

# Objective = add a web-service

In this tutorial you will learn how to add a web-service that computes

$$y = ax + b$$

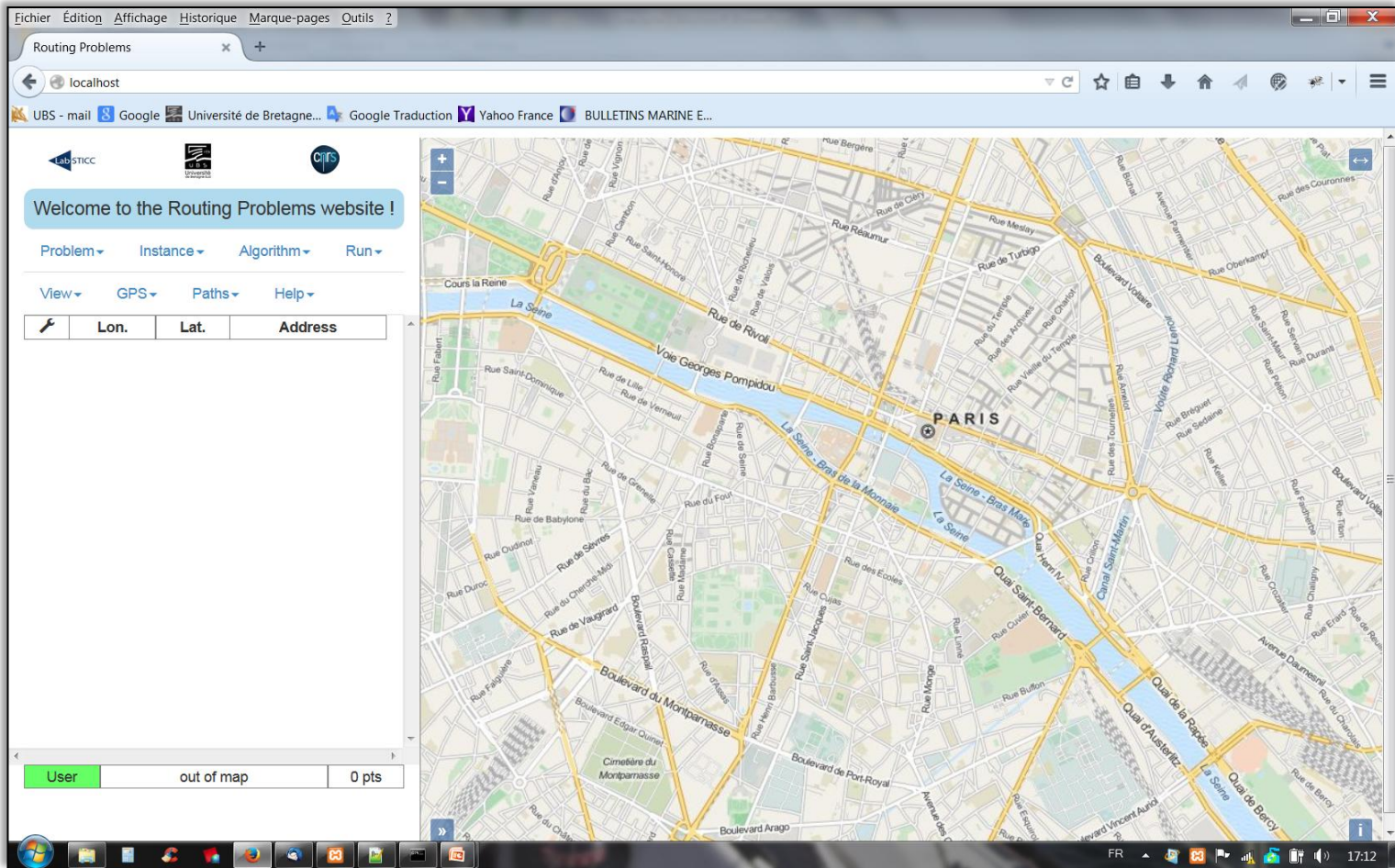


We suppose you are using Firefox, Chrome or IE.

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



# Start a browser with the URL "localhost"



You should see exactly this.



# Start a browser with the URL "localhost"

1st, we will specify:

- a problem, with parameters
- an instance, with input values
- an algorithm, with a PHP file that will be remotely executed
- and a run to automate the process

2nd, we will install the web-service under the apache server's root

3rd, we will execute the web-service

# Specifications: open the list of problems.

The screenshot shows a web browser window with the URL 'localhost'. The page title is 'Routing Problems'. The navigation menu includes 'Problem', 'Instance', 'Algorithm', and 'Run'. A red arrow points to the 'Problem' dropdown menu. Below the menu, there is a table with columns 'at.' and 'Address'. The main content area displays a map of Paris with various streets and landmarks labeled. The Windows taskbar at the bottom shows the system tray with the date 'FR' and time '17:12'.

Welcome to the Routing Problems website !

Problem Instance Algorithm Run

New Paths Help

at.	Address
-----	---------

User out of map 0 pts

Click here  
to open the list of problems.



# Specifications: create a problem.

The screenshot shows a web browser window with the URL 'localhost'. The page title is 'Routing Problems'. The browser's address bar shows 'localhost'. The page content includes a navigation menu with the following items: 'Problem', 'Instance', 'Algorithm', 'Run', 'Paths', and 'Help'. A red arrow points to the 'New' button under the 'Problem' dropdown. Below the navigation menu, there is a table with columns 'at.' and 'Address'. The main content area displays a map of Paris, France, with the Seine river and various streets labeled. The map is centered on Paris. The browser's taskbar at the bottom shows the system tray with the date 'FR' and time '17:12'.

Welcome to the Routing Problems website !

Problem ▾ Instance ▾ Algorithm ▾ Run ▾

New Paths ▾ Help ▾

at.	Address
-----	---------

User out of map 0 pts

Click here  
To create a problem.

# Specifications: fill the form for the problem and create parameters.

The screenshot shows a web browser window titled 'Routing Problems' with a URL of 'localhost'. The main content area is a form titled 'Specify a problem'. The form has three sections: 'Name:' with a text input containing 'line'; 'Comment:' with a text input containing 'compute y = ax + b'; and 'Parameters' which is currently empty. Below the form are three buttons: 'Add parameter' (green), 'Save problem' (green), and 'Delete problem' (red). Three white callout boxes with black text and red arrows point to specific elements: '1/ Give a name to the problem' points to the 'Name:' label; '2/ Add an optional comment' points to the 'Comment:' text input; and '3/ Click here to add a parameter' points to the 'Add parameter' button. The background of the browser shows a map of a city street grid.

A window named « Specify a problem » opens.

# Specifications: fill the form for the problem and create parameters.

Specify a problem

Name:

Comment:

Parameters

Native  Set  Struct

Type is:  Integer  Real  Word  String  List

1/ Give a name to the parameter

2/ You can add an optional comment

3/ Click on "Real" to specify the type of the parameter

# Specifications: fill the form for the problem and create parameters.

The screenshot shows a web browser window with the title "Routing Problems" and the address bar set to "localhost". The main content area is titled "Specify a problem" and contains the following elements:

- Name:** A text input field containing the word "line".
- Comment:** A text input field containing the mathematical expression "compute  $y = ax + b$ ".
- Parameters:** A section containing:
  - A parameter named "a" with a value of "constant" and a red "Delete parameter" button.
  - Radio buttons for "Native", "Set", and "Struct", with "Native" selected.
  - A "Type is:" section with radio buttons for "Integer", "Real", "Word", "String", and "List", with "Integer" selected.
- At the bottom right of the form, there are three buttons: "Add parameter" (green), "Save problem" (green), and "Delete problem" (red).

A red arrow points from a text box that says "Click here to add a 2<sup>nd</sup> parameter" to the "Add parameter" button.

# Specifications: fill the form for the problem and create parameters.

The screenshot shows a web browser window with the title "Routing Problems" and the URL "localhost". The main content area is a form titled "Specify a problem". The form has the following fields and controls:

- Name:** A text input field containing the text "line".
- Comment:** A text input field containing the text "compute y = ax + b".
- Parameters:** A section containing two parameter entries:
  - Parameter a:** A text input field containing "a", a dropdown menu set to "constant", and a red "Delete parameter" button. Below it are radio buttons for "Native" (selected), "Set", and "Struct". Underneath, the "Type is:" section has radio buttons for "Integer", "Real" (selected), "Word", "String", and "List".
  - Parameter b:** A text input field containing "b", a dropdown menu set to "constant", and a red "Delete parameter" button. Below it are radio buttons for "Native" (selected), "Set", and "Struct". Underneath, the "Type is:" section has radio buttons for "Integer", "Real" (selected), "Word", "String", and "List".
- Bottom Buttons:** A green "Add parameter" button, a green "Save problem" button, and a red "Delete problem" button.

A red arrow points from a text box "Click here to add a 3rd parameter" to the "Add parameter" button.

# Specifications: fill the form for the problem and create parameters.

The screenshot shows a web browser window with the following content:

- Name:**
- Comment:**
- Parameters:**
  - Parameter 1:     
 Native  Set  Struct  
Type is:  Integer  Real  Word  String  List
  - Parameter 2:     
 Native  Set  Struct  
Type is:  Integer  Real  Word  String  List
  - Parameter 3:     
 Native  Set  Struct  
Type is:  Integer  Real  Word  String  List
- Buttons:**

A text box with the text "Click here to add a 4th parameter" has a red arrow pointing to the "Add parameter" button.

# Specifications: fill the form for the problem and create parameters.

The screenshot displays a web browser window with the title "Routing Problems". The browser's address bar shows "localhost". The page content includes a form with four parameter entries:

- Parameter **a**: constant, with a "Delete parameter" button.
- Parameter **b**: constant, with a "Delete parameter" button.
- Parameter **x**: variable, with a "Delete parameter" button.
- Parameter **y**: result, with a "Delete parameter" button.

Each parameter entry includes radio buttons for "Native", "Set", and "Struct", and a "Type is:" section with radio buttons for "Integer", "Real", "Word", "String", and "List". At the bottom right of the form, there are three buttons: "Add parameter", "Save problem", and "Delete problem". A callout box with the text "Click here to save the problem" has a red arrow pointing to the "Save problem" button.



