

Objective = a Linux bash file as a web-service

In this tutorial you will learn how to

transform a Linux bash file into a web-service



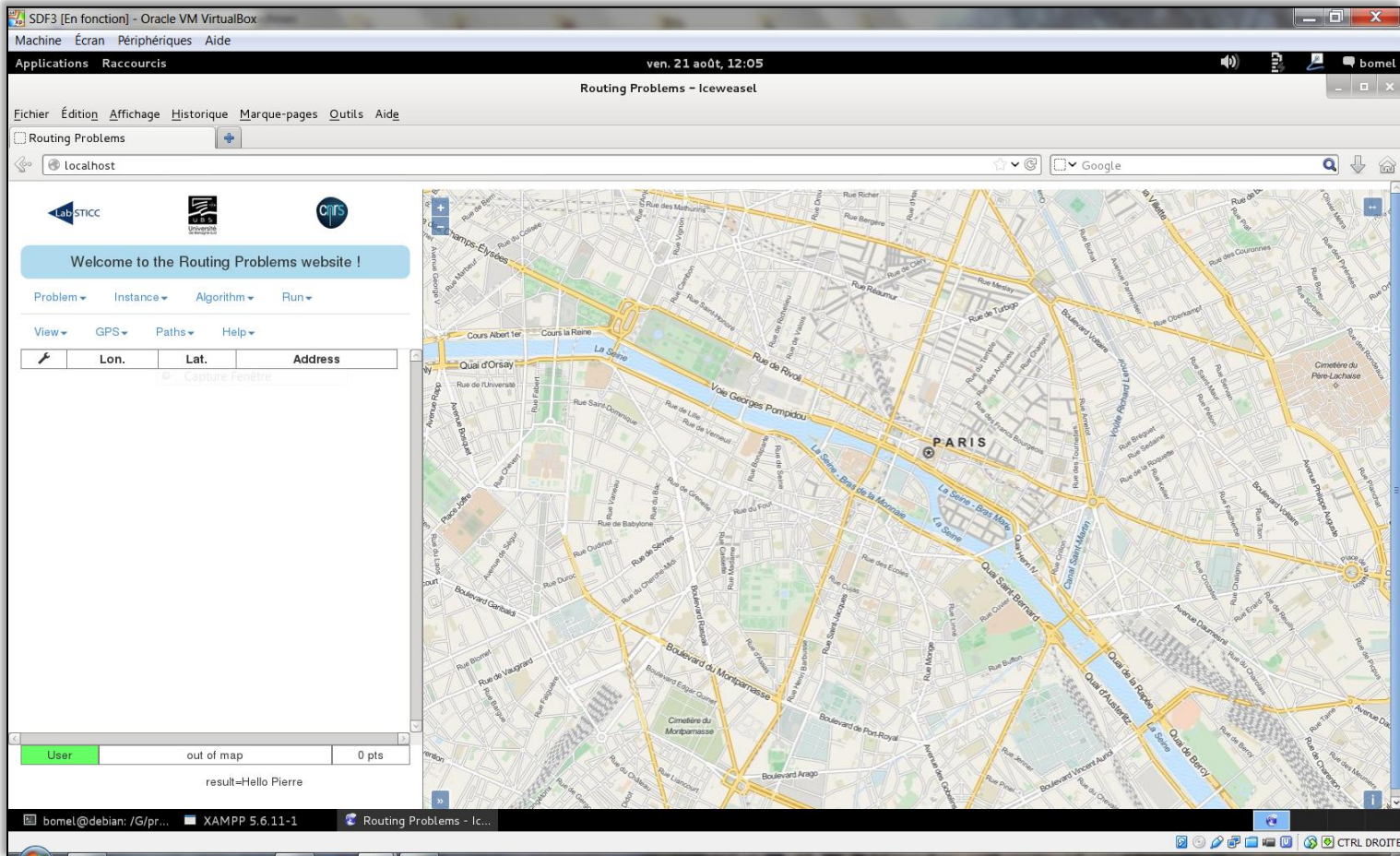
We suppose you are using Firefox, Chrome or IE.

We suppose you have a running, and well-configured, apache server on your PC, under Linux.

