management.jmx.timeserver;

import java.lang.management.ManagementFactory;

import javax.management.InstanceAlreadyExistsException;

import javax.management.MBeanRegistrationException;

import javax.management.MBeanServer;

import javax.management.MalformedObjectNameException;

import javax.management.NotCompliantMBeanException;

import javax.management.ObjectName;


import fun.mooc.management.jmx.mbeans.TimeServerBaseMO;

public class TimeServer {

 public static boolean searchCity = true;

 public static void main(String[] args) throws InterruptedException{
```

```
ThreadPoolServer server = new ThreadPoolServer(9000,helper);

// Start the MBean server

MBeanServer mbs = ManagementFactory.getPlatformMBeanServer();

ObjectName stdName = null;

try {

 stdName = new
ObjectName("fun.mooc.management.jmx.mbeans:type=TimeServerBaseMOMBean");

 TimeServerBaseMO sMbean= new TimeServerBaseMO(server);

 mbs.registerMBean(sMbean, stdName);

} catch (InstanceAlreadyExistsException | MBeanRegistrationException

  | NotCompliantMBeanException | MalformedObjectNameException e) {

 e.printStackTrace();

}

// Start the time server

Thread thServer = new Thread(server);

thServer.start();

thServer.join();

}

}
```