

r au cœur d'applications web pour la gestion et le traitement des données sur serveur Apache

Boulain Nicolas (SPN)
10 juin 2016



rApache



Présenté pour la première fois en 2005, rApache permet le développement d'applications web en utilisant le langage et l'environnement R et le serveur web Apache.

Les distributions actuelles de rApache ne tournent que sous les environnements UNIX/Linux et Mac OS X.

Jeffrey Horner (2013). rApache: Web application development with R and Apache.
<http://rapache.net/>

r et les applications Web

Plusieurs projets ont été développés pour répondre aux besoins d'utiliser r en interaction avec le web

- shiny, permet de créer serveur et interface
- rserve, permet d'interfacer un serveur rserve et des applications java/C/C++
- rsoap, plutôt orienté webservice
- Sumo, pour techno java, tourne avec tomcat
- openCPU, javascript API qui appel un serveur R
- gWidgetsWWW
- ...

rApache permet de construire aussi bien des applications web que des web services

Installation de rApache

PRES REQUIS

- une machine sous UNIX/Linux ou Mac OS
- serveur apache installé
- Apache Prefork Multi-Processing Module recommandé
- r installé avec les librairies souhaitées

Installation de rApache

INSTALLATION

- Depuis une Debian

```
>$ sudo apt-get install devscripts git
>$ sudo apt-get install apache2-prefork-dev apache2-mpm-prefork
libapreq2-dev r-base-dev
>$ git clone https://github.com/jeffreyhorner/rapache.git
>$ cd rapache
>$ debuild -us -uc
>$ cd ..
>$ sudo dpkg -i libapache2-mod-r-base*.deb
```

- Depuis une UBUNTU (>=12)

```
>$ sudo add-apt-repository ppa:opencpu/rapache
>$ sudo apt-get update
>$ sudo apt-get install libapache2-mod-r-base
```

Installation de rApache

CONFIGURATION

- mod_R

Vérifier la présence, sinon créer le fichier **mod_R.load** dans **/etc/apache2/mods-available** qui contient:

```
LoadModule R_module /usr/lib/apache2/modules/mod_R.so
```

Puis rendre accessible le module **mod_R.load** avec la commande:

```
sudo a2enmod mod_R
```

Installation de rApache

CONFIGURATION

- site.conf

Passer par un hôte virtuel dans, **/etc/apache2/sites-available** , pour configurer le futur site avec les instructions spécifiques rapache

exemple: fichier testrapache.conf:

```
<VirtualHost *:80>  
</VirtualHost>
```

cf https://doc.ubuntu-fr.org/tutoriel/virtualhosts_avec_apache2 sur ubuntu-fr

Puis activer le site par:

```
sudo a2ensite testrapache.conf
```

Configuration de rApache

OPTION CONFIGURATION

- chargement de librairies et choix du dossier de travail au démarrage du serveur

```
<VirtualHost *:80>  
  bla bla bla  
  
  REvalOnStartup "library(brew); setwd('/var/www/html/R') "  
  
  bla bla bla  
</VirtualHost>
```


Configuration de rApache

OPTION CONFIGURATION

- Exécution d'un script r (source) au démarrage du serveur

```
<VirtualHost *:80>  
  bla bla bla  
  
  RSourceOnStartup "./R/startup.r"  
  
  bla bla bla  
</VirtualHost>
```

