

R语言公众号后台实践

上海 2017

郎大为

J.D. Power

为什么有这个�故事

为什么有这个�故事

- 大面积封杀非官方网页接口^[*]

为什么有这个�故事

- 大面积封杀非官方网页接口^[*]
- R语言的部署
 - 结果插入到数据库
 - 部署API
 - 直接构建服务器

为什么有这个�故事

- 大面积封杀非官方网页接口^[*]
- R语言的部署
 - 结果插入到数据库
 - 部署API
 - 直接构建服务器
- 某些微信群、朋友的定制化需求
 - 一个不读书就被踢出去的群想记录群友读什么书
 - 朋友想做一个可以跟用户聊天的机器人

[*]没错我指的是itchat

开发

按照开发资源划分

- 回复机器人: 微信公众号(个人)
- 后台机器人: 微信公众号(个人), 一台服务器(带外网IP)
- 用户管理: 认证公众号(认证)
- 自动生成图文/自动群发: 需要一个认证公众号(认证)
- 小程序: 微信开发走好不送

0行代码绑定机器人

绑定机器人

- 免费账号每天可以获得1000次调用机会

图灵机器人 智能机器人走进每个家庭
TURING ROBOT

机器人管理 >> 读书君 >> 机器人设置

读书君 ▾

- 机器人概览
- 机器人设置
- 知识库
- 能力扩展
- 数据分析
- 版本升级

机器人昵称： 机器人年龄：

体验版

机器人性别： 男 女 机器人星座：

妈妈的名字： 爸爸的名字：

快速接入

一个写读书笔记的人

[*]实在不行你申请十个账号

用R调用图灵机器人

```
library(httr)
library(jsonlite)
getAnswer = function(str){
  url = 'http://www.tuling123.com/openapi/api'
  apiInfo = list(key = '【**YOURAPIKEY**】',
                 info = str,
                 userid = '123')
  answer = POST(url, body=toJSON(apiInfo, auto_unbox = T))
  fromJSON(rawToChar(answer$content))
}
startRobot = function(){
  repeat{
    str = readline(prompt = ">>>")
    if(str=='') break
    cat(getAnswer(str)$text)
  }
  invisible(NULL)
}
```

~~【*】谁也别给我提RCurl好用，能用httr就别用RCurl~~

用R调用图灵机器人

```
> startRobot()
>>>会讲话不
你想说什么啊？
>>>你叫啥
我是棒棒哒读书君
>>>好累，幻灯片做到了凌晨
宝宝快休息会儿
>>>你太照顾我了
你是小甜心吗这么贴心。
>>>好吧，我带你上电视
电视是生活里的好帮手呀。
>>>又蠢又萌，再见
我们聊点别的什么，好不？
>>>
> |
```

构建后台服务器

R构建服务器

- shiny: 依赖于httpuv, 一个websockets的服务
- openCPU: 部分服务器的形式比较复杂
- fiery: 可以构建各种符合http要求的服务器

[*]~~plumber也可以用来构建服务器的包，但我没用过，就这样吧~~

fiery

一个只用于做服务器的R包

- Shiny uses magic to make everything work from R,
 - Fiery lets you do all the hard work.
- Shiny wants the main app-logic to be server-side,
 - Fiery don't care what you do.
- Shiny uses a reactive model to define the app-logic,
 - Fiery don't care what you do (see a pattern emerge).
- Shiny wants you to use htmltools to build the html,
 - Fiery really don't care what you use.

fiery

```
library(fiery)

# Create a New App
app <- Fire$new()

# Setup the data every time it starts
app$on('start', function(server, ...) {
  server$set_data('visits', 0)
  server$set_data('cycles', 0)
})

# Count the number of cycles (internal loops)
app$on('cycle-start', function(server, ...) {
  server$set_data('cycles', server$get_data('cycles') + 1)
})
```

fiery

```
# Count the number of requests
app$on('before-request', function(server, ...) {
  server$set_data('visits', server$get_data('visits') + 1)
})

# Handle requests
app$on('request', function(server, request, ...) {
  response <- request$respond()
  response$status <- 200L
  response$body <- paste0('<h1>This is indeed a test. You are numb')
  response$type <- 'html'
})

# Show number of requests in the console
app$on('after-request', function(server, ...) {
  message(server$get_data('visits'))
  flush.console()
})
```