*There is another tutorial ($Y=AX+B$), which is very detailed.
So, this time, we won't go into much GUI's details.*

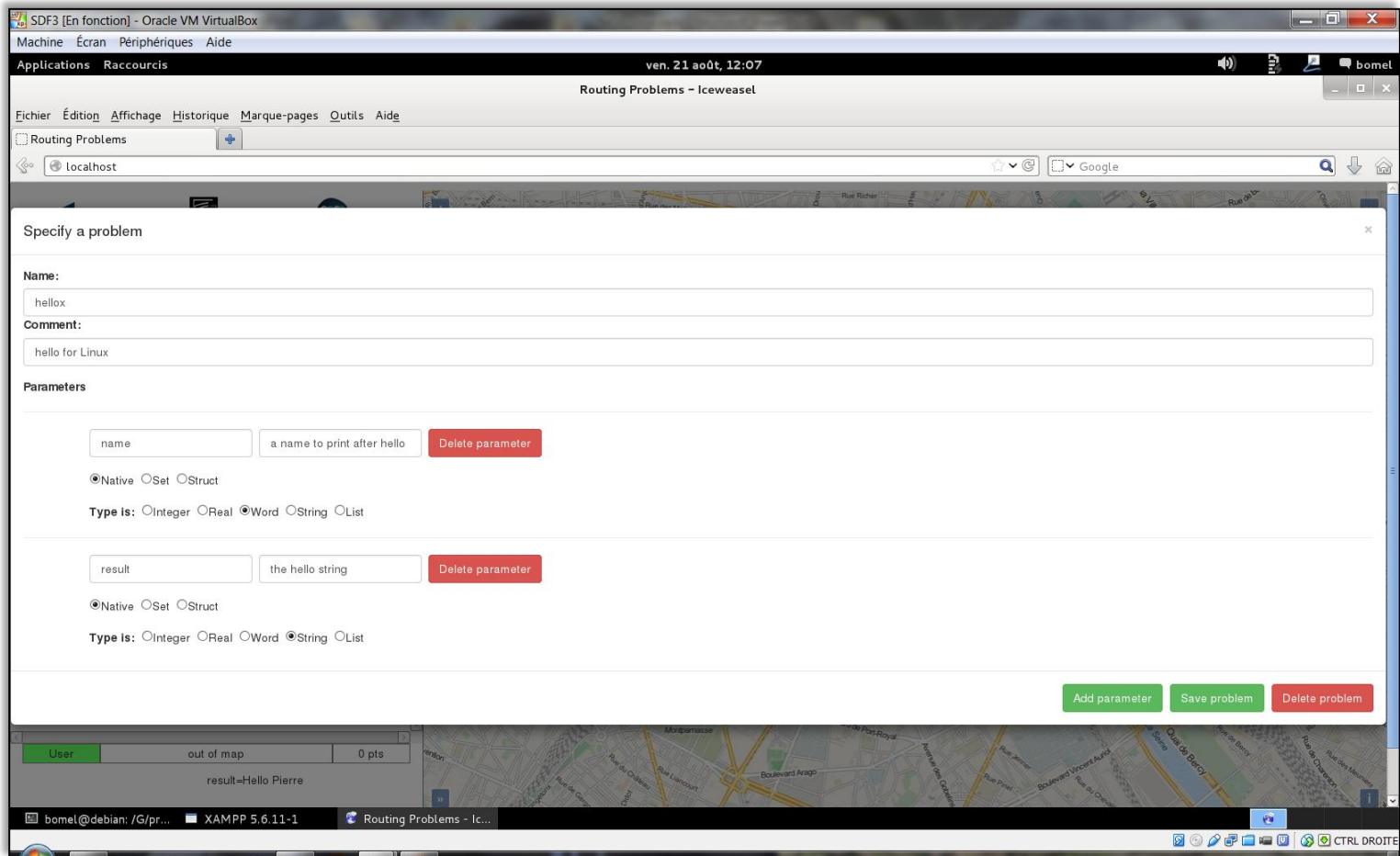


Start a browser with the URL "localhost"