Configuration de rApache

OPTION CONFIGURATION

- les options de SetHandler

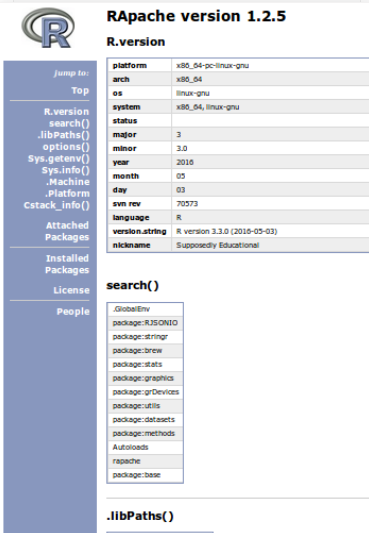
exemple simple:

r-info pour avoir un rapport détaillé de la version de R qui tourne sur le serveur

```
<VirtualHost *:80>
  bla bla bla

# http://serveur.bla.bla/RApacheInfoinfo
<Location /RApacheInfoinfo>
  SetHandler r-info
</Location>

  bla bla bla
</VirtualHost>
```



RApache version 1.2.5

R.version

platform	x86_64-pc-linux-gnu
arch	x86_64
os	linux-gnu
system	x86_64, linux-gnu
status	
major	3
minor	3.0
year	2010
month	05
day	03
svn rev	70573
language	R
version.string	R version 3.3.0 (2010-05-03)
nickname	Supposedly Educational

search()

GlobalEnv
package:R33CRND
package:stringr
package:brew
package:stats
package:graphics
package:grDevices
package:utils
package:datasets
package:methods
Autoloads
rapache
package:base

.libPaths()

Configuration de rApache

OPTION CONFIGURATION

- les 2 options utiles de SetHandler

-> r-handler

Appel d'une fonction R, généralement dans une instruction <Location>

-> r-script

Appel d'une fonction R , généralement dans une instruction <Directory>

Configuration de rApache

OPTION CONFIGURATION

- Les directives RHandler, RFileHandler, REval et RFileEval

Pour chaque requêtes, on peut évaluer:

-> spécifier une fonction R : RHandler

```
RHandler foo::bar
```

-> spécifier un fichier ou une fonction dans un fichier R : RFileHandler

```
RFileHandler /var/www/R/bar.R::foo
```

-> spécifier une expression R : REval

```
REval "foo::bar()"
```

-> spécifier une expression contenue dans un fichier R : RFileEval

```
RFileEval "/var/www/R/bar.R:foo()"
```

Configuration de rApache

OPTION CONFIGURATION

- r-handler exemple

```
<VirtualHost *:80>
  bla bla bla

  # L'accès à l'emplacement serveuradresse:80/Test
  # entraîne l'exécution du script test.r situé dans /var/www/html/R

  <Location /Test>
    SetHandler r-handler
    RFileHandler /var/www/html/R/test.r
  </Location>

  bla bla bla
</VirtualHost>
```

Configuration de rApache

OPTION CONFIGURATION

- r-script exemple

```
<VirtualHost *:80>
  bla bla bla

  # Tous les fichiers contenus dans /var/www/html/R/brew
  # sont interprétés avec les fonctions brew du package brew

  <Directory /var/www/html/R/brew>
    SetHandler r-script
    RHandler brew::brew
  </Directory>

  bla bla bla
</VirtualHost>
```

Fonctions de rApache

FONCTIONNALITES UTILISABLES DANS LE CODE R

- Fonctions rApache

Function	Exemple
setHeader	setHeader(header='X-Powered-By',value='rApache')
setContentType	setContentType(type='image/png')
setCookie	setCookie(name='sessionID',value=paste(rnorm(1)))
urlEncode and urlDecode	urlDecode(str='hello+world%40example.com')
sendBin	sendBin(object=readBin(t,'raw',n=file.info(t)\$size))
receiveBin	receiveBin(length=8192)
setStatus	setStatus(status=200L)

- rApache Variables

- > GET
- > POST
- > COOKIES
- > FILES
- > SERVER

Utilisation de rApache

- Exemple de la documentation :

1- configurer le virtualhost:

```
<VirtualHost *:80>
  bla bla bla

  <Location /rapachetest>
    SetHandler r-handler
    RFileHandler /var/www/R/test.R
  </Location>

  bla bla bla
</VirtualHost>
```