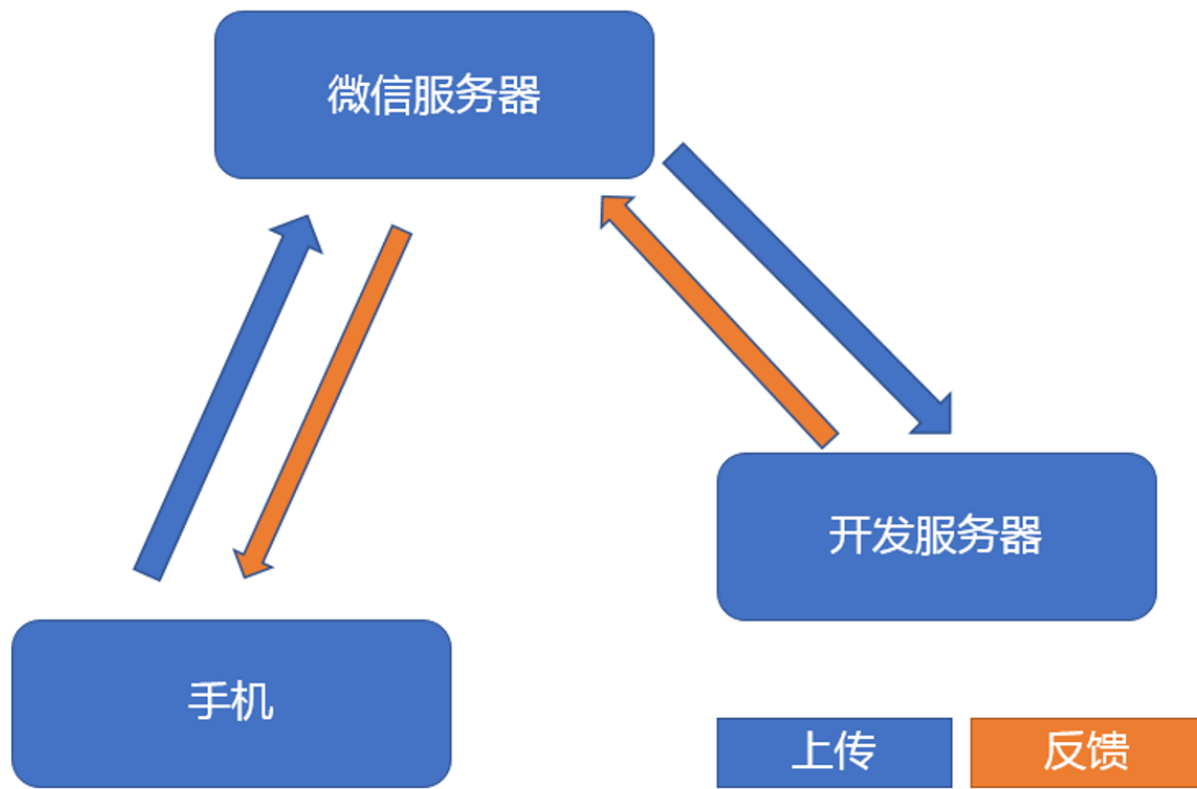
fiery

```
# Terminate the server after 50 cycles
app$on('cycle-end', function(server, ...) {
  if (server$get_data('cycles') > 50) {
    message('Ending...')
    flush.console()
    server$extinguish()
  }
})

# Be polite
app$on('end', function(server) {
  message('Goodbye')
  flush.console()
})

app$ignite(showcase = TRUE)
#> Fire started at 127.0.0.1:8080
#> 1
#> Ending...
#> Goodbye
```


