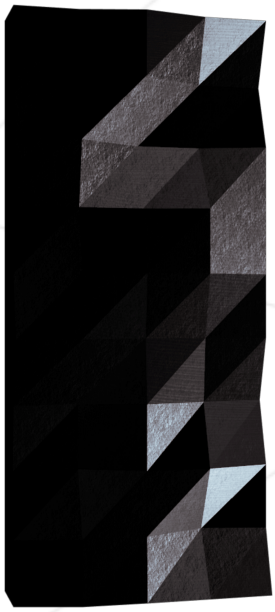

Ruby on Rails Training - Rush 01

Every companies want some

Summary: Here is the second (relatively) complex project you will have to achieve.

Version:

Contents

Chapter I

Consignes

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les jours de cette Piscine.

- Seul ce sujet sert de référence : ne vous fiez pas aux bruits de couloir.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices. L'url de votre dépot `GIT` pour cette journée est disponible sur votre intranet.
- Vos exercices seront évalués par vos camarades de Piscine.
- Vous ne devez laisser aucun autre fichier que ceux explicitement specifiés par les énoncés des exercices dans votre dépot de rendu.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Toutes les réponses à vos questions techniques se trouvent dans les `man` ou sur Internet.
- Pensez à discuter sur le forum Piscine de votre Intra et sur Slack !
- Lisez attentivement les exemples car ils peuvent vous permettre d'identifier un travail à réaliser qui n'est pas précisé dans le sujet à première vue.
- Réfléchissez. Par pitié, par Thor, par Odin !

Chapter II

Today's specific rules

-
- Besides the Rails Gems, you can only use the following Gems (including in the bonus part):
  - Pdf generation :
    - `Prawn`
    - `Wicked pdf`
    - `Pdfkit`
    - `Wkhtmltopdf`
  - Registration :
    - `Devise`
    - `OmniAuth`
    - `Authlogic`
  - Styling:
    - `Bootstrap-sass`
    - `Rails-bootstrap-forms`
  - Ajax:
    - `Best in place`
  - Rich text :
    - `wysiwyg-rails`
  - CSV:
    - `roo`
  - Charts:
    - `LazyHighCharts`
  - Date range:
    - `jquery datepicker`

Chapter III

Preamble

`Usefull Facts`

Google runs on 5000 times more code than the original space shuttle

The microcontroller inside a Macbook charger is about as powerful as the original Macintosh computer.

"The quick brown fox jumps over the lazy dog" is an English-language pangram, a phrase that contains all of the letters of the alphabet.

If you eat a teaspoon of sugar after eating something spicy, it will completely neutralize the heat.

A day on Venus is longer than a year on Venus.

The term “nerd” originated from Dr. Seuss’ 1950’s book, If I Ran The Zoo.”

Scotland’s national animal is the unicorn

A donkey will sink in quicksand but a mule won’t.

A woodchuck breathes only 10 times in hibernation.

A cat’s jaw cannot move sideways.

An ostrich’s eye is bigger than its brain.

If you put a drop of liquor on a scorpion, it will instantly go mad and sting itself to death.

The average person falls asleep in seven minutes.

The Declaration of Independence was written on hemp (marijuana) paper.

A group of kangaroos is called a mob.

IBM's motto is 'Think'. Apple later made their motto 'Think different'.

Albert Einstein and Charles Darwin both married their first cousins.

Charlie Chaplin once won third prize in a Charlie Chaplin look-alike contest.

Most dust particles in your house are made from dead skin!

In the "May I have a large container of coffea" sentence, the number of letters in each words makes the (begining of) the pi number.

It would take approximately 31.7 years to count off 1 billion seconds.

Bikinis and tampons were invented by men.

Kissing is more hygienic and healthier than shaking hands.

It is impossible for a human to sneeze with the eyes open.

Chapter IV

Mandatory part

Here is the second rush that will end this two weeks piscine.

Here is the opportunity to demonstrate your skills with Rails and prove your amazing efficiency as a startup-freelance-whatev-developer.

'Views' are not imposed. You will choose if you need one page or as many as you will need words. However, you will have to set a layout and a slight structure:

- `Incentive`: prompts the user to make specific actions.
- `Distinction`: groups similar type functions making access, memorisation and learning easier.
- `Readability`: highlights and limits the number of different elements featured on the interface.
- `Brevity`: displays content to make the data exploitation easier.