# Specifications: see all the problems.

The screenshot shows a web browser window titled "Routing Problems" with the URL "localhost". The browser's address bar and tabs are visible. The website's navigation menu includes "Problem", "Instance", "Algorithm", and "Run". A dropdown menu is open under "Problem", showing a "New" button and a list of problem names, with "line (local)" selected. A red arrow points from the "line (local)" entry to a text box. The main content area displays a map of Paris with various streets and landmarks labeled. At the bottom of the browser window, the Windows taskbar is visible, showing the system clock at 17:41.

Welcome to the Routing Problems website !

Problem Instance Algorithm Run

New

at. Address

line (local)

Click here to see all the problems.  
A new problem called "line" has been added.

# Specifications: open the list of instances.

The screenshot shows a web browser window with the URL 'localhost'. The page title is 'Routing Problems'. The navigation menu includes 'Problem', 'Instance', 'Algorithm', and 'Run'. The 'Instance' menu is open, showing a 'New' option. A red arrow points from a text box to the 'Instance' menu. The main content area displays a map of Paris with various streets and landmarks labeled. The bottom of the browser window shows the Windows taskbar with the system clock at 17:55.

Click here to open the list of instances.



# Specifications: create an instance.

The screenshot shows a web browser window with the URL 'localhost'. The page title is 'Routing Problems'. The navigation menu includes 'Problem', 'Instance', 'Algorithm', and 'Run'. A dropdown menu is open under 'Instance', showing 'View' and 'New'. A red arrow points from a text box to the 'New' option. The main content area displays a map of Paris with various streets and landmarks labeled. The bottom status bar shows 'User out of map 0 pts' and the system clock '17:59'.

Click here to create an instance.

# Specifications: fill the form for the instance and give values to the parameters.

The screenshot shows a web browser window with a map in the background. A modal window titled "Specify an instance" is open. The form contains the following fields and buttons:

- Problem name:** A text input field containing the value "line". A red arrow points to this field from callout 1.
- Instance name:** A text input field containing the value "line0". A red arrow points to this field from callout 2.
- Comment:** A text input field containing the value "values to compute  $y = ax + b$ ". A red arrow points to this field from callout 3.
- Buttons:** Three buttons are located at the bottom right of the form: "Load parameters" (green), "Save instance" (green), and "Delete instance" (red). A red arrow points to the "Load parameters" button from callout 4.

Callout 1: 1/ Specify what problem is concerned

Callout 2: 2/ Give a unique name to the instance

Callout 3: 3/ You can add an optional comment

Callout 4: 4/ Click here to read the parameters

A window named « Specify an instance » opens.



# Specifications: fill the form for the instance and give values to the parameters.

Specify an instance

Problem name:

Instance name:  Comment:

Values

<input type="text" value="1"/>	a (real, constant)
<input type="text" value="1"/>	b (real, constant)
<input type="text" value="10"/>	x (real, variable)
<input type="text" value="0"/>	y (real, result)

1/ Give values to all input parameters

3/ Click here to save the instance

2/ Also give a default value to y, it will be recomputed by the web-service

Load parameters Save instance Delete instance

# Specifications: see all the instances.

The screenshot shows a web browser window with the URL 'localhost'. The page title is 'Routing Problems'. The navigation menu includes 'Problem', 'Instance', 'Algorithm', and 'Run'. The 'Instance' dropdown menu is open, showing 'New' and 'line0 (line)'. A red arrow points to the 'line0 (line)' option. The main content area displays a map of Paris with various streets and landmarks labeled. The bottom status bar shows 'User out of map 0 pts' and the system clock '18:16'.

Click here to see all the instances.  
A new instance called "line0" has been added.



# Specifications: open the list of algorithms.

The screenshot shows a web browser window with the title "Routing Problems". The address bar shows "localhost". The browser's address bar contains several bookmarks: "UBS - mail", "Google", "Université de Bretagne...", "Google Traduction", "Yahoo France", and "BULLETINS MARINE E...". The website's header includes logos for "LAB-STICC", "UNIV BREST", and "CIRIS". A blue banner reads "Welcome to the Routing Problems website!". Below the banner is a navigation menu with "Problem", "Instance", "Algorithm", and "Run" buttons. A red arrow points from a text box to the "Algorithm" button. Below the menu are "View", "GPS", "Point", and "New" buttons, followed by input fields for "Lon." and "Lat.". The main content area is a map of Paris, showing the Seine river and various streets. At the bottom left, there is a status bar with "User", "out of map", and "0 pts". The Windows taskbar is visible at the bottom of the screen.

Click here to open the list of algorithms.

# Specifications: create an algorithm.

The screenshot shows a web browser window titled "Routing Problems" with the URL "localhost". The browser's address bar and tabs are visible. The website's navigation menu includes "Problem", "Instance", "Algorithm", and "Run". A sub-menu is open under "Algorithm", showing options for "View", "GPS", "Path", and "New". A red arrow points from a text box to the "New" option. The main content area displays a map of Paris with various streets and landmarks labeled. The Windows taskbar is visible at the bottom of the screen.

Welcome to the Routing Problems website !

Problem Instance Algorithm Run

View GPS Path New

Lon. Lat.

User out of map 0 pts

Click here  
to create an algorithm.



# Specifications: create an algorithm.

The screenshot shows a web browser window with a 'Specify an algorithm' dialog box open. The dialog box contains the following fields and buttons:

- Problem name:** A text input field containing the text 'line'. A red arrow points from this field to callout 1.
- Algorithm name:** A text input field containing the text 'line\_algo'. A red arrow points from this field to callout 2.
- Remote PHP file name:** A text input field containing the text 'exec.php'. A red arrow points from this field to callout 3.
- Comment:** A text area containing the text 'the exec.php php file computes  $y = ax + b$  with the values of a, b and x taken from an instance'. A red arrow points from this area to callout 4.
- Buttons:** Two buttons are located at the bottom right: a green 'Save algorithm' button and a red 'Delete algorithm' button. A red arrow points from the 'Save algorithm' button to callout 5.

The background of the browser shows a map of a city street grid. The browser's address bar shows 'localhost' and the title bar shows 'Routing Problems'.

1/ Specify what problem is concerned

4/ You can add an optional comment

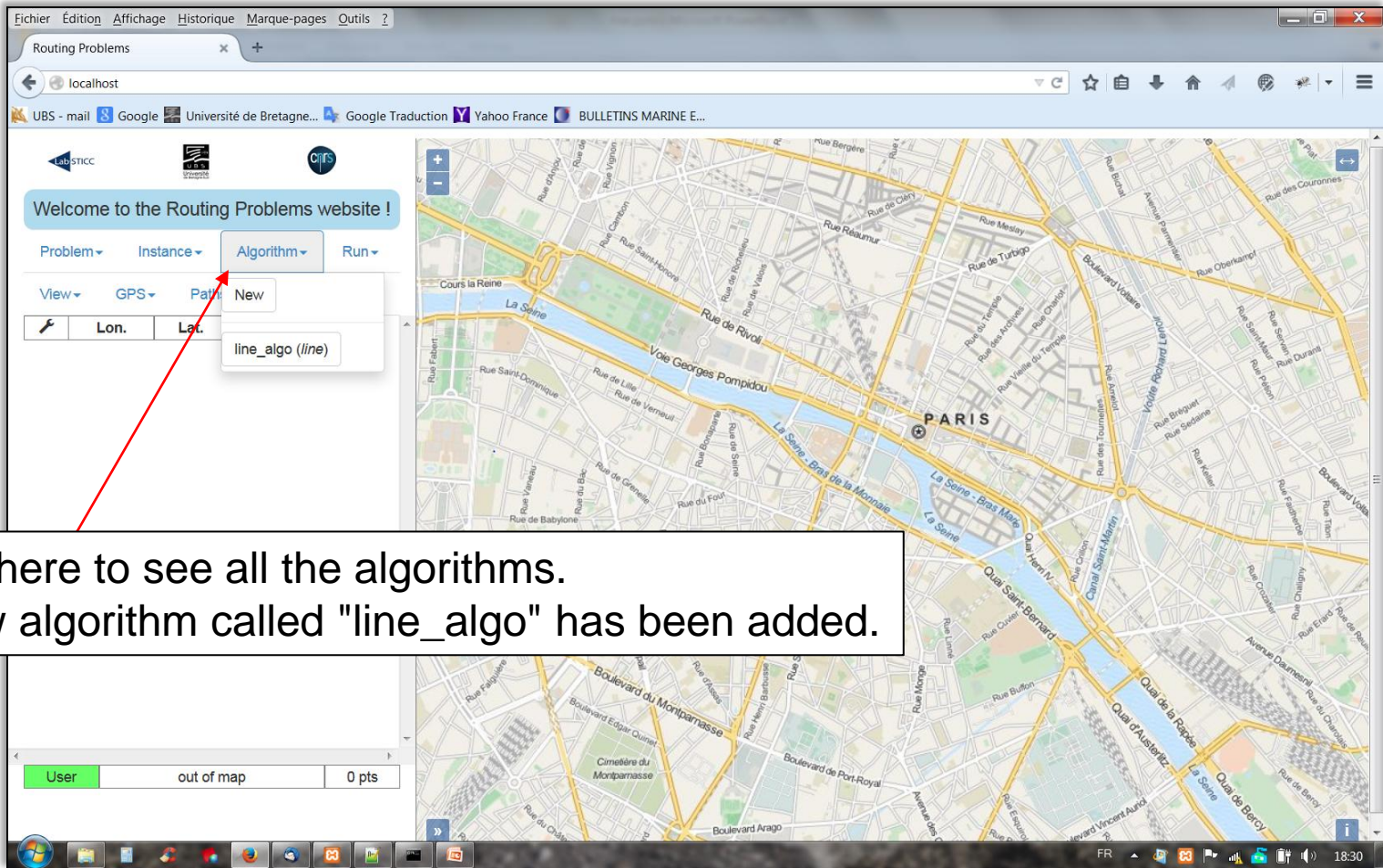
2/ Give a unique name to the algorithm

5/ Click here to save the algorithm

3/ Give the name of the remote php file that will interface the web-service with your program

A window named « Specify an algorithm » opens.

# Specifications: see all the algorithms.



The screenshot shows a web browser window titled "Routing Problems" at the URL "localhost". The browser's address bar and tabs are visible. The website's navigation bar includes "Problem", "Instance", "Algorithm", and "Run" menus. A red arrow points to the "Algorithm" dropdown menu, which is open and shows "New" and "line\_algo (line)" options. Below the navigation bar, there are input fields for "Lon." and "Lat." and a "View" button. The main content area displays a map of Paris with various streets and landmarks labeled, such as "La Seine", "Voie Georges Pompidou", and "PARIS". At the bottom of the interface, there is a status bar with a "User" field containing "out of map" and "0 pts".