服务器原理



与微信服务器沟通交互

1. 验证身份
2. 微信服务器传递信息
3. 内部处理（存储，反馈）
4. 回复相应信息

1. 验证身份

开发者提交信息后，微信服务器将发送GET请求到填写的服务器地址URL上，GET请求携带参数如下表所示：

参数	描述
signature	微信加密签名，signature结合了开发者填写的token参数和请求中的timestamp参数、nonce参数。
timestamp	时间戳
nonce	随机数
echostr	随机字符串

开发者通过检验signature对请求进行校验（下面有校验方式）。若确认此次GET请求来自微信服务器，请原样返回echostr参数内容，则接入生效，成为开发者成功，否则接入失败。加密/校验流程如下：

- 1) 将token、timestamp、nonce三个参数进行字典序排序
- 2) 将三个参数字符串拼接成一个字符串进行sha1加密
- 3) 开发者获得加密后的字符串可与signature对比，标识该请求来源于微信

1. 验证身份

```
if('echostr' %in% names(theQuery)){  
  response$body = regmatches(request$querystring,  
                              gregexpr("(?<=echostr=).+(?=&t)",  
                                         request$querystring,  
                                         perl = TRUE))  
}else{  
  print(123)  
  response$body = returnMsg(ori,user, time, msgType, content, message)  
}
```

2. 微信服务器传递消息

```
<xml>  
  <ToUserName><![CDATA[toUser]]></ToUserName>  
  <FromUserName><![CDATA[fromUser]]></FromUserName>  
  <CreateTime>1348831860</CreateTime>  
  <MsgType><![CDATA[text]]></MsgType>  
  <Content><![CDATA[this is a test]]></Content>  
  <MsgId>1234567890123456</MsgId>  
</xml>
```

2. 微信服务器传递消息

```
# 微信数据截取函数
extractWeixin = function(msgXML, pattern,CDATA = T){
  if(CDATA)
    regPattern = paste0("(?<=", pattern, "><\\!\\[CDATA\\[\\s\\s]+
                        pattern, ")")
  else
    regPattern = paste0("(?<=", pattern, ">).+(?=</",
                        pattern, ")")
  regmatches(msgXML,
             gregexpr(regPattern,
                     msgXML,
                     perl = TRUE))[[1]]
}
```

