

# APiculture

Vincent Goulet



# APIculture

**Vincent Goulet**

École d'actuariat, Université Laval

Insurance Data Science Conference  
Hannover, 9–10 June, 2026

© ⓘ ⓘ ⓘ © 2026 by Vincent Goulet. *APiculture* is licensed under a Creative Commons [Attribution-ShareAlike 4.0 International](#) license. You are free to :

- **Share** — copy and redistribute the material in any medium or format;
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms :



**Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.



**ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

**No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

#### Source code

[❤️ View on GitLab](#)

#### Cover

Male specimen of a digger bee (*Anthophora montana*) collected as part of a survey of Rocky Mountain National Park (Colorado). Anthophorini species are solitary and make nests in the soil. Digger bees are similar in size to a bumblebee, but the cream colored markings on their face are unique to them. Photo credit : public domain image provided by the *Bee Inventory and Monitoring Lab* of the *United States Geological Survey* in Beltsville (Maryland); via [Wikimedia Commons](#), where the image was a finalist in [Picture of the Year 2015](#).

# APIS AND THE FUTURE OF DATA SCIENCE

MARK SELLORS



# HIERARCHY OF DATA SCIENCE COMMUNICATION



## Programmation lettrée et R Markdown



## Rapports dynamiques avec Shiny



## Conception de paquetages R



## Programmation lettrée et R Markdown



## Rapports dynamiques avec Shiny



## Conception de paquetages R



## Utilisation et conception d'interfaces API



# Ligne de commande Unix



# APPLICATION PROGRAMMING INTERFACE 101

Set of functions that can be called over the network



# APPLICATION PROGRAMMING INTERFACE 101

Set of functions that can be called over the network

client request  
with operands



`https://example.com/api/add/2/2`



+



network  
addressable  
function

4



server response

## CONTENTS OF THE SHORT COURSE



use



build



deploy

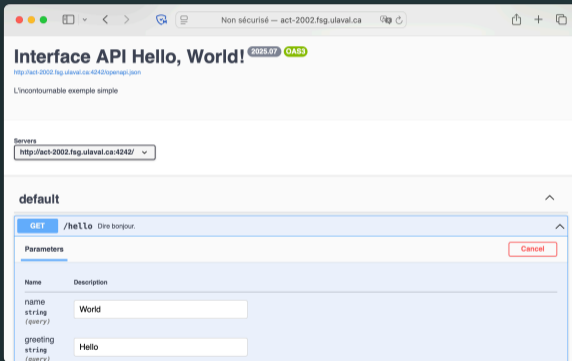
# USE

## curl

```
$ curl -s "http://act-2002.fsg.ulaval.ca:4242/hello"
"Hello, World!"%
$ curl -s "http://act-2002.fsg.ulaval.ca:4242/hello?name=Vincent"
"Hello, Vincent!"%
$ curl -u <ID> -s -X POST -H "X-Atlassian-Token: no-check" -F avatar=@<file> \
"https://projets.fsg.ulaval.ca/git/rest/api/1.0/users/<ID>/avatar.png"
```

# USE

## Swagger interface



The screenshot shows a web browser window displaying the Swagger API interface. The browser's address bar shows the URL `http://act-2002.fsg.ulaval.ca`. The page title is "Interface API Hello, World!" with a version number "2025.07" and an "OAS3" badge. Below the title, there is a subtitle "L'incontournable exemple simple".

The interface includes a "Servers" section with a dropdown menu showing the selected server: `http://act-2002.fsg.ulaval.ca:4242/`.

The main section is titled "default" and shows a list of API endpoints. The selected endpoint is `GET /hello` with the description "Dire bonjour.".

Below the endpoint information, there is a "Parameters" section with a "Cancel" button. The parameters are listed in a table:

Name	Description
name string (query)	<input type="text" value="World"/>
greeting string (query)	<input type="text" value="Hello"/>

# BUILD

api.R



```
## Plumber API

## @apiTitle Interface API Hello, World!
## @apiDescription L'incontournable exemple simple
## @apiVersion 2025.07

## Dire bonjour.
## @param name:string
## @param greeting:string
## @serializer unboxedJSON
## @get /hello

function(greeting = "Hello", name = "World")
  paste0(greeting, ", ", name, "!")
```

## DEPLOY LOCALLY



Run API



## DEPLOY LOCALLY

```
> pr_run(plumb("api.R"))
```

# DEPLOY LOCALLY

deploy-local.R

```
## Load required package
library(plumber)

## Deploy on localhost at automatic port
pr_run(plumb(file = "api.R"))
```

```
$ Rscript -e 'source("deploy-local.R")' &
[1] 18072
Running plumber API at http://127.0.0.1:15708
Running swagger Docs at http://127.0.0.1:15708/__docs__/
```