Click here to see all the algorithms.  
A new algorithm called "line\_algo" has been added.

# Specifications: open the list of runs.

The screenshot shows a web browser window with the title "Routing Problems". The address bar shows "localhost". The browser's address bar contains several tabs: "UBS - mail", "Google", "Université de Bretagne...", "Google Traduction", "Yahoo France", and "BULLETINS MARINE E...". The website's header includes logos for "STICC", "STIC", and "Cijis". A blue banner reads "Welcome to the Routing Problems website!". Below the banner is a navigation menu with the following items: "Problem", "Instance", "Algorithm", "Run", "View", "GPS", "Paths", "Help", and "New". A red arrow points from a text box to the "Run" menu item. Below the menu is a table with columns: "Lon.", "Lat.", and "Address". The main content area is a map of Paris, showing the Seine river and various streets. The map is centered on the city of Paris. At the bottom of the browser window, the Windows taskbar is visible, showing the system tray with the time "18:33" and the language "FR".

Lon.	Lat.	Address
		User
		out of map
		0 pts

Click here  
to open the list of runs.



# Specifications: create a run.

The screenshot shows a web browser window with the title "Routing Problems". The address bar shows "localhost". The browser's address bar contains several tabs and a search bar. The main content area features a map of Paris with various streets and landmarks labeled. On the left side, there is a navigation menu with the following items: "Problem", "Instance", "Algorithm", "Run", "View", "GPS", "Paths", "Help", and "New". A red arrow points from a text box to the "New" button. Below the navigation menu, there is a table with columns for "Lon.", "Lat.", and "Address". At the bottom of the page, there is a status bar with the text "User", "out of map", and "0 pts".

Welcome to the Routing Problems website !

Problem Instance Algorithm Run

View GPS Paths Help **New**

Lon.	Lat.	Address

User out of map 0 pts

Click here to create a run.

# Specifications: create a run.

The screenshot shows a web browser window with a map in the background. A modal window titled "Specify a run" is open. It contains the following fields and buttons:

- Problem name:** A text input field containing the value "line".
- Run name:** A text input field containing the value "run\_line".
- Instance name:** A text input field containing the value "line0".
- Algorithm name:** A text input field containing the value "line\_algo".
- Buttons:** A green "Save Run" button and a red "Delete Run" button.

Five numbered callouts with red arrows point to these elements:

- 1/ Specify what problem is concerned (points to the "Problem name" field)
- 2/ Give a unique name to the run (points to the "Run name" field)
- 3/ Give the name of the instance from which the values of the parameters will be taken (points to the "Instance name" field)
- 4/ Give the name of the algorithm (points to the "Algorithm name" field)
- 5/ Click here to save the run (points to the "Save Run" button)

A window named « Specify a run » opens.

# Specifications: see all the runs.

The screenshot shows a web browser window titled "Routing Problems" at the URL "localhost". The page features a navigation menu with "Problem", "Instance", "Algorithm", and "Run" dropdowns. A "Run" dropdown menu is open, showing a list of runs: "New", "run\_line (line)", and a green "Run" button. A red arrow points from the "Run" button in the dropdown to a text box. Another red arrow points from the "Run" button in the main navigation menu to the same text box. The main content area displays a map of Paris with a highlighted route. At the bottom, a status bar shows "User" and "out of map" with "0 pts".

Welcome to the Routing Problems website !

Problem ▾ Instance ▾ Algorithm ▾ Run ▾

View ▾ GPS ▾ Paths ▾ Help ▾

Lon. Lat. Address

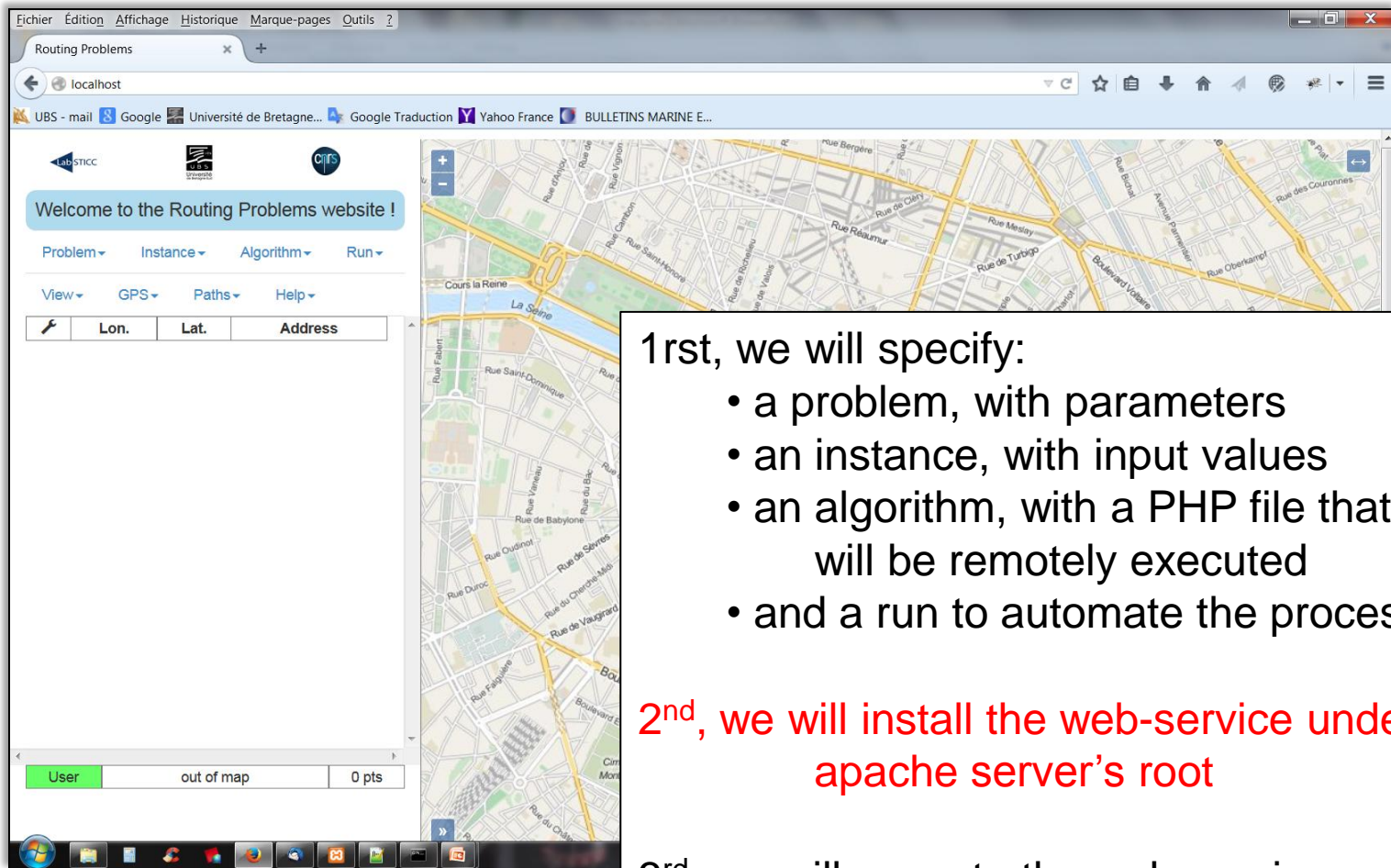
- New
- run\_line (line)
- Run

User out of map 0 pts

If you click on this green button it will execute the web-service for you. But we need to create this web-service first.

Click here to see all the runs.  
A new run called "run\_line" has been added.

# Installation of the web-service



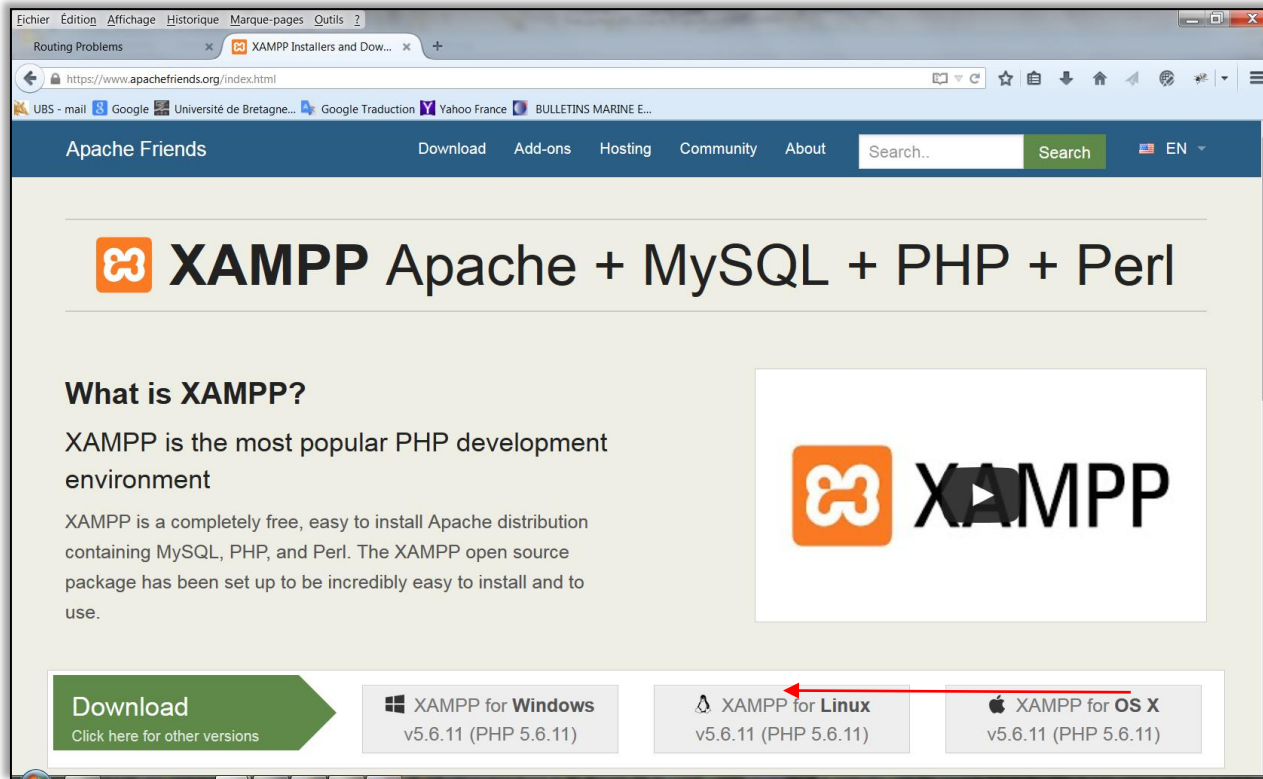
1<sup>st</sup>, we will specify:

- a problem, with parameters
- an instance, with input values
- an algorithm, with a PHP file that will be remotely executed
- and a run to automate the process

2<sup>nd</sup>, we will install the web-service under the apache server's root

3<sup>rd</sup>, we will execute the web-service

# Installation of the web-service

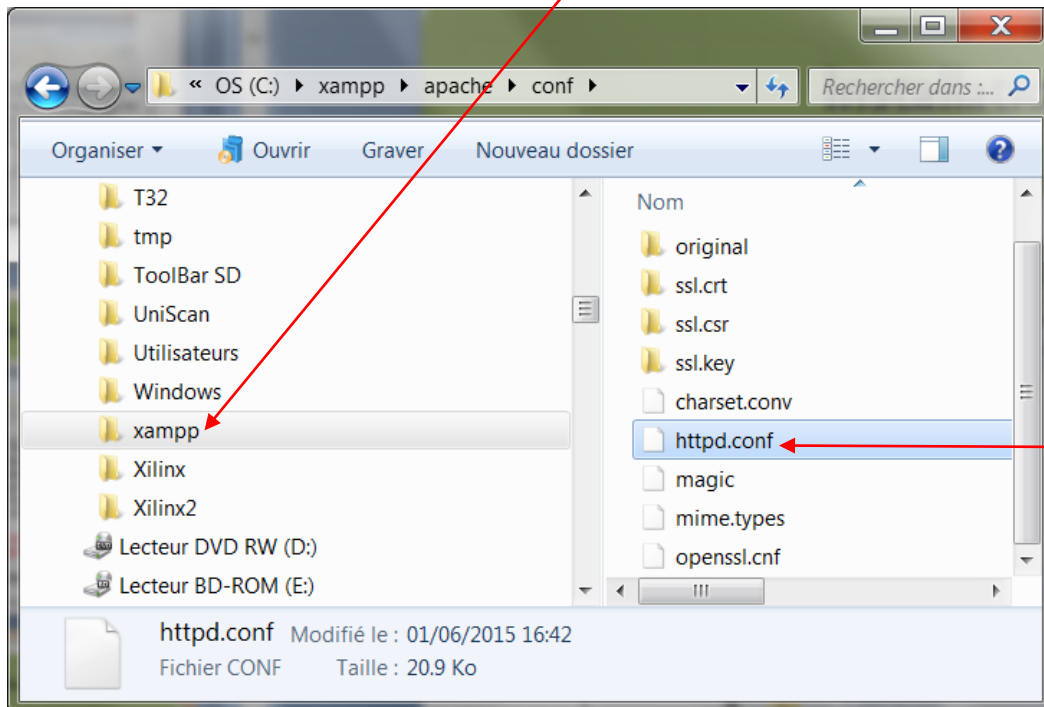


The screenshot shows a web browser window displaying the Apache Friends website. The page title is "XAMPP Apache + MySQL + PHP + Perl". Below the title, there is a section titled "What is XAMPP?" with the following text: "XAMPP is the most popular PHP development environment. XAMPP is a completely free, easy to install Apache distribution containing MySQL, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use." To the right of this text is a video player showing the XAMPP logo. At the bottom of the page, there are three download buttons: "Download" (with a sub-link "Click here for other versions"), "XAMPP for Windows v5.6.11 (PHP 5.6.11)", "XAMPP for Linux v5.6.11 (PHP 5.6.11)", and "XAMPP for OS X v5.6.11 (PHP 5.6.11)". A red arrow points to the Linux download button.

If you need to install XAMPP  
go to <https://www.apachefriends.org/index.html>  
and follow the instructions.

# Installation of the web-service

The Apache server we have selected is XAMPP. It should be installed on the C: drive Under the *xampp* directory.

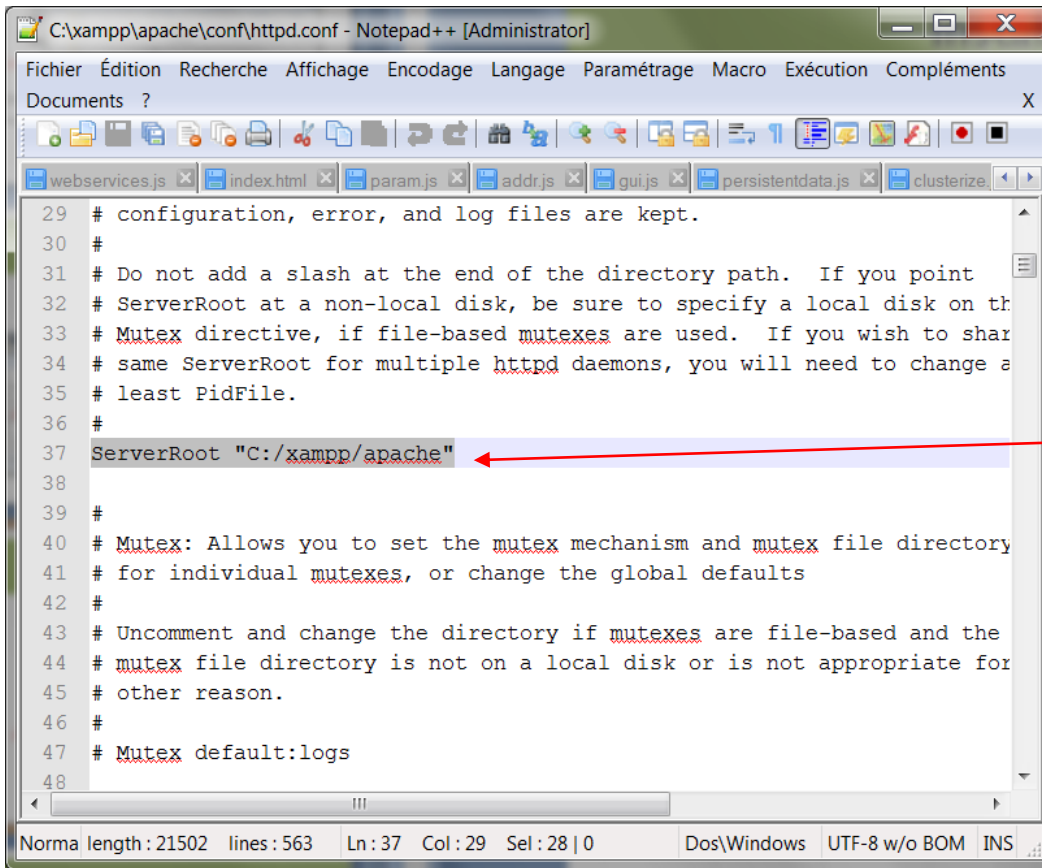


The configuration file is *httpd.conf*

You will modify it to locate your web-services.

Open it with an editor.

# Installation of the web-service



```
C:\xampp\apache\conf\httpd.conf - Notepad++ [Administrator]
Fichier Édition Recherche Affichage Encodage Langage Paramétrage Macro Exécution Compléments
Documents ?
webservices.js | index.html | param.js | addr.js | gui.js | persistentdata.js | clusterize
29 # configuration, error, and log files are kept.
30 #
31 # Do not add a slash at the end of the directory path. If you point
32 # ServerRoot at a non-local disk, be sure to specify a local disk on th
33 # Mutex directive, if file-based mutexes are used. If you wish to shar
34 # same ServerRoot for multiple httpd daemons, you will need to change a
35 # least PidFile.
36 #
37 ServerRoot "C:/xampp/apache"
38 #
39 #
40 # Mutex: Allows you to set the mutex mechanism and mutex file directory
41 # for individual mutexes, or change the global defaults
42 #
43 # Uncomment and change the directory if mutexes are file-based and the
44 # mutex file directory is not on a local disk or is not appropriate for
45 # other reason.
46 #
47 # Mutex default:logs
48
```

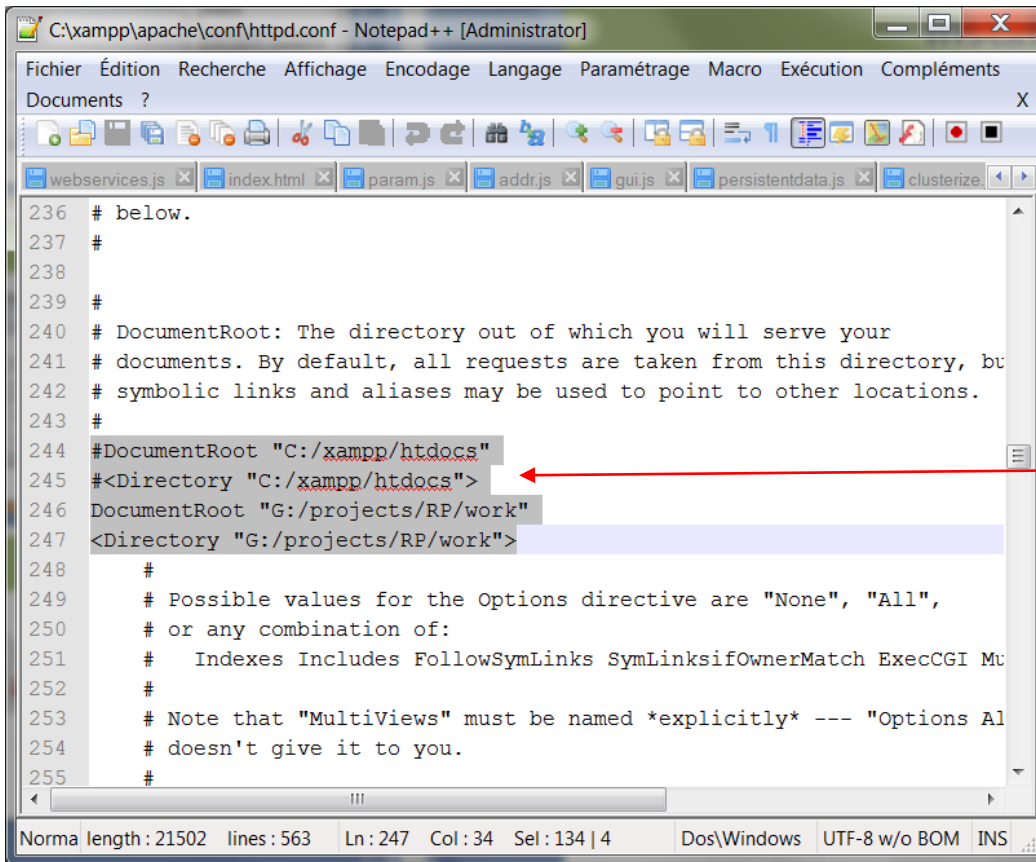
Look at the *ServerRoot* directive.