3.内部处理（存储，反馈）

- 保存到数据库中？
- 调用个API（自己写个机器人）？
- 放个深度学习的框架？
- 。。。

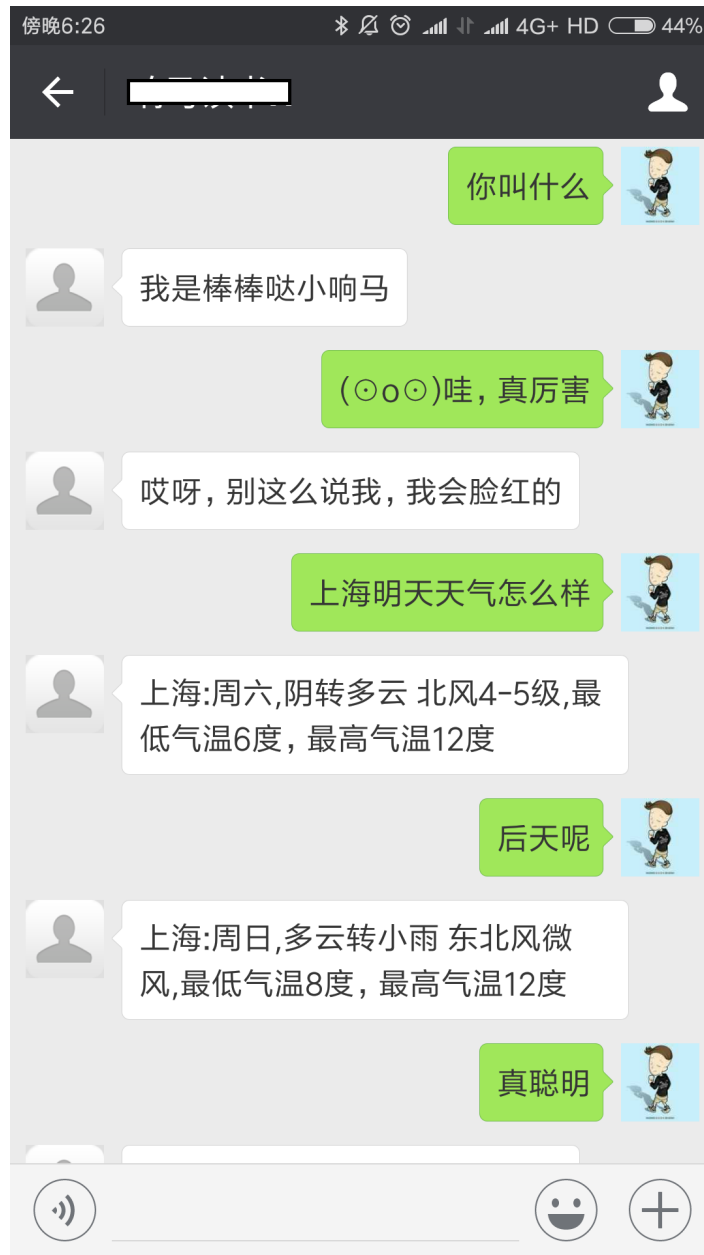
4. 反馈信息

```
output = sprintf("<xml>  
  <ToUserName><![CDATA[%s]]></ToUserName>  
  <FromUserName><![CDATA[%s]]></FromUserName>  
  <CreateTime>%s</CreateTime>  
  <MsgType><![CDATA[text]]></MsgType>  
  <Content><![CDATA[介尼玛似一个%s]]></Content>  
</xml>", user, ori, as.numeric(Sys.time()), showPic(filename))
```