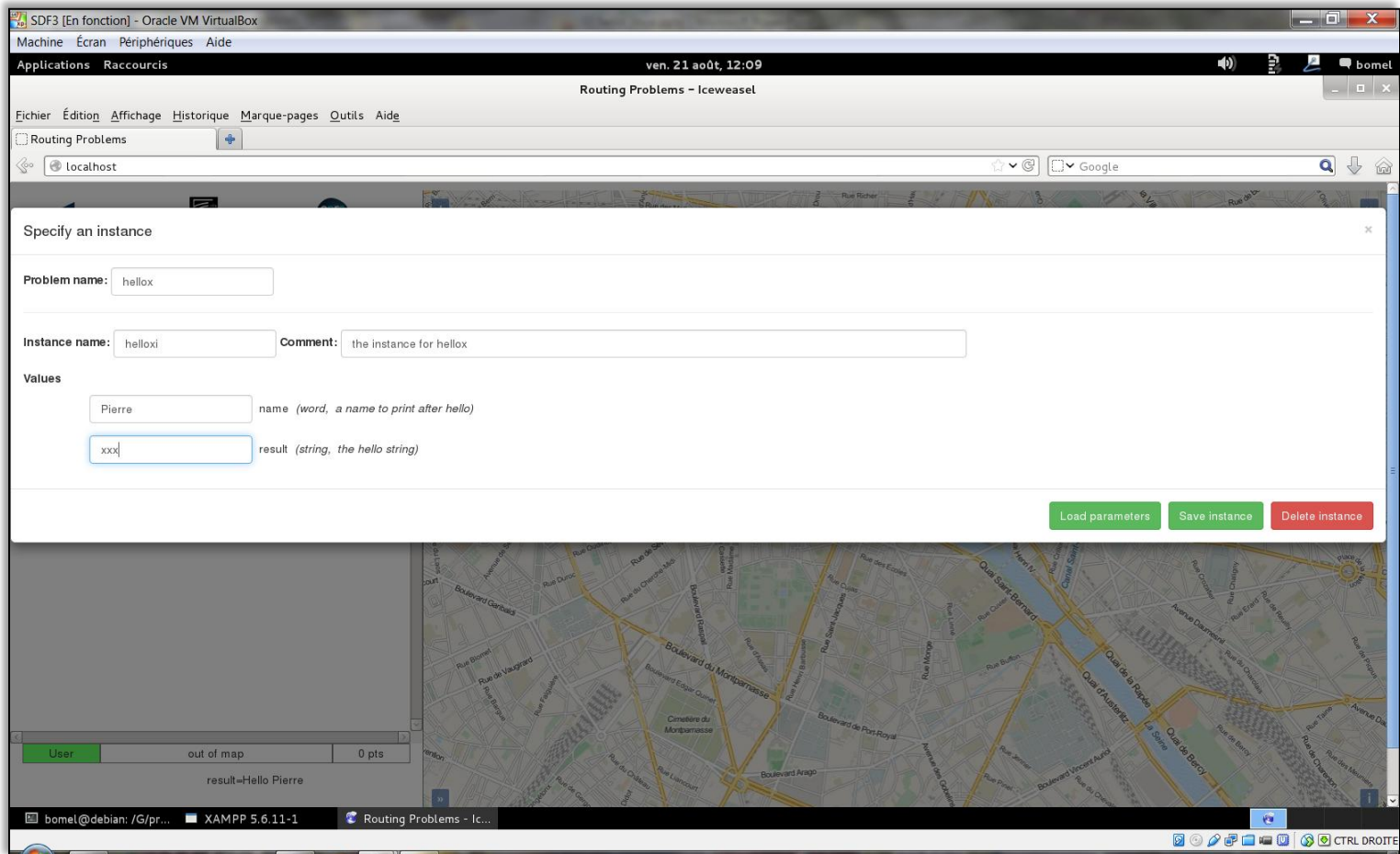
You should see exactly this.

