Cacique
Installation Guide

Indice

1. Introduction

The main objective of this guide is bringing into detail the necessary steps Cacique's tool correct installation. Explain the different existing processes and which the roles within the tool are.

2. ¿What is Cacique?

Cacique is a Web tool for automating tasks developed on Ruby on Rails. For its operation it is necessary that a web server is online constantly.

We recommend and confirm its correct behavior on Firefox.

As an initial requirement a computer is needed, to which we will refer later as Cacique server, with Ubuntu or Debian Operative system.

3. Installation

Cacique's Installation steps are found below, which must be executed always from Cacique's root directory.

(To know them in detail, read the paragraph Installation Steps)

```
./installer.sh
```

Edit the file: cacique/config/cacique.yml with the corresponding information, and then:

```
ruby config.rb

rake db:create  RAILS_ENV=production

rake db:migrate RAILS_ENV=production

sudo /etc/init.d/apache2 start
```

Ready!

We can navigate Cacique at the following url: http://<server_ip>/

4. Execution

In order to execute a script in Cacique it is necessary for some processes to run on the server (to know in detail each one of them, read the paragraph called Processes).

Cacique – do it once!   Installation Guide   http://cacique.mercadolibre.com

4.1 Local Execution

For executing in the user's local computer, run the following processes on the server:

```
/script/starling.rb

/script/workling_client start
```

4.2 Remote Execution

Optional for remote execution:

<path_to_project>/extras/selenium-grid-1.0.8 and execute in command line:

```
ant launch-hub
```

Then, run one or more RC agents in the computers you wish, pointing to the previously running Hub.

Cacique allows you to program executions; in order to do that run the following process on the server:

```
rake jobs:work RAILS_ENV=production
```

4.3 CaciqueOnMyComputer Execution

For the execution of a script on the same computer as the user, it is necessary to run an agent in that computer. For more information, read chapter 6.3.

5. Installation Steps

5.1 Installer

./installer.sh

It is necessary to install a series of packages that are specified in the installer.sh file, located on the root of the project Cacique.

In order to execute the installer, open a console, go to the root directory of the tool and execute the command **./installer.sh**

The different gems and packages that are being installed will start to appear on the console.

MySql Database engine is on the list of packages to install. When this installation starts, a password will be asked for the engine's root user. Remember this password, because you will need it later.

If a package can't be installed, it will be logged in a file called installer.log on the same directory where the installer is found. In that case we must re-execute the installer, or installing the pending package ourselves.

Once the installation is complete, we will have all the software packages needed to run Cacique in the server.
In order to continue, the user profile must be reloaded, executing the command:
source /etc/profile

5.2 Configuration

Configuration File (YML).

Cacique has a configuration file called "cacique.yml" that can be found inside the directory "cacique/config". This file must be used to configurate everything related to the application's operation:

Server IP
Database connection
Worker's IP
Selenium Grid IP
Memcached
etc.

```
*cacique.yml ✖

 1  :server:
 2    :ip: localhost
 3
 4  :db:
 5    :development:|
 6      :adapter: mysql
 7      :encoding: utf8
 8      :database: development
 9      :pool: 10
10      :username: root
11      :password: 1234
12      :host: localhost
13    :test:
14      :adapter: mysql
15      :encoding: utf8
16      :database: test
17      :pool: 10
18      :username: root
19      :password: 1234
20      :host: localhost
21    :production:
22      :adapter: mysql
23      :encoding: utf8
24      :database: production
25      :pool: 10
26      :username: root
27      :password: 1234
28      :host: localhost
29
```

We must remember that the numeric values (IP type or port) must be written between quotation marks. This will guarantees the correct configuration of the application.

In this step, amongst other configurations, we will generate Cacique's Database where all the information will be stored. MySql Database was installed in the installer executed at step 1, but in case you'd want any other Database, the steps are the same.

Once the configuration file is completed with the necessary data, it will be enough to execute the following command on our Cacique directory:

Ruby config.rb

Take into account that every time the file cacique/config/cacique.yml is changed, **Ruby config.rb** command must be executed for changes to be updated.