聊天机器人

- 调用图灵的API来构建聊天机器人

你说我怎么向你们证明这真是一个机器人

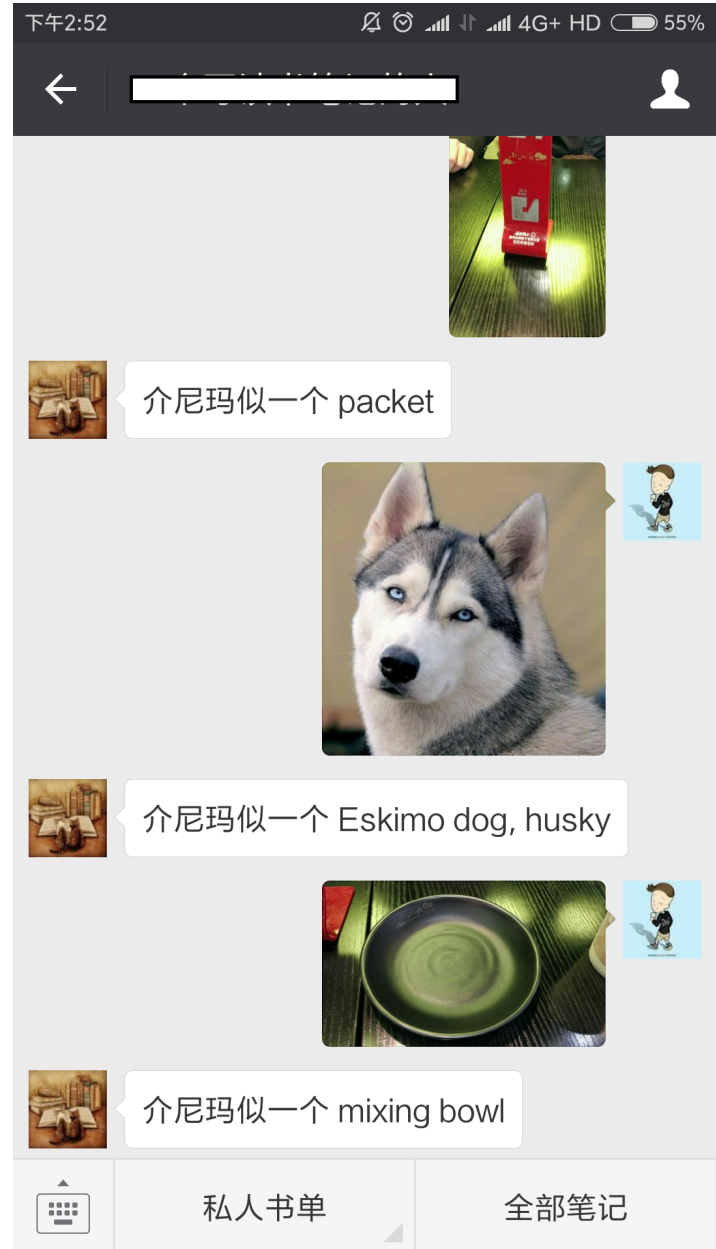


深度学习部署

- 部署一个深度学习的模型
- 基于Mxnet训练好的图像识别模型
- 用户发送图片给公众号
- 公众号根据图片进行识别，回复识别结果
- <https://github.com/Lchiffon/Example-for-R-Weixin>

[*]可以用来背单词





代码结构

```
app <- Fire$new()
app$on('request', function(server, request, ...) {
  #1. 验证身份
  #2. 微信服务器传递信息
  #3. 内部处理（存储，反馈）
  #4. 回复相应信息
})
app$on('end', function(server) {
  flush.console()
})
app$ignite(showcase = TRUE)
```

```
returnMsg = function(ori,user, time, msgType, content, messageId,
                    PicUrl){
  if(length(PicUrl)==0){
    cat(234)
    return('success')
  }
  cat(123)
  filename = paste0("data/",format(Sys.time(),"%Y%m%d%M"))
  download.file(PicUrl,destfile = filename)

  output = sprintf("<xml>

    <ToUserName><![CDATA[%s]]></ToUserName>

    <FromUserName><![CDATA[%s]]></FromUserName>

    <CreateTime>%s</CreateTime>

    <MsgType><![CDATA[text]]></MsgType>

    <Content><![CDATA[介尼玛似一个%s]]></Content>

  </xml>",user,ori,as.numeric(Sys.time()),showPic(filename))
  return(output)
}
```

Rweixin

Rweixin

- 一个用于用户管理，素材管理的package
- 需要一个认证的账号
- <http://github.com/lchiffon/Rweixin>

```
library(Rweixin)
```

```
AppID = '[**你的APPID**]'  
AppSecret = '[**你的APPKEY**]'  
registerAccounts("xiangmax", AppID, AppSecret)  
w1 <- createweixin("xiangmax", ssl.verifypeer = F)  
u1 <- getUsers(w1, ssl.verifypeer = F)  
head(u1)
```

Rweixin

```
names(u1)  
head(u1)
```

```
[1] "nickname"      "sex"           "city"          "province"  
[5] "country"       "language"      "subscribe"     "subscribe_ti  
[9] "openid"        "unionid"      "remark"       "headingurl"
```

	sex	city	province	country	language	subscribe
1	2			百慕大	zh_CN	1
2	1	伦敦	英格兰	英国	zh_CN	1
3	1	西城	北京	中国	zh_CN	1
4	1	海淀	北京	中国	zh_CN	1
5	2			列支敦士登	zh_CN	1
6	2	休斯敦	德州	美国	zh_CN	1

Rweixin 其他函数

- getMaterialList 获取素材列表
- getMaterialNum 获取素材数量
- uploadImage 上传图片
- uploadNews 上传图文
- sendNews 群发信息

图文相关

图文消息(共5条)



新建图文素材



更新于 昨天 07:00

每日书单



最新一期书单2017-12-01



更新于 星期五 07:00

每日书单



最新一期书单2017-11-30



更新于 星期四 07:00

每日书单



最新一期书单2017-11-29



Where to go?

Where to go?

- 构建一个后台服务
- fiery运行效率
- 跨域访问
- 内网穿透/端口映射

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port
```

谢谢