Add a new problem

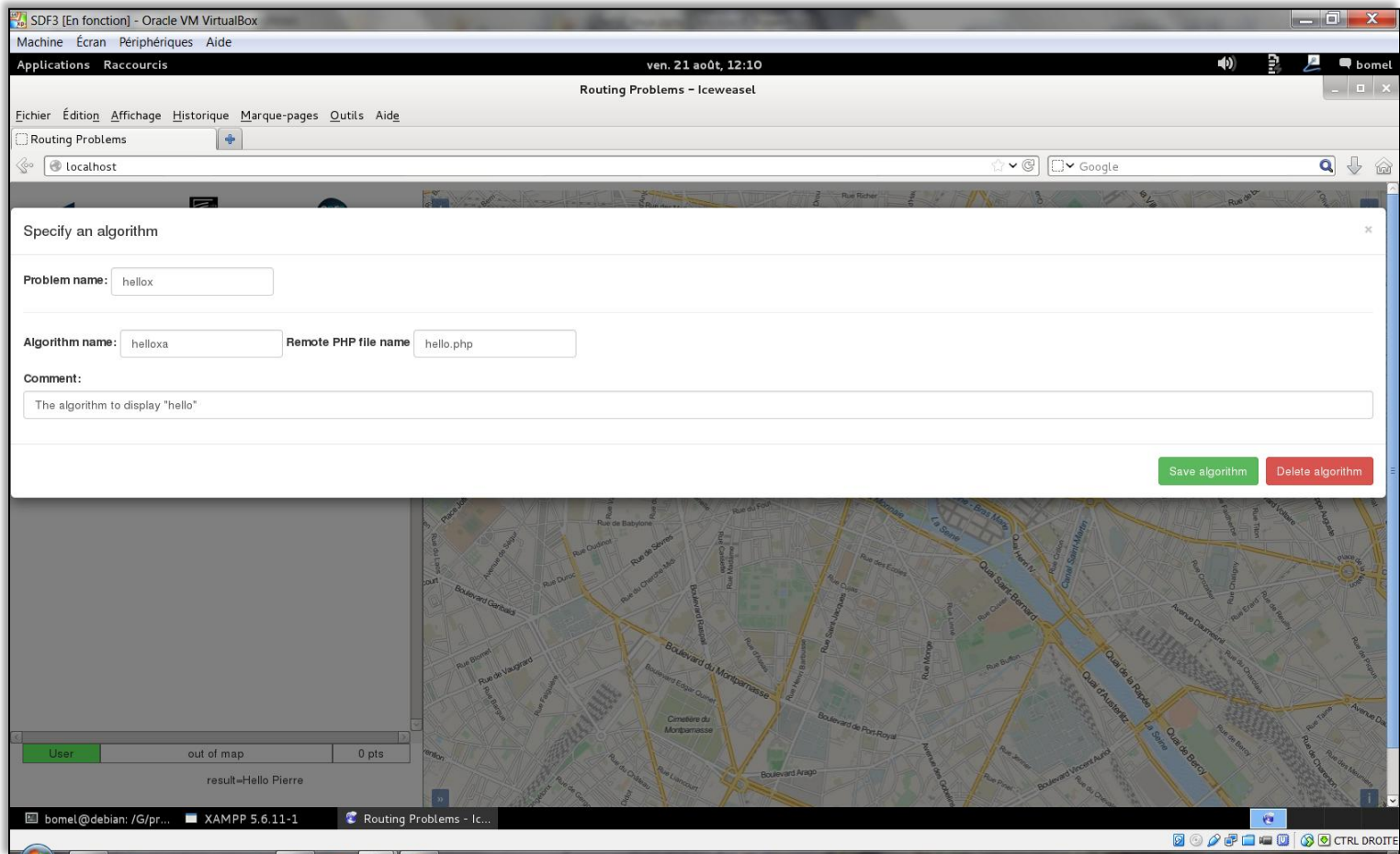


Two parameters: *name* will be an input, and *result* an output

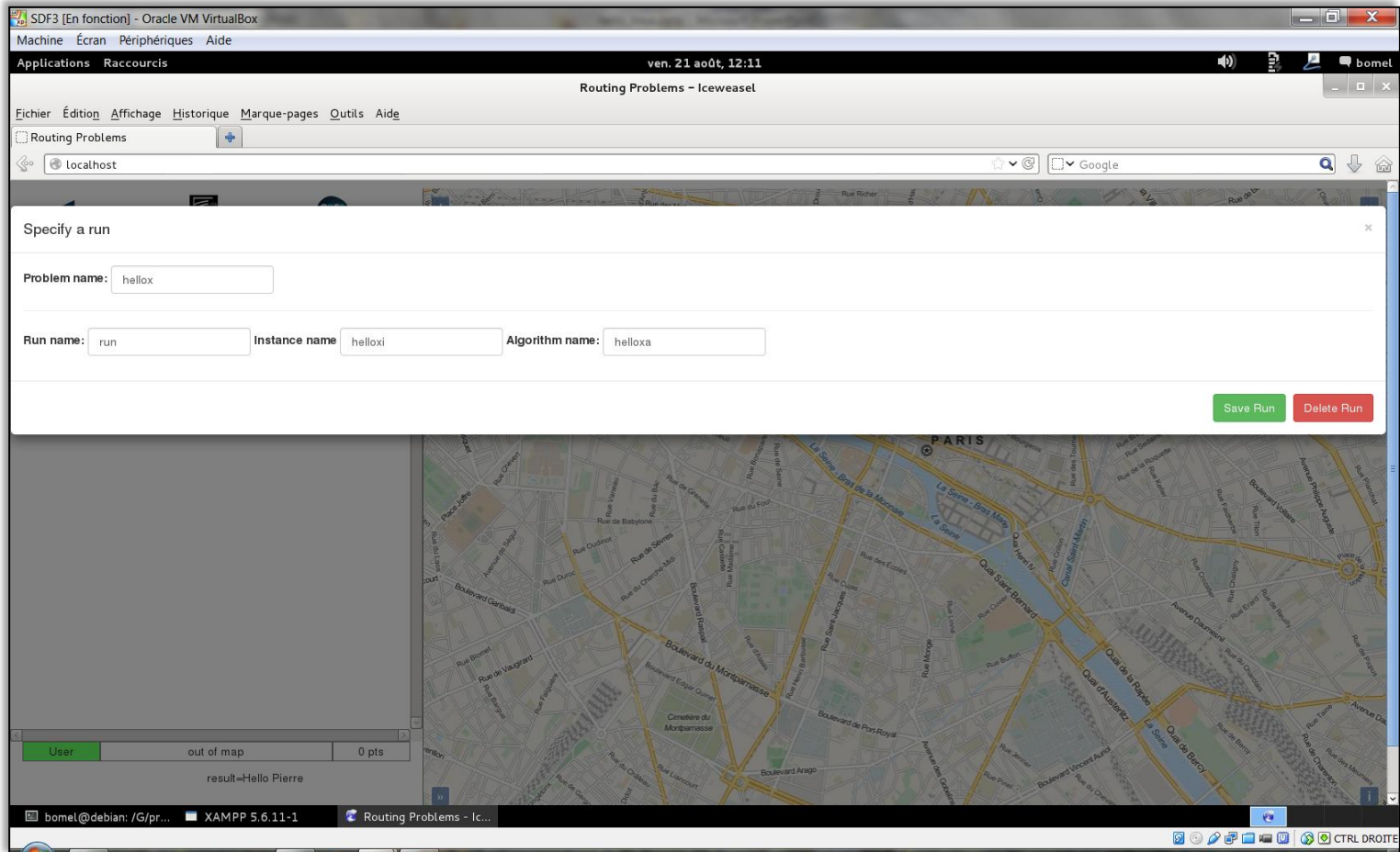
Add a new instance



Add a new algorithm

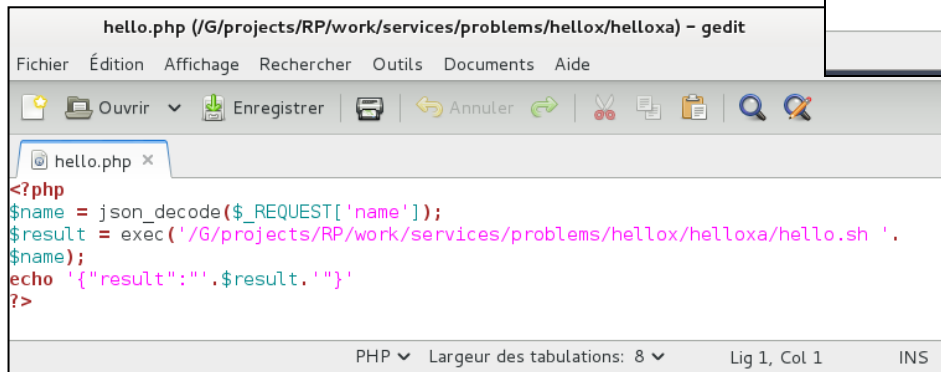
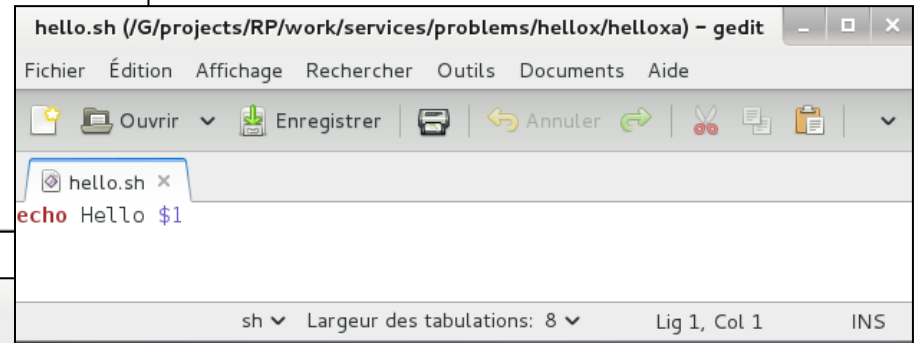
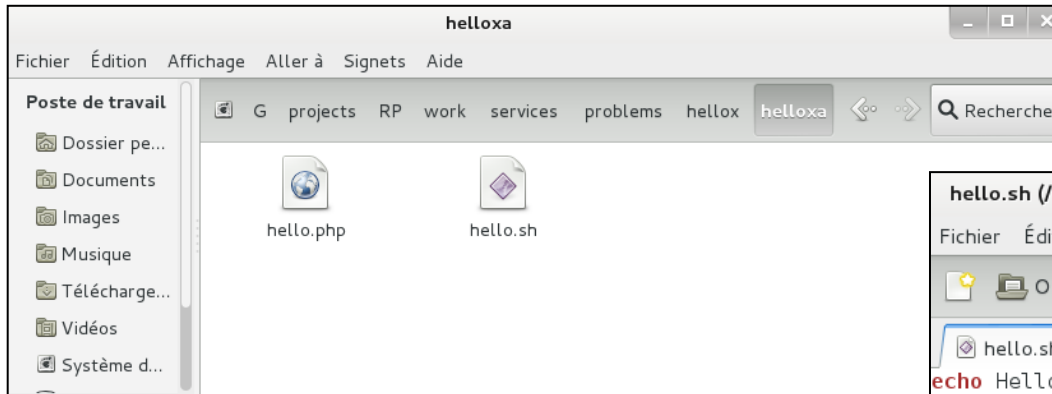


Add a new run



Create the web-service

Under the `problems` directory, add the new directories
`hellox` for the problem and `helloxa` for the algorithm
Add the PHP file `hello.php` under `hellox/helloxa`
Add the batch file `hello.sh` under `hellox/helloxa`



See the result at the bottom