It should point to the location of the XAMPP installation of the apache server.

You should not modify it once it has been specified, unless you want to reinstall XAMPP somewhere else on your disk.

XAMPP has been installed somewhere, *ServerRoot* contains this location.

# Installation of the web-service



```
C:\xampp\apache\conf\httpd.conf - Notepad++ [Administrator]
Fichier Édition Recherche Affichage Encodage Langage Paramétrage Macro Exécution Compléments
Documents ?
webservices.js | index.html | param.js | addr.js | gui.js | persistentdata.js | clusterize
236 # below.
237 #
238 #
239 #
240 # DocumentRoot: The directory out of which you will serve your
241 # documents. By default, all requests are taken from this directory, bu
242 # symbolic links and aliases may be used to point to other locations.
243 #
244 #DocumentRoot "C:/xampp/htdocs"
245 #<Directory "C:/xampp/htdocs"> ←
246 DocumentRoot "G:/projects/RP/work"
247 <Directory "G:/projects/RP/work">
248 #
249 # Possible values for the Options directive are "None", "All",
250 # or any combination of:
251 #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI Mu
252 #
253 # Note that "MultiViews" must be named *explicitly* --- "Options Al
254 # doesn't give it to you.
255 #
Norma length : 21502 lines : 563 Ln : 247 Col : 34 Sel : 134 | 4 Dos\Windows UTF-8 w/o BOM INS
```

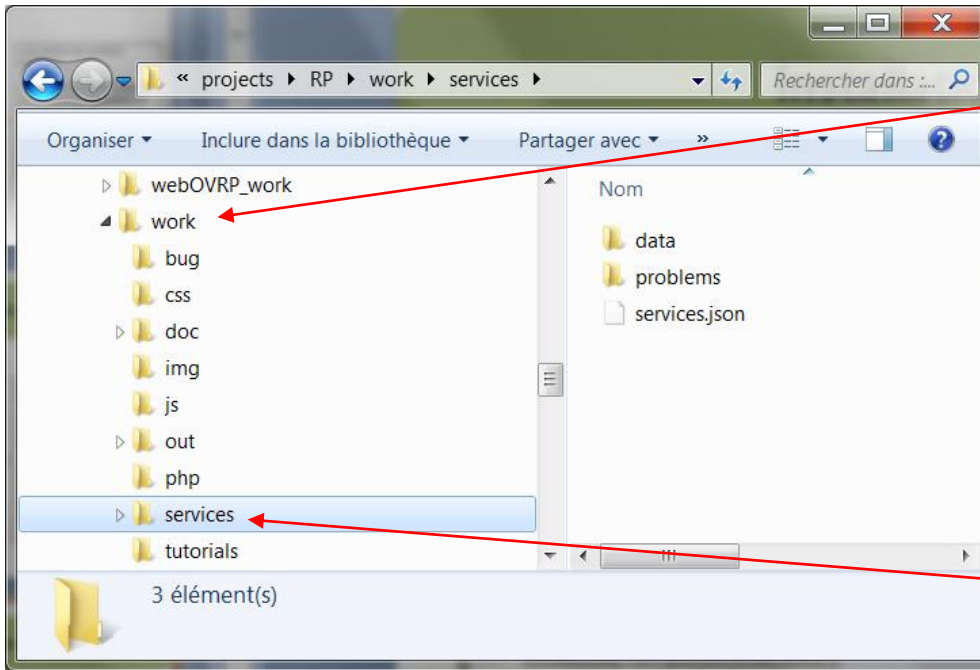
Look at the *DocumentRoot* directive.

It should point to the location of the private directory of the apache server.

You should not modify it once it has been specified, unless you want to move this directory somewhere else on your disk.

Apache works with private data, *DocumentRoot* contains this location.

# Installation of the web-service



Open a file explorer and go to the location of the web-services.

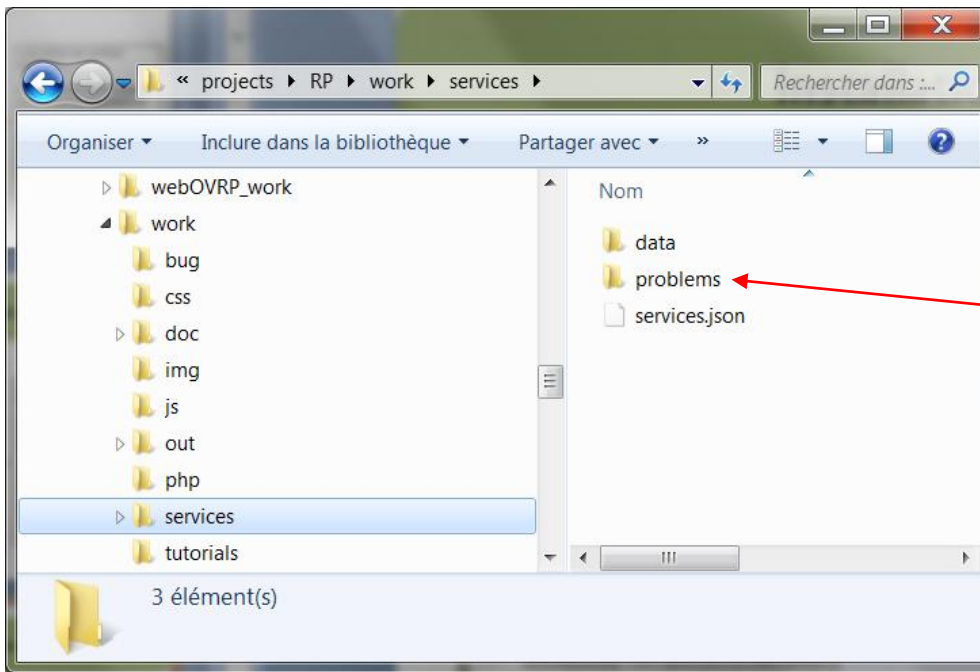
This location is the value of the *DocumentRoot* directive.

You may see several directories.

Among them, the *services* directory is the one where are stored all the web-services.

If it does not exist, create it.

# Installation of the web-service

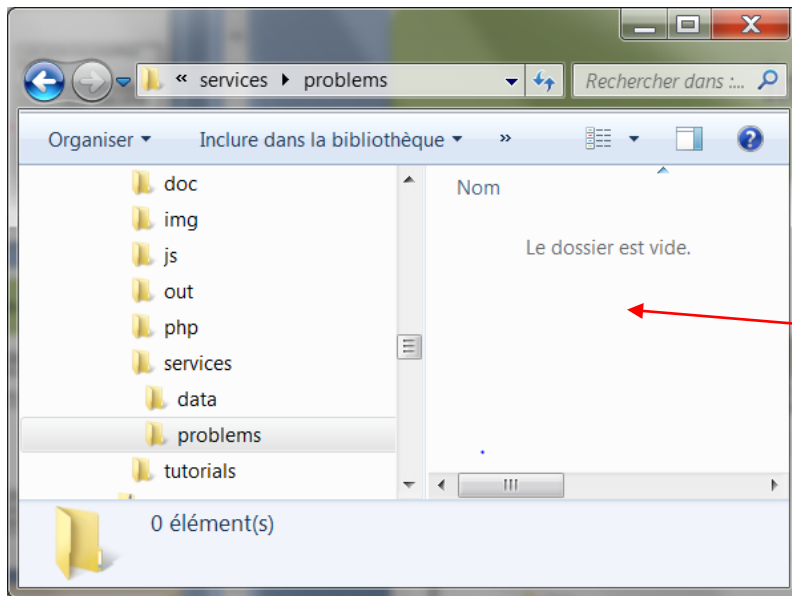


Under *services* there is another directory named *problems*.

This is where all the web-services (called problems in OR) are stored.

Open it.

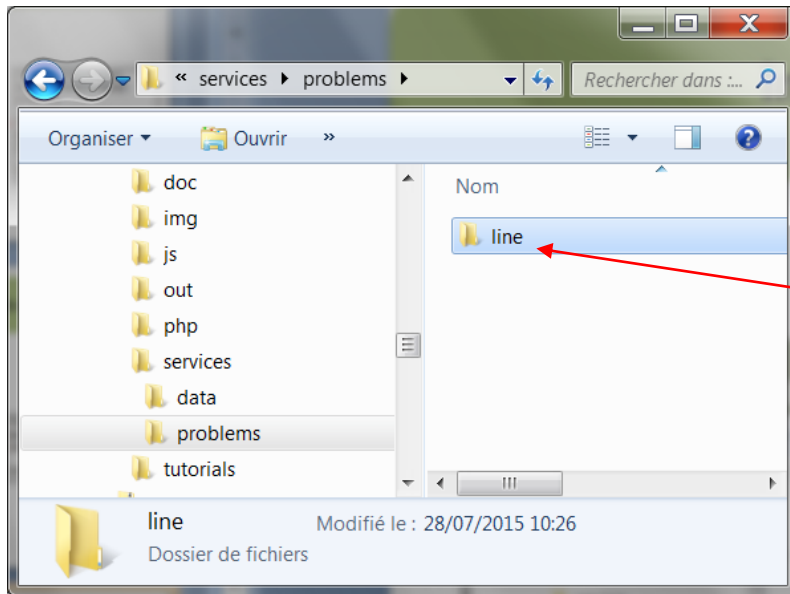
# Installation of the web-service



This directory is empty because you have currently no web-service.

We'll add the *line* web-service.