Utilisation de rApache

- Exemple de la documentation :
(<https://github.com/jeffreyhorner/rapache/blob/master/test/test.R>)

2- le code du fichier test.R part 1

```
hrefify <- function(title) gsub('[\\.(\\)]', '_', title, perl=TRUE)
scrub <- function(str){
  if (is.null(str)) return('NULL')
  [...]
}
cl<-'e'
zebary <- function(i){
  cl <- ifelse(cl=='e', 'o', 'e')
  cat('<tr class=""', cl, '"><td>', scrub(i), '</td></tr>\n', sep='')
}
zeblast <- function(i,l){
  cl <- ifelse(cl=='e', 'o', 'e')
  cat('<tr class=""', cl, '"><td class="l">', names(l)[i], '</td><td>')
  [...]
  cat('</td></tr>\n', sep='')
}
zebra <- function(title,l){
  cat('<h2><a name=""', hrefify(title), '"> </a>', title, '</h2>\n<table>
<tbody>', sep='')
  ifelse(is.list(l), lapply(1:length(l), zeblast, l), lapply(l, zebary))
  cat('</tbody></table>\n<br/><hr/>')
}
```

Utilisation de rApache

- Exemple de la documentation :

2- le code du fichier test.R part 2

```
# Output starts here
setContentType("text/html")
if(is.null(GET)){  called <- 1} else { called <-
as.integer(GET$called) + 1}

setCookie('called',called,expires=Sys.time()+100)
setCookie('anotherCookie','foo',expires=Sys.time()+100)
cat('<HTML><head><style type="text/css">\n')
[... ]
cat('</style></head><BODY><H1>Canonical Test for RApache</H1>\n')
cat('<form enctype=multipart/form-data method=POST action="/test
/R?called=',called,'">\n',sep='')
cat('Enter a string: <input type=text name=name value="">
<br>\n',sep='')
cat('Enter another string: <input type=text name=name value="">
<br>\n',sep='')
cat('Upload a file: <input type=file name=fileUpload><br>\n')
cat('Upload another file: <input type=file name=anotherFile><br>\n')
cat('<input type=submit name=Submit>')
cat("<hr>\n")
zebra('CGI GET Data',GET)
zebra('CGI POST Data',POST)
zebra('Cookies',COOKIES)
if (!is.null(FILES)){  cat('<h2>Files Uploaded in POST
Data</h2>\n')  for (n in names(FILES)){  zebra(paste("Form
Variable",n),FILES[[n]])  }}
cat("</BODY></HTML>\n")
```

Utilisation de rApache

- Exemple de la documentation :

2- resultat du fichier test.R en allant sur <http://leserveur/rapachetest>

Canonical Test for rApache

Enter a string:

Enter another string:

Upload a file: Aucun fichier sélectionné.

Upload another file: Aucun fichier sélectionné.

CGI GET Data

.

CGI POST Data

.

Cookies

.

Utilisation de rApache

- Exemple de la documentation :

2- resultat du fichier test.R en allant sur <http://leserveur/rapachetest>

Canonical Test for RApache

Enter a string:

Enter another string:

Upload a file: Aucun fichier sélectionné.

Upload another file: Aucun fichier sélectionné.

CGI GET Data

called	1
--------	---

CGI POST Data

name	test_test
name	bidon
fileUpload	cities.R
anotherFile	Classeur3.csv
Submit	Envoyer

Cookies

called	1
anotherCookie	foo

Files Uploaded in POST Data

Form Variable fileUpload

name	cities.R
tmp_name	/tmp/apreqhCiCi5

Utilisation de rApache

- Exemple 2: du code R dans la page html (fichier test placé dans le répertoire brew)

Configuration du virtualhost:

```
<VirtualHost *:80>
  bla bla bla

  <Directory /var/www/html/projectName/lib/brew>
    SetHandler r-script
    RHandler brew::brew
    AddDefaultCharset UTF-8
  </Directory>

  bla bla bla
</VirtualHost>
```