The screenshot shows a web browser window titled "Routing Problems - Iceweasel" displaying the Routing Problems website. The website has a navigation menu with "Problem", "Instance", "Algorithm", and "Run" dropdowns. Below the menu is a table with columns "Lon." and "Lat.". The table contains four rows, each with a "Run" button:

Problem	Instance	Algorithm	Run
run (line)			Run
run (rev)			Run
run (hello)			Run
run (hellox)			Run

A red arrow points from the "Run" button for "run (hello)" to a text box containing "1/ Click here".

At the bottom of the page, the text "result=Hello Pierre" is displayed. A red arrow points from this text to a text box containing "2/ Look at the result!".

The background of the website is a map of Paris, France, showing the Seine river and various streets. The browser's address bar shows "localhost" and the search engine is set to "Google". The browser's status bar at the bottom shows "bomel@debian: /G/pr... XAMPP 5.6.11-1 Routing Problems - Ic...".

See the instance

Specify an instance

Problem name:

Instance name: Comment:

Values

<input type="text" value="Pierre"/>	name (word, a name to print after hello)
<input type="text" value="Hello Pierre"/>	result (string, the hello string)

Look !

Load parameters Save instance Delete instance

result=Hello Pierre

Conclusion



Congratulations !

You have created your 1rst web-service dealing with a Linux bash file.