# Installation of the web-service



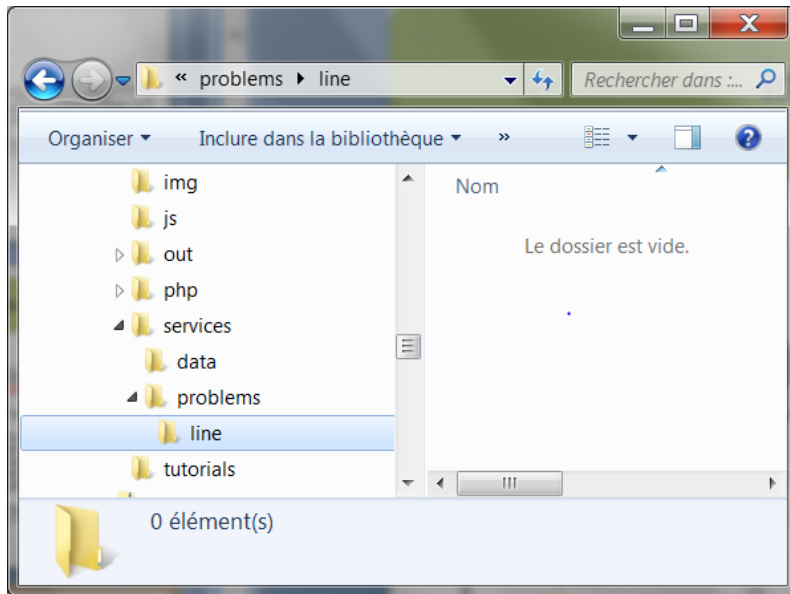
Rule1 : each problem is identified by a unique name.

In our case it is *line*.

To represent this problem on the web-server's side, you must create a directory with the same name than the problem.

Create the *line* directory here.

# Installation of the web-service



Of course this new directory is empty.

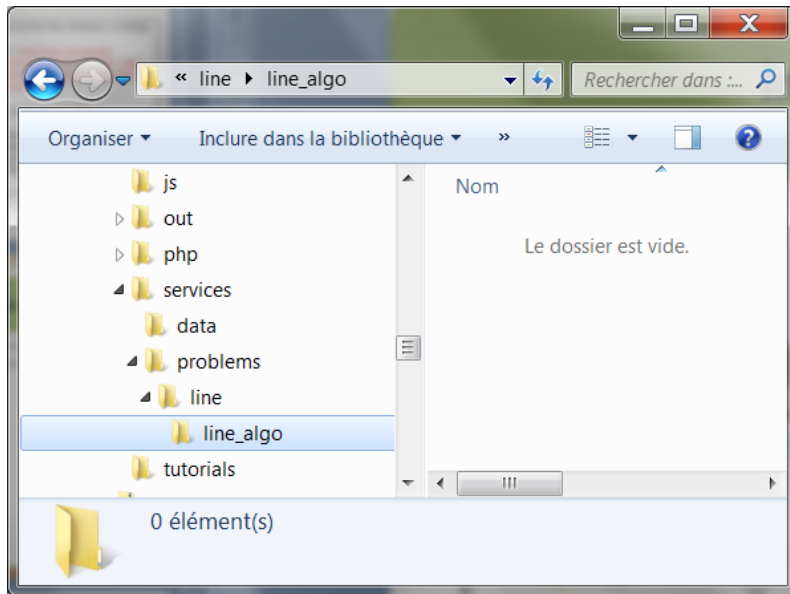
Rule 2: there may be several algorithms to solve a problem. Each algorithm has a unique name, for a given problem.

In our case it is *line\_algo*.

To represent this algorithm on the web-server's side, you must create a directory with the same name than the algorithm.

Create the *line\_algo* directory here.

# Installation of the web-service



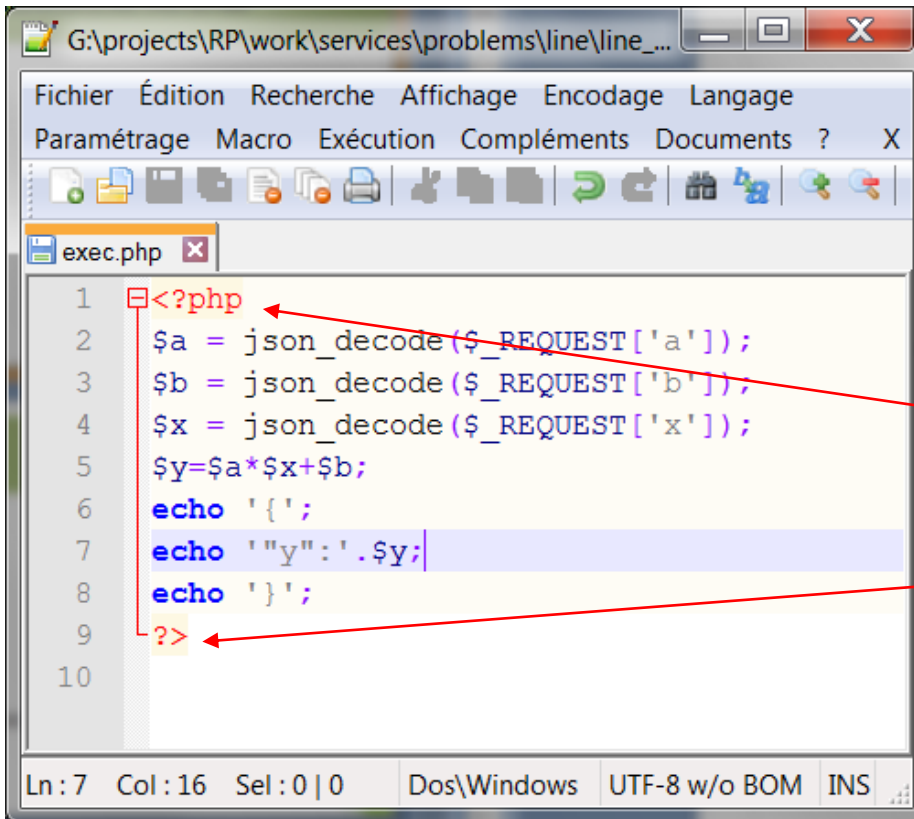
Of course this new directory is empty.

Do you remember the name we have chosen for the PHP file while creating the algorithm ?

It was *exec.php*

So, use an editor and create a php file with name *exec.php* here.

# Installation of the web-service



```
1 <?php
2 $a = json_decode($_REQUEST['a']);
3 $b = json_decode($_REQUEST['b']);
4 $x = json_decode($_REQUEST['x']);
5 $y=$a*$x+$b;
6 echo '{';
7 echo '"y":'. $y;
8 echo '}' ;
9 ?>
```

Learning PHP is out of the scope of this tutorial, but we think that it is simple enough to be understood very quickly.

A php file always has

a header

and a footer.

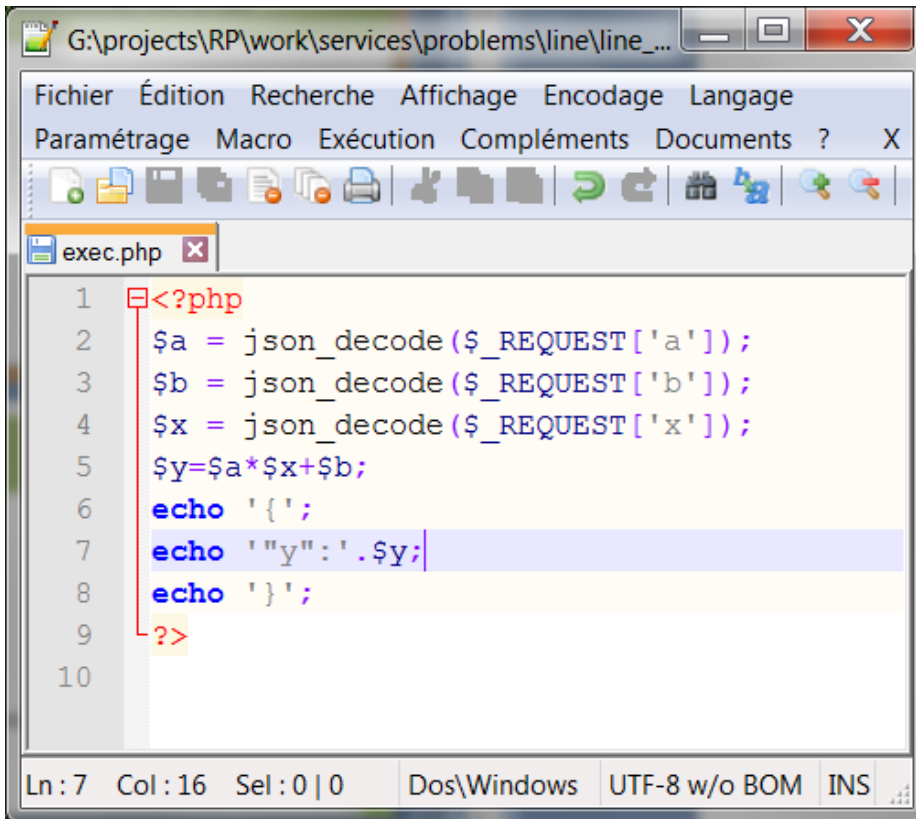
# Installation of the web-service

```
1 <?php
2 $a = json_decode($_REQUEST['a']);
3 $b = json_decode($_REQUEST['b']);
4 $x = json_decode($_REQUEST['x']);
5 $y=$a*$x+$b;
6 echo '{';
7 echo '"y":'. $y;
8 echo '}' ;
9 ?>
```

A web-service does 3 things:

- 1/ read values of parameters
- 2/ do some processing
- 3/ return values of results

# Installation of the web-service



```
1 <?php
2 $a = json_decode($_REQUEST['a']);
3 $b = json_decode($_REQUEST['b']);
4 $x = json_decode($_REQUEST['x']);
5 $y=$a*$x+$b;
6 echo '{';
7 echo '"y":'.$y;
8 echo '}' ;
9 ?>
```

The GUI of the RP web-site has been developed with Javascript. All the transferred values are JSON coded and identified by their corresponding parameter's name.

