- 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 4 (W3_PE4): ADDING A NOTIFICATION

In this fourth exercise, you will extend the standard MBean of the exercise W3_PE2 with a notification support. The notification is triggered when the size of the threads pool has been changed, i the method *setSizeOfThreadsPool* has been called.

The template of the implementation of the standard MBean with the notification support follows:

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

```
import javax.management.AttributeChangeNotification;
import javax.management.MBeanNotificationInfo;
import javax.management.NotificationBroadcasterSupport;

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

public class TimeServerBaseMO extends
NotificationBroadcasterSupport implements
TimeServerBaseMOMBean{
    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 = /* To be completed */;
        server.setSizeOfThreadsPool(size);
        AttributeChangeNotification acn =
                new AttributeChangeNotification(this,0,0,"Threads
Pool size changed","SizeOfThreadsPool",
                        "int",oldSize,size);
        /* To be completed: check the documentation
     https://docs.oracle.com/javase/tutorial/jmx/notifs/index.html */
    }
    @Override
    public int getNumberOfRequests() {
        return server.getNumberOfrequests();
    }
    @Override
    public int getNumberOfUnknownCities() {
        return server.getNumberOfUnkownCities();
    }
    @Override
    public void stopServer() {
        server.stop();
```

```
                    return new MBeanNotificationInfo[] {
                                        new MBeanNotificationInfo(
                                                  new String[] {
AttributeChangeNotification.ATTRIBUTE_CHANGE },

AttributeChangeNotification.class.getName(),
                                                  "This notification is emitted when the size of
the threads pool changed.")
          };
     }
}
```

To validate your work, you have to execute the validation script, available in the folder */home/user/jmx/validation*, as following:

python validate.py 3

If the script returns the **code 200**, so you succeeded the third exercise.

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.

QUESTION W3.PE4.1 (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