Tous les fichiers contenus dans le répertoire lib/brew seront interprétés à l'aide des fonctions brew

Utilisation de rApache

- Exemple 2 : du code R dans la page html (fichier test placé dans le répertoire brew)

```
<html><body>
<h1> Test RApache: distribution normale avec parametres</h1>
  <form action="helloworld" method="POST">
    n: <input type="text" name="n" /> <br />
    mean: <input type="text" name="mean" /> <br />
    sd: <input type="text" name="sd" /> <br />
    <input type="submit" value="Submit!">
  </form>
</body></html>
```

L'affichage en allant sur <http://leserveur/lib/brew/test>

Test RApache: distribution normale avec parametres

n:

mean:

sd:

Utilisation de rApache

- Exemple 2 : du code R dans la page html

Insertion du code R

```
<html><body>
<h1> Test RApache: distribution normale avec parametres</h1>
  <form action="helloworld" method="POST">
    n: <input type="text" name="n" /> <br />
    mean: <input type="text" name="mean" /> <br />
    sd: <input type="text" name="sd" /> <br />
    <input type="submit" value="Submit!">
  </form>

  <pre><code>
  <%
    myN <- as.numeric(POST$n);
    myMean <- as.numeric(POST$mean);
    mySD <- as.numeric(POST$sd);

    if(length(as.numeric(POST$n))!=0){
      print(rnorm(myN,myMean,mySD));
    }else{
      cat("Veuillez completer le formulaire!")
    }
  %>
  </code></pre>
</body></html>
```

Utilisation de rApache

- Exemple 2 : du code R dans la page html

Test RApache: distribution normale avec parametres

n:
mean:
sd:

Veuillez completer le formulaire!

Test RApache: distribution normale avec parametres

n:
mean:
sd:

```
[1] 331.0350 370.0926 287.9225 356.2567 340.5785 409.7865 345.7911 309.9915
[9] 309.7681 390.5095 345.8107 354.3785 334.6345 373.0865 320.0961 364.2655
[17] 318.9069 325.5643 337.9705 297.4285 335.0497 342.0625 394.0506 398.7720
[25] 348.2410 404.6706 339.2094 354.1770 329.3939 311.3029 293.7014 310.0210
[33] 363.6871 385.5446 412.7816 344.8941 412.0492 320.0537 376.6882 345.6207
[41] 363.9530 399.4394 334.0684 373.5627 392.4540 302.9425 325.3520 437.5525
[49] 334.4000 376.9831 295.0764 293.2388 293.1149 379.9747 363.4714 327.7711
[57] 339.8916 422.0361 302.3910 350.6818 327.3254 402.0420 315.8564 315.9643
[65] 339.0549 345.2255 305.6953 347.0231 287.8303 379.9307 307.8797 322.3456
[73] 342.5704 292.6195 347.0472 301.9303 332.7140 308.0726 357.5095 318.1995
[81] 321.7164 313.4264 400.2740 415.0123 325.1375 324.1170 368.0288 361.4029
[89] 323.2832 349.3423 368.5108 352.6467 308.9527 393.0619 315.6523 332.7774
[97] 392.0175 360.6302 367.6546 372.0849 331.0995 423.3518 367.5563 382.4268
[105] 299.0261 332.5271 347.2466 415.9749 335.6594 318.8300 380.5427 338.1481
[113] 298.7387 441.5588 305.9758 340.1357 328.3172 280.1408 317.7394 379.6759
```


Utilisation de rApache

- Exemple 2 : du code R dans la page html

gestion des erreurs

Test RApache: distribution normale avec parametres

n:

mean:

sd:

Utilisation de rApache

- Exemple 2 : du code R dans la page html

gestion des erreurs

```
myVal<-function(){
  myN <- as.numeric(POST$n);
  myMean <- as.numeric(POST$mean);
  mySD <- as.numeric(POST$sd);

  if(length(as.numeric(POST$n))!=0){
    print(rnorm(myN,myMean,mySD));
  }else{
    cat("Veuillez completer le formulaire!")
  }
}
printError<-function(e){print(e$message);}
tryCatch(myVal(),error=printError);
```

Test RApache: distribution normale avec parametres

n:

mean:

sd:

[1] "invalid arguments"

Utilisation de rApache

- Exemple 3 : html, ajax et R le code html simple

```
<html>
  <head>
    <link rel="shortcut icon" href="icon2.png" />
    <title>Indicateurs</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com
/bootstrap/3.3.6/css/bootstrap.min.css">
    <link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs
/font-awesome/4.4.0/css/font-awesome.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0
/jquery.min.js"></script>
    <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6
/js/bootstrap.min.js"></script>
    <script src="//d3js.org/d3.v3.min.js" charset="utf-
8"></script>
  </head>
  <body>

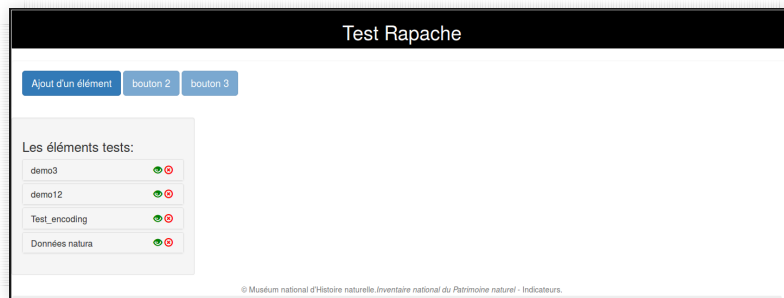
    <nav class="navbar navbar-default ">
      <div class="container-fluid text-center" style="background-
color:black">
        <div class="navbar-head ">
          <h1 style="color:white">Test Rapache</h1>
        </div>
      </div>
    </nav>

[...]
```

Utilisation de rApache

- Exemple de la documentation :

2- resultat du fichier test.R en allant sur <http://leserveur/>



Utilisation de rApache

- Exemple 3 : html, ajax et R
le code html simple

```
[...]
  <div class="modal" id="elementAjoutModal" tabindex="-1"
role="dialog" aria-labelledby="myModalLabel">
  <div class="modal-dialog modal-lg" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"
aria-label="Close"><span aria-hidden="true">&times;</span></button>
        <h4 class="modal-title"
id="collectionAnalyseModalLabel">Création d'un nouvel indicateur</h4>
      </div>
      <div class="modal-body">
        <!-- from pour parametre analyse à afficher dans une modal-->
        <form role="form" id="formAjoutElm" method="POST" >

          </form>
        </div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default"
data-dismiss="modal">Fermer</button>
        <button type="button" class="btn btn-primary" id="b1"
data-dismiss="modal">Créer </button>
      </div>
    </div>
  </div>
</div>
[...]
```

Utilisation de rApache

- Exemple 3 : html, ajax et R
A l'affichage:

Création d'un nouvel élément

L'element:

Nom de l'element:

Libellé court:

Définition courte:

Version:

Méthodes:

Description technique:

Objectifs:

Limites et avantages:

Dates d'utilisation :

Début : Fin :

Personnes :

Responsable: Commanditaire:

Tags :

Utilisation de rApache

- Exemple 3 : html, ajax et R
le code javascript pour l'asynchrone

```
<script>
$(document).ready(function(){
  $("#b1").click(function(e){
    var formdata =
JSON.stringify($("#formAjoutElm").serializeArray());
    var finish = false;

    $('#elementAjoutModal').toggle('true');
    document.getElementsByClassName('modal-backdrop')
[0].parentNode.removeChild(document.getElementsByClassName('modal-
backdrop')[0]);

    var request = $.ajax({
      type: "POST",
      url: "../lib/brew/createElement.rhtml",
      data: formdata,
      contentType: "application/json"
    });
    request.done(function(msg, ret_status, xhr) {
      finish = true;
      console.log("reponse principale : "+xhr.responseText);
      getIndic();
    });
  });
});
</script>
```