To get the value of the **a** parameter, one has to write:

```
$_REQUEST['a']
```

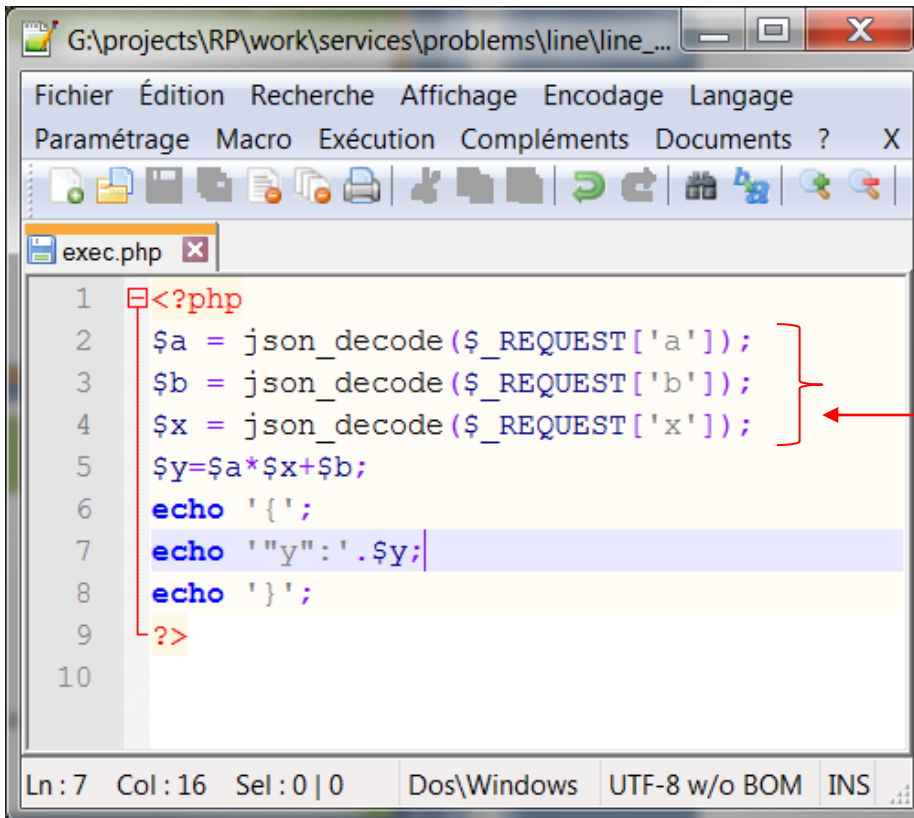
to decode it:

```
json_decode($_REQUEST['a'])
```

to copy the value into a variable:

```
$a = ...
```

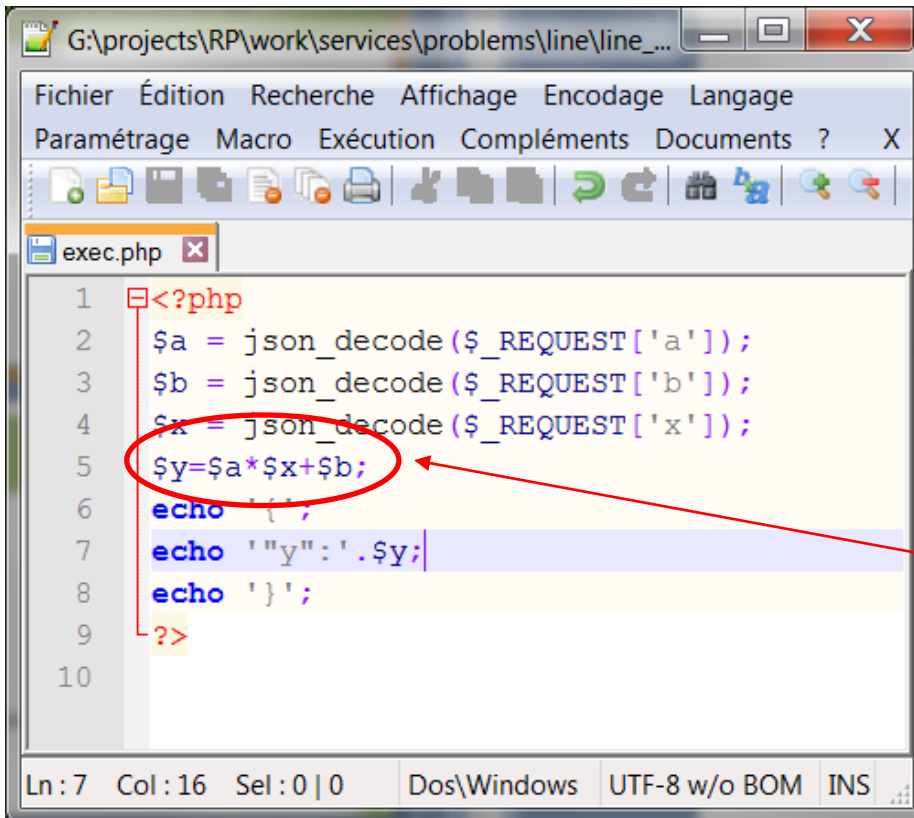
# Installation of the web-service



```
1 <?php
2 $a = json_decode($_REQUEST['a']);
3 $b = json_decode($_REQUEST['b']);
4 $x = json_decode($_REQUEST['x']);
5 $y=$a*$x+$b;
6 echo '{';
7 echo '"y":'. $y;
8 echo '}' ;
9 ?>
```

As you can see, the start of the php file reads the values of the *a*, *b*, and *x* parameters and copy them into the variables *\$a*, *\$b* and *\$x*.

# Installation of the web-service

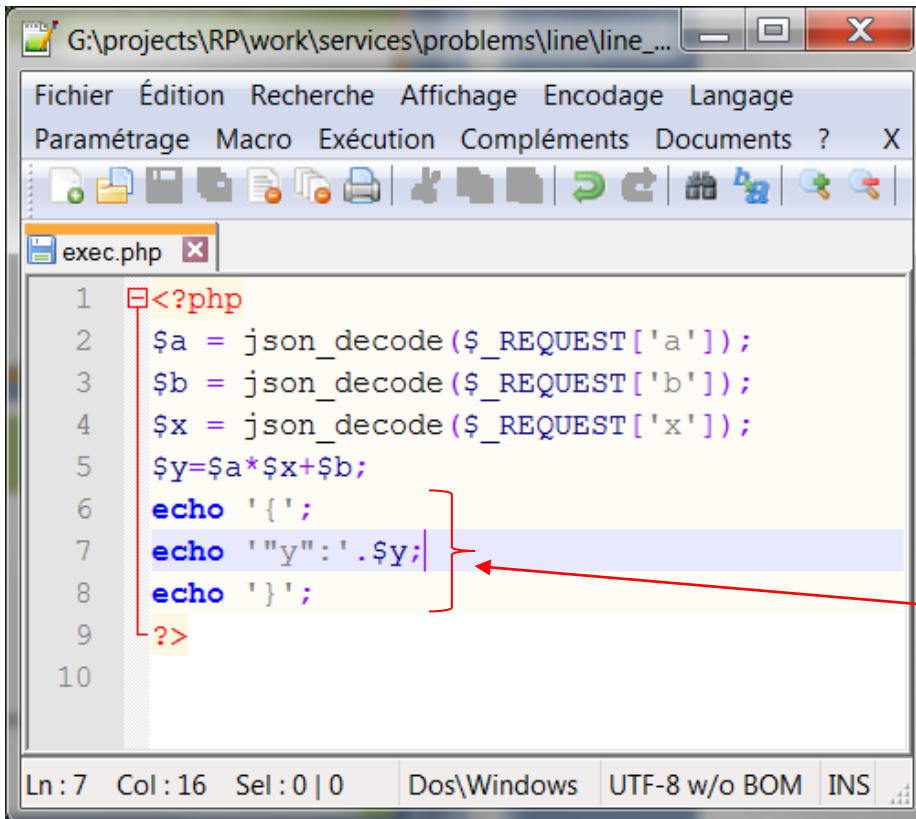


```
1 <?php
2 $a = json_decode($_REQUEST['a']);
3 $b = json_decode($_REQUEST['b']);
4 $x = json_decode($_REQUEST['x']);
5 $y=$a*$x+$b;
6 echo '{';
7 echo '"y":'. $y;
8 echo '}' ;
9 ?>
```

The processing part is here the simplest.

We compute into the  $y$  variable the value  $a * x + b$ , which is exactly what we wanted to do.

# Installation of the web-service



```
1 <?php
2 $a = json_decode($_REQUEST['a']);
3 $b = json_decode($_REQUEST['b']);
4 $x = json_decode($_REQUEST['x']);
5 $y=$a*$x+$b;
6 echo '{';
7 echo '"y":'.$y;
8 echo '}';
```

The result part may not seem obvious, as long as you do not know what JSON is.

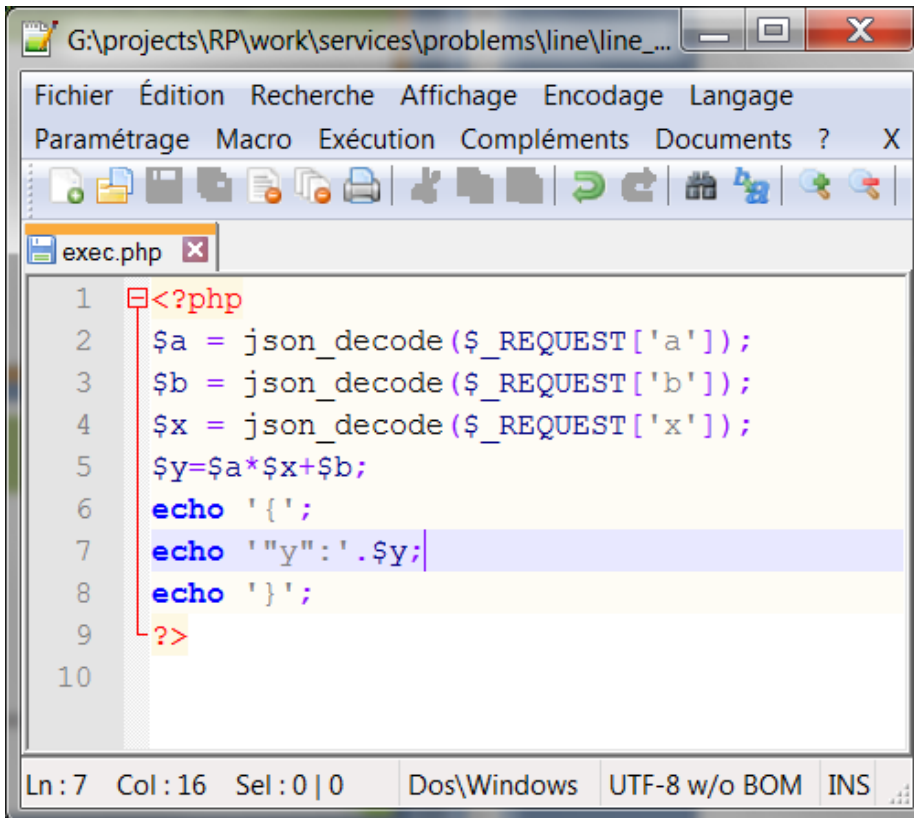
But, believe us, it is very easy to understand.

Here, we send back a string that is JSON coded. This string is exactly:

```
{"y":0}
```

Of course  $y = 0$ , because we assume that  $a$ ,  $b$  and  $x$  were all equal to 0 in that case.

