



Haute Ecole d'Ingénierie et de Gestion  
du Canton de Vaud

***Protocoles d'Applications (PDA)***  
***Laboratoire 1 (HTTP)***

---

Olivier Liechti  
October 10, 2012

# Contents

<b>1</b>	<b>Guidelines</b>	<b>2</b>
<b>2</b>	<b>Setting a multi-tiered application environment for development</b>	<b>3</b>
2.1	Lab Task 1: understand the architecture of the MonSys system . . . . .	3
2.2	Lab Task 2: setup of the development environment . . . . .	4
<b>3</b>	<b>Setting a multi-tiered application environment for production</b>	<b>5</b>
3.1	Lab Task 3: design of the production environment . . . . .	5
3.2	Lab Task 4: setup of the production environment . . . . .	5
3.3	Lab Task 5: setting up session stickiness (optional) . . . . .	6

# 1 Guidelines

- The lab can be done either individually or in teams of two students. The laboratory will be evaluated after a demonstration by each team on Wednesday, October 2nd.
- Questions have to be answered in english.
- The answers should be packaged in a zip file named `lastname1_lastname2.zip`, with the following structure:
  - `doc`: folder containing your answers and detailed explanations, in one file named `lastname1_lastname2.pdf`. Please make sure to use lab task numbers as section headings in your text.
  - `config`: folder containing the configuration files related to the lab tasks; the files should be organized in sub-folders, where the name of the sub-folder corresponds to the lab task (e.g. `src/task2.2`).
  - `tests`: folder containing the test scripts (shell scripts, jmeter scripts, etc.)
  - `screenshots`: folder containing the raw screenshots; please also include the screenshots directly in the main document.
- Please send your files to `olivier.liechti@heig-vd.ch` until October 18th (8 AM) at the latest.

## 2 Setting a multi-tiered application environment for development

*MonSys* is a system, which was developed to monitor IT infrastructures. Its goal is to provide a dashboard to system administrators, providing a consolidated view of the state of all sorts of *nodes* in the infrastructure. Physical servers, virtual machines, printers are all examples of such nodes.

You have been asked to create both the development environment (running on each developer's machine) and the production environment (which must be *scalable* and *highly available*) for *MonSys*.

### 2.1 Lab Task 1: understand the architecture of the MonSys system

*MonSys* has been designed as a RESTful multi-tiered system. The application components are organized in two layers. The first layer (the *back-end*) hosts the business logic. *MonSys* nodes are seen as *resources* and their state can be retrieved with HTTP GET requests. Implemented in Java EE (version 6), the *MonSys* back-end can be deployed in an application server such as Glassfish. The back-end has a simple web UI (implemented with servlets and JSPs) used mostly for configuration purposes and which should be accessible only in the intranet). The second layer (the *front-end*) hosts presentation logic. Its role is to retrieve the state of nodes and to create a visual representation for it. The front-end implemented in PHP and it uses the JQuery javascript library. The *MonSys* front-end sends AJAX requests to the *MonSys* back-end in order to retrieve the state of nodes.

**Lab Task 1.1** Create an architecture diagram that represents the architecture for the *MonSys* system. Make sure to represent *middleware components* (in other words, the software that you will need to install in order to deploy *MonSys* components). Make sure to also represent the *MonSys application components*. Document the network aspects of the system (protocols, ports). You will also have a *reverse proxy* in your setup. The diagram must be clear and consistent.

**Lab Task 1.2** Describe what is AJAX and give one advantage to this approach. In this context, describe what is the so called *same origin policy*. This is an important point, which will require special setup for the development environment.

**Lab Task 1.4** Make the list of all the components that you will need to install and deploy in order to setup the development environment and to deploy *MonSys*.

**Lab Task 1.5** Make the list of tasks that you will need to perform in order to do the setup and deployment. Present and discuss this task list with us in order to validate it (have your architecture diagram ready for that discussion).