Utilisation de rApache

- Exemple 3 : html, ajax et R
le code du fichier createElement.rhtml (début)

```
<%  
# Nettoyage de la memoire  
rm(list=ls())  
init<-as.numeric(format(Sys.time(), "%s"))  
Sys.setenv(NLS_LANG="FRENCH_FRANCE.AL32UTF8")  
  
# appel des librairies  
suppressPackageStartupMessages(library(stringr))  
suppressPackageStartupMessages(library(DBI))  
suppressPackageStartupMessages(library(ROracle))  
suppressPackageStartupMessages(library(data.table))  
suppressPackageStartupMessages(library(RJSONIO))  
  
# reception du POST  
binpost <- receiveBin()  
x <- rawToChar(binpost)  
  
# transformation des données POST pour mettre au format data.frame  
plus facilement  
x<-str_replace_all(x,"\"name\":",",")  
x<-str_replace_all(x,"\"value\":",",")  
x<-str_replace_all(x,"\"",",","\":")
```


Utilisation de rApache

- Exemple 3 : html, ajax et R
le code du fichier createElement.rhtml (fin)

```
# creation du data.frame correspondant aux données POST
if (isValidJSON(x, TRUE)) {
  data<-data.frame(t(unlist(fromJSON(x, simplifyWithNames=FALSE))))
  ora <- Oracle()
  conex <- dbConnect(Oracle(),"user","mdp","database")

  requet<-paste("Mettre ici la commande sql")

  MTDindic <- try(dbGetQuery(conex, requet))
  if(class(MTDindic) != "try-error"){
    dbCommit(conex)
    dataRetour<-data.frame(POSTjson = "Ok INSERT MTD")

  }else{
    dataRetour<-data.frame(POSTjson = "FAILED INSERT MTD")
  }
  dbDisconnect(conex)

}else{
  dataRetour<-data.frame(POSTjson = "ERROR JSON POST invalid")
}
json<-toJSON(dataRetour)
DONE
%>

<%=cat(json) %>
```

Utilisation de rApache

- Exemple 3 : html, ajax et R
A l'affichage:

The screenshot displays a web application interface with a modal dialog box and a data entry form. The modal dialog, titled "NOM_INDICATEUR :", contains a text input field with the value "demo3" and two buttons: "Close" and "Valider".

The background interface features a top navigation bar with buttons "Ajout d'un élément", "bouton 2", and "bouton 3". On the left, a section titled "Les éléments tests:" lists three items: "demo3" (with a green eye icon), "demo12" (with a green eye icon), and "Test_encoding" (with a green eye icon). Each item has a red 'x' icon next to it. The "demo3" item is expanded, showing details: "Libellé : toto", "Responsable : bidon", and "Version : 2".

The main content area displays a form with the following fields:

- NOM_INDICATEUR :** demo3
- LIBELLE_COURT :** toto
- METHODE_TECHNIQUE :** Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.
- RESPONSABLE :** bidon

At the bottom of the form, there is a copyright notice: "© Muséum national d'Histoire naturelle. Inventaire national du Patrimoine naturel - Indicateurs."

Les +, les - de rApache

Les Plus

- serveur personnalisé
- installation rapide, peu de configuration
- souplesse d'utilisation: html écrit dans R ou R appelé par html
- R devient le langage serveur, pas besoin de connaître php, python, java
- indépendant, ce n'est pas du rstudio

Les Moins

- serveur personnalisé
- connaissance de base pour apache, html, js... et R
- monothread, perte dans la gestion des env, difficile de sourcer du code depuis le code principale
- indépendant, ce n'est pas du rstudio, beaucoup moins de support