It is also important to configure correctly the time zone in which the Cacique server is found, for this, go into config/environment.rb and modify the default zone, Buenos Aires.
<path_to_project>

```
82    # Make Time.zone default to the specified zone, and make Active Record store time values
83    # in the database in UTC, and return them converted to the specified local zone.
84    # Run "rake -D time" for a list of tasks for finding time zone names. Comment line to use default local
85    config.time_zone = 'Buenos Aires'
86
```

5.3 DataBase

Once the configuration is done, open the console, go to Cacique's root directory and execute the following command **rake** db:create **RAILS_ENV=production**

This command is typical of a Rails application and creates the database according to the configuration specified in the file config/database.yml

Once the database is created you should run the migration to generate the tables, to do this you must run the command **rake db:migrate RAILS_ENV=production**

On the console, executed migrations will be shown as follows:

```
==  CreateQueueObservers: migrating =======================================
-- create_table(:queue_observers)
   -> 0.0115s
==  CreateQueueObservers: migrated (0.0117s) ==============================

==  AddLanguageColumnToUser: migrating ====================================
-- add_column(:users, :language, :string, {:limit=>5, :default=>"en_US"})
   -> 0.0236s
==  AddLanguageColumnToUser: migrated (0.0241s) ===========================
```

Once these migrations are completed, we will have a database ready for being used.

5.4 Navigating Cacique

As a last installation step we only have to set up our web server, in this case we'll use Apache with Passenger, but it can be any other such as Mongrel.

In order to do this, follow the instructions at this website: http://www.modrails.com/install.html

At this point, we are able to run Cacique server as follows:

- In case of using Apache with Passenger, open a console and execute the following command: **sudo /etc/init.d/apache2 start**

- In case of using Mongrel, open a console, login as root user using the **su** comando, enter password and then execute the following command: **script/server -e production -p 80**

Now, we can navigate Cacique at the url http://<server_ip>/

6. Execution

6.1 Local Execution

This section gives a brief explanation of the role of each process within the architecture of Cacique.

Apache/Passenger: Cacique's web page server.

MySql: Cacique's database engine for storing.

Memcached: Rails's application Cache.
To run this service, use the command line:
/usr/bin/memcached -u cacique -d

Starling: Process that administrates queues and workers. It receives different executions that are generated on Cacique, queues them up and sends them to different connected workers for their execution.
To run this service, use the command line:
<path_to_project>/script/starling.rb

Workling: A process that runs in the background that is used for executing in Cacique.
To run this service, use the command line:
<path_to_project>/script/workling_client start
Execute this command as many times as workers you'd want to run.

Selenium-Grid: Process used to execute Selenium scripts in different browser configurations.
To run this service, go to **<path_to_project>/extras/selenium-grid-1.0.8** and use the command line:
ant launch-hub

For more Selenium-Grid documentation, you can visit: http://selenium-grid.seleniumhq.org/

Jobs: Rails process that is responsible for implementing programmed suites.
To run this service, go to **<path_to_project>/extras/selenium-grid-1.0.8** and use the command line:
rake jobs:work RAILS_ENV=production

Each process in Cacique is completely detachable from the other, they can all be found on the same server, or distributed in different.

6.2 Remote Execution

Cacique can run its tests on remote computers or on the user's computer.
For remote execution, the following is needed:

     1- Having a computer installed with the OS and the browser in which we want to run the test.

     2- Run a Selenium agent. For this, it is required to copy the directory that is can be found in /extras/selenium-grid-1.0.8 to the remote machine, open the console, go to the copied directory and execute the following command:

          ant -Dport=<port> -Denvironment="<tag>" -DseleniumArgs="-singleWindow -trustAllSSLCertificates " -Dhost=<remote_ip> -DhubURL=http://<server_ip>:4444 launch-remote-control

  Replace the values between <> in the command, where:

     <port>: Any free port where the agent will stay connected.

     <tag>: Configuration name that represents the agent. E.g.: Firefox3 on Windows.

     <remote_ip>: ip of the remote computer, it will be used along with the label to identify an agent.

     <server_ip>: Ip of the computer where Cacique's server is located.

After following these two steps, Cacique is in conditions of executing a script on a remote computer.
In order to see the active agents, you can go to: http://<server_ip>:4444/console

Note that the "labels" that I use to lift the agents have to be valid labels that Selenium Hub understands. To configure the Hub labels modify the file <path_to_project>/extras/selenium-grid-1.0.8/grid_configuration.yml

6.3 CaciqueOnMyComputer Execution

     For the execution of a script on the same computer as the user, it is necessary to run an agent in that computer. For that, follow these steps:

     1- Download and extract in Cacique's home the .zip file that can be found here Cacique on My Computer.

     2- Execute the runFirefox.bat or runIE.bat, depending on the browser you wish to use. A console will open and will show the running agent. Both agents can be running simultaneously.

After executing these two steps, Cacique is in conditions of executing a script on the user's computer.

7. Contacts

For more information, please visit:
 http://cacique.mercadolibre.com