---

## 2.2 Lab Task 2: setup of the development environment

The goal of this task is to have the *MonSys* up and running on each student's machine. This will be done by completing the tasks defined in the previous step.

**Lab Task 2.1** Do the setup and write notes about the issues that you encountered and how you solved them. Keep track of the references (documentation, blogs, etc.) that you used to find the information.

**Lab Task 2.2** In order to validate the installation, use a web browser to access the *MonSys* back-end. You will see a simple web page, where you will be able to enter your name and a *Request code* button. Hit the button and take a screenshot of the result (make sure to have the browser *location bar* in the screenshot). Copy the value of the UUID, Time, Name and Code fields in your report.

**Lab Task 2.3** In order to validate the installation, use a web browser to access the *MonSys* front-end. You will see a simple web page, where you should see the list of nodes monitored by the system. Take a screenshot of the page (make sure to have the browser *location bar* in the screenshot).

**Lab Task 2.4** In order to validate the installation, write a script that uses a command line tool (such as `curl` or `wget`) in order to retrieve both an XML and a JSON representation of the resources managed by *MonSys*. Make sure to include both the script and the results both in your report and in the `test` folder in your delivery package).

**Lab Task 2.5** Describe how you have configured apache in order to complete the installation (and make sure that you provide your configuration file in your delivery package).

**Lab Task 2.6** Review the architecture diagram created in Task 1.1 and make the required modifications, so that the diagram accurately represents your setup.

## 3 Setting a multi-tiered application environment for production

Once you have successfully completed the development environment, you will *design* the production environment (note: you will need two machines to do the setup). You want the system to be highly available (in other words, users should be able to use *MonSys* even after a hardware server failure). You also want the system to be scalable. If the user population grows, it should be possible to add new hardware servers to host additional instances of the *MonSys* front-end and back-end components. To achieve these goals, you will use apache to create a software load balancer.

### 3.1 Lab Task 3: design of the production environment

**Lab Task 3.1** Create an architecture diagram that represents the architecture for the *MonSys* system. Make sure to represent *middleware components* (in other words, the software that you will need to install in order to deploy *MonSys* components). Make sure to also represent the *MonSys components*. Document the network aspects of the system (protocols, ports). Make sure to represent the physical components (in other words, the servers) and provide two versions of the diagram: one version that shows the target production environment and one version that shows the environment that you will setup during the lab.

**Lab Task 3.2** Explain what *session stickiness* is in the context of load balancing. Will you configure stickiness in your setup? If yes, why and how. If not, why is it not required and what how you set it up if required?

**Lab Task 3.3** Make the list of tasks that you will need to perform in order to do the setup and deployment. Present and discuss this task list with us in order to validate it (have your architecture diagram ready for that discussion).

### 3.2 Lab Task 4: setup of the production environment

The goal of this task is to have a (simplified) production environment *MonSys* up and running on a group of two machines.

**Lab Task 4.1** Do the setup and write notes about the issues that you encountered and how you solved them. Keep track of the references (documentation, blogs, etc.) that you used to find the information.

---

**Lab Task 4.2** In order to validate the installation, write JMeter scripts. Describe what these scripts are doing, take screenshots of the results and comment them. Make sure to include the scripts in your delivery package.

### 3.3 Lab Task 5: setting up session stickiness (optional)

Very often, applications that are built on a similar architecture have the need for application-level sessions in the front-tier. When this is the case, the load balancing must be done in a specific way. When they send the first request, users should be directed to any server in the cluster. When they send subsequent requests, users should always be directed to this server.

**Lab Task 5.1** How would you modify the *MonSys* front-end in order to be able to test such a scenario? Describe your approach and explain how you plan to test it. Do the implementation.

**Lab Task 5.2** Modify the configuration of the apache load balancer and describe the issues that you have encountered. Provide the configuration file.

**Lab Task 5.3** Describe your test method, document your observations and your interpretation.