## DEPLOY ON A SERVER



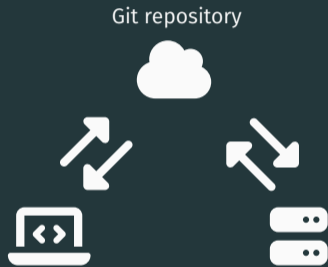
Linux (virtual) server  
behind university firewall

## DEPLOY ON A SERVER



command line only  
using ssh

## DEPLOY ON A SERVER



transfer of API source code via Git

## DEPLOY ON A SERVER

deploy.R

```
## Load required package
library(plumber)

## Deploy on server at fixed port
pr_run(plumb(file = "api.R"),
       host = "132.203.190.200", port = 4242)
```

```
$ Rscript -e 'source("deploy.R")' &
[1] 4175580
Running plumber API at http://132.203.190.200:4242
Running swagger Docs at http://132.203.190.200:4242/___docs___/
```

## DEPLOY ON A SERVER

deploy.R

```
## Load required package
library(plumber)

## Deploy on server at fixed port
pr_run(plumb(file = "api.R"),
       host = "132.203.190.200", port = 4242)
```

assigned port!  
↓

```
$ Rscript -e 'source("deploy.R")' &
[1] 4175580
Running plumber API at http://132.203.190.200:4242
Running swagger Docs at http://132.203.190.200:4242/__docs__/
```

## DEPLOY ON A SERVER

deploy.R

```
## Load required package
library(plumber)

## Deploy on server at fixed port
pr_run(plumb(file = "api.R"),
       host = "132.203.190.200", port = 4242)
```

assigned port!  
↓

```
$ Rscript -e 'source("deploy.R")' &
[1] 4175580
Running plumber API at http://132.203.190.200:4242
Running swagger Docs at http://132.203.190.200:4242/__docs__/
```



## DEPLOY ON A SERVER

```
$ nohup Rscript -e 'source("deploy.R")' &  
[1] 4176723  
Running plumber API at http://132.203.190.200:4242  
Running swagger Docs at http://132.203.190.200:4242/___docs___/  
$
```

## DEPLOY ON A SERVER

```
$ nohup Rscript -e 'source("deploy.R")' &
```

```
[1] 4176723
```

```
Running plumber API at http://132.203.190.200:4242
```

```
Running swagger Docs at http://132.203.190.200:4242/___docs___/
```

```
$ ps -cux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
vigou3	4141988	0.0	0.1	17324	9668	?	Ss	11:41	0:00	systemd
vigou3	4141989	0.0	0.0	198968	7664	?	S	11:41	0:00	(sd-pam)
vigou3	4142141	0.0	0.1	44672	9264	?	R	11:41	0:00	sshd
vigou3	4142142	0.0	0.0	18904	6844	pts/1	Ss	11:41	0:00	bash
vigou3	4176723	3.1	1.0	304780	87480	pts/1	Sl	12:45	0:00	R
vigou3	4176783	0.0	0.0	19588	3952	pts/1	R+	12:45	0:00	ps

```
$
```

## DEPLOY ON A SERVER

```
$ nohup Rscript -e 'source("deploy.R")' &
```

```
[1] 4176723
```

```
Running plumber API at http://132.203.190.200:4242
```

```
Running swagger Docs at http://132.203.190.200:4242/___docs___/
```

```
$ ps -cux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
vigou3	4141988	0.0	0.1	17324	9668	?	Ss	11:41	0:00	systemd
vigou3	4141989	0.0	0.0	198968	7664	?	S	11:41	0:00	(sd-pam)
vigou3	4142141	0.0	0.1	44672	9264	?	R	11:41	0:00	sshd
vigou3	4142142	0.0	0.0	18904	6844	pts/1	Ss	11:41	0:00	bash
vigou3	4176723	3.1	1.0	304780	87480	pts/1	Sl	12:45	0:00	R
vigou3	4176783	0.0	0.0	19588	3952	pts/1	R+	12:45	0:00	ps

```
$ kill 4176723
```

UPSIDE



integrative project

# INTERNAL CHALLENGES

Large number of concepts to introduce

REST   JSON   IP addresses   port  
HTTP response code   HTTP requests  
curl   ssh   background processes

## plumber-INDUCED CHALLENGES



documentation



unusual syntax



debugging

ALTERNATIVE?



# EXTERNAL CHALLENGE



AI

Many thanks to  
Alexandre Coutu

This document was typeset with the  $\text{\LaTeX}$  document preparation system using the **beamer** class and the Metropolis theme. The main text is in Fira Sans, the computer code in Fira Mono, and icons are from Font Awesome.