# Installation of the web-service

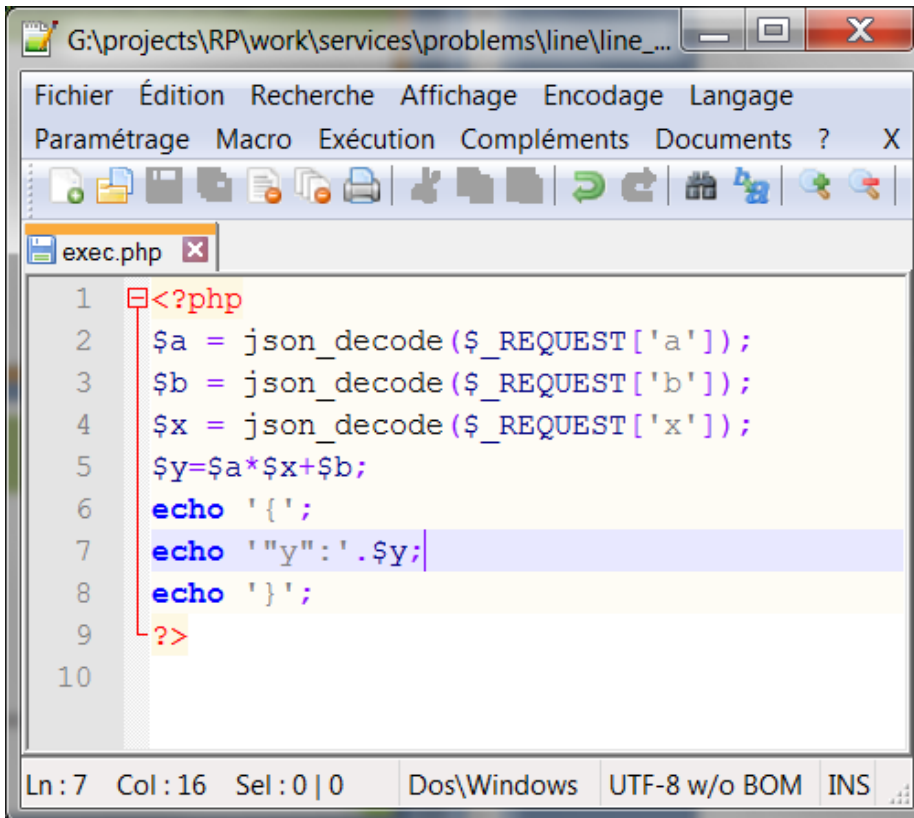


```
1 <?php
2 $a = json_decode($_REQUEST['a']);
3 $b = json_decode($_REQUEST['b']);
4 $x = json_decode($_REQUEST['x']);
5 $y=$a*$x+$b;
6 echo '{';
7 echo '"y":'.$y;
8 echo '}';
```

Rq: if we had several values to send back, we would have sent a string like this one:

```
{"y1":0, "y2":1}
```

# Installation of the web-service

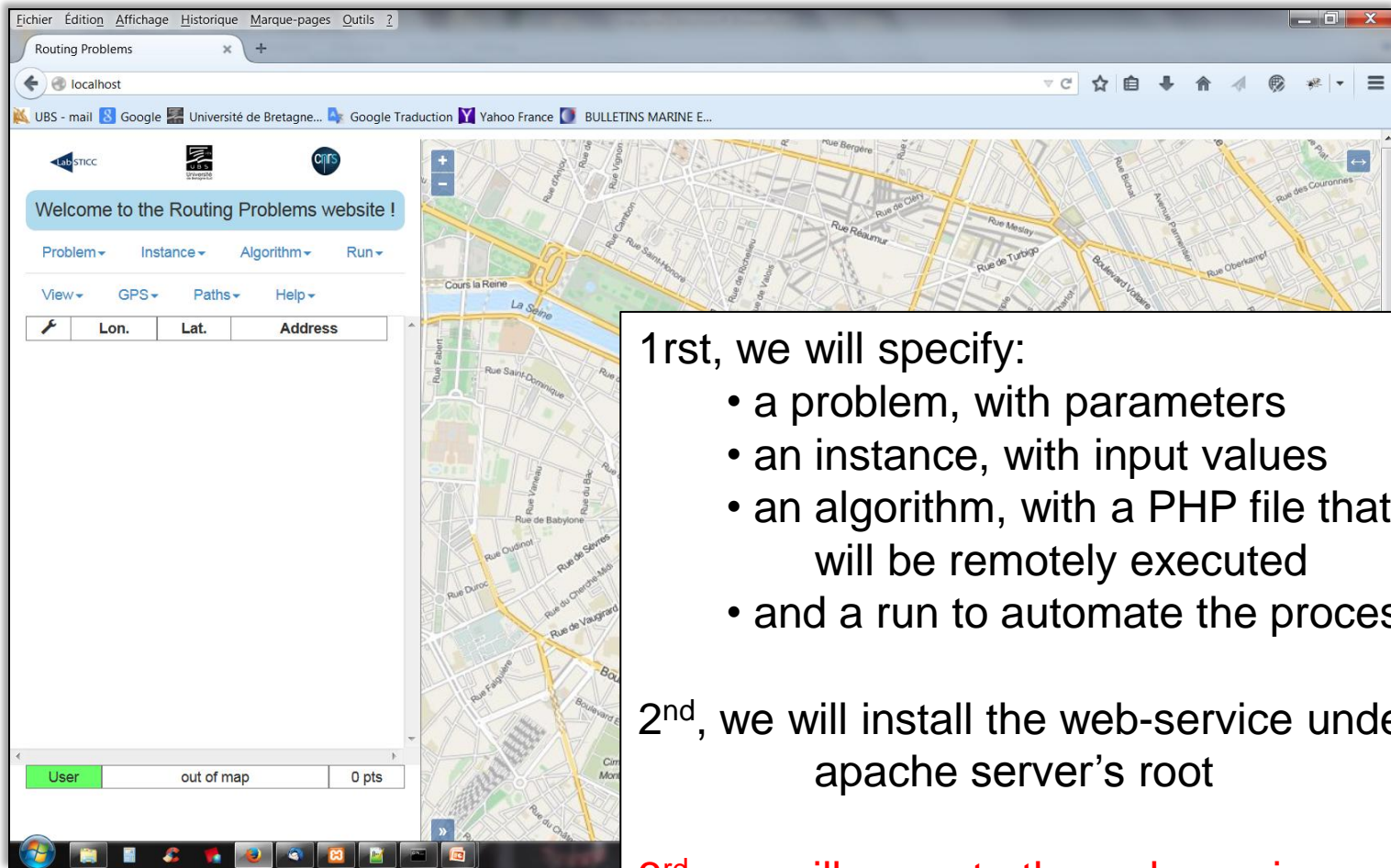


```
G:\projects\RP\work\services\problems\line\line_...
Fichier  Édition  Recherche  Affichage  Encodage  Langage
Paramétrage  Macro  Exécution  Compléments  Documents  ?  X
exec.php x
1  <?php
2  $a = json_decode($_REQUEST['a']);
3  $b = json_decode($_REQUEST['b']);
4  $x = json_decode($_REQUEST['x']);
5  $y=$a*$x+$b;
6  echo '{';
7  echo '"y":'. $y;
8  echo '}'';
9  ?>
10
Ln: 7  Col: 16  Sel: 0 | 0  Dos\Windows  UTF-8 w/o BOM  INS
```

This is all for the *line* web-service.

Save the file and exit.

# Execution of the web-service



1<sup>st</sup>, we will specify:

- a problem, with parameters
- an instance, with input values
- an algorithm, with a PHP file that will be remotely executed
- and a run to automate the process

2<sup>nd</sup>, we will install the web-service under the apache server's root

3<sup>rd</sup>, we will execute the web-service

# Execution: launch the web-service and see the result

The screenshot shows a web browser window displaying the 'Routing Problems' website. The interface includes a navigation menu with 'Problem', 'Instance', 'Algorithm', and 'Run' dropdowns. A table with columns 'Lon.', 'Lat.', and 'Address' is visible. A 'Run' button is highlighted in green. A dropdown menu is open, showing 'run\_line (line)' and a green 'Run' button. A text box with '2/ Click on the green button.' points to this button. Another text box with '1/ Click here to see all the runs.' points to the 'Run' dropdown. A third text box with '3/ See the result here.' points to the 'y=0' label in the bottom left corner of the map area.

1/ Click here to see all the runs.

2/ Click on the green button.

3/ See the result here.

	Lon.	Lat.	Address
User	2.33614 E	48.85983 N	0 pts

y=0



# Execution: change the values of the instance

Specify an instance

Problem name:

Instance name:  Comment:

Values

<input type="text" value="1"/>	a (real, constant)
<input type="text" value="2"/>	b (real, constant)
<input type="text" value="8"/>	x (real, variable)
<input type="text" value="0"/>	y (real, result)

1/ Enter these new input values.

2/ Don't touch **y**, it will be recomputed

Load parameters Save instance Delete instance

# Execution: launch the web-service and see the result

The screenshot shows a web browser window with the URL 'localhost'. The page title is 'Routing Problems'. The interface includes a navigation menu with 'Problem', 'Instance', 'Algorithm', and 'Run' dropdowns. Below the menu is a table with columns 'Lon.', 'Lat.', and 'Address'. A red arrow points from the 'Run' dropdown to a callout box containing a 'Run' button. Another red arrow points from the 'Run' button to a callout box. A third red arrow points from the 'Run' button to a callout box. A fourth red arrow points from the 'Run' button to a callout box. A fifth red arrow points from the 'Run' button to a callout box.

1/ Click here to see all the runs.

2/ Click on the green button.

3/ See the new result here.

# Execution: see the new values in the instance

Routing Problems

localhost

UBS - mail Google Université de Bretagne... Google Traduction Yahoo France BULLETINS MARINE E...

### Specify an instance

Problem name:

Instance name:  Comment:

Values

<input type="text" value="1"/>	a (real, constant)
<input type="text" value="2"/>	b (real, constant)
<input type="text" value="8"/>	x (real, variable)
<input type="text" value="10"/>	y (real, result)

3/ See the new value of y here.

Load parameters Save instance Delete instance

User out of map 0 pts  
y=10

# Conclusion

Now you know all the basics  
to start **developing**,  
**installing**  
and **testing**  
your own web-services.

Congratulations !

You have created your 1rst web-service.