This is not a Rush rule, but a recommendation to use common sense and facilitate you evaluation.

You don't have to run tests. I mean it won't be part of the evaluation, but you're never immune to edge effects of a feature development.

However, you `must` have a seed that will be enough to demonstrate the proper operations of each functionality.

You should really read the WHOLE subject before your begin ;) .

IV.1 Iteration 1: Non commercial functionalities

First, you must create a basic project including several parts, all gathered in a sprint:

IV.1.1 Sprint1 : User accounts

You must set an authentication system with the following mandatory features:

- No clear password (in logs or DB)
- Sign-in // sign-out form
- Display:
  - Login
  - Events flash notification system
  - Users can modify their information

As well as a privilege management regarding the accounts:

- `admin`: has their own back-office with a total control on `all` the CRUD.
- `operator`: basic functionalities (accessing personal data) and must belong to the following groups (it can only belong to a single):
  - `Production`
  - `International`
  - `Commercial`
- `manager`: basic functionalities (accessing personal data), accessing back-offices of each sprints (except the admin's one), accessing personal data in read mode (all), and assigning users to groups.
- `unregistered`: accessing only the public parts.

`Reminder: You cannot use Gems that will work for you, like 'rolify', 'can-can'...`

IV.1.2 Sprint 2 : Admin back-office

As mentioned above, all the `CRUD` of the whole application is represented here. Everything you will develop along today's exercises will be added here in free access for administration use.

Only they can:

- assign/modify privileges to users.
- create/modify a 'user account' and delete one.

`The admin is almighty regarding the data, but they must not create errors: 'stored nil values', 'non uniques IDS' or 'invalid data'`

`Reminder: You cannot use 'rails-admin', 'Active-admin'... type Gems.`

IV.1.3 Sprint 3 : Window site

A homepage introducing you fake company. This part is public
You're responsible for this page. It `has` to include:

- links to: registration, login
- photos, one (or more) video(s)

Back-office :

- None

IV.1.4 Sprint 4: In mail

Do I have to explain the 'in mail' concept? Well, if ever, here is a list of mandatory features:

- `in box`: for all the registered users with all the in-mails destined to the logged user mentioning the status: 'read or new'. (VOIR L'ENONCE FRANÇAIS PAS TROP CLAIR JE TROUVE)
- `out box`: for all the registered users with all in-mails sent by the logged user mentioning 'read' if the recipient has opened it.
- `send`: everyone can send 'in mails' individually.
- `contact list`: list of all the registered users with links to "in mail this person" (I won't explain that) and mention of the group they belong to.
- `pdf`: sent/received in-mails have a button that generates a pdf with the mail content (going through the print function of the browser will not be tolerated... of course you knew).

Back-office :

- Sending in-mails to groups.
- Sending mails to all.
- activity history, send list with mention 'read' if opened by the recipient, like a log [date/time] [Title][Sender][Recepient] [Link to content]:
  - For all
  - For users
  - For groups
- Likewise, a manager can read all the in-mails via the history (not scary at all, come on) Mr Tuttle)

IV.2 Iteration 2: Commercial functionalities

After setting a basic infrastructure, you must now take care of the core of your fake company.

IV.2.1 Sprint 1: Client Database

The name says it all: it's a client list that will be prototyped as follows:

name, surname, mail, phone, company, function

It will be accessible to any registered user and sorted out by alphabetical order and company. You also must be able to import new ones via CSV thank to the roo Gem.

Back-office :

- export in CSV
- Activity history: by company and by client, we must see the actions regarding these entities. (It will be irrelevant with the current code, but it will make sense with next spring's features.) It will have to feature:
  - Imports, date and whom
  - Quotes, date and whom
  - Orders, date and whom
  - Invoices, date and whom
- company deletion
- assigning clients to operators (with all groups)

IV.2.2 Sprint 2: Project follow-up

A project a:

- a name, a client and one or several quotes (invoice, proposal made to a client by an operator)
- an order (order created by an operator at the client's request)
- one or several in-going invoices (operating cost invoice, goods)
- an out-going invoice (final client invoice matching the right order)

Each of these elements has:

- A header featuring:
  - a client
  - a company
  - their operator
  - a creation date
- Items that have:
  - a price
  - a description
- an intro in 'rich text'
- a sub-total in euro '€'

Operators can create invoices, orders, quotes that `must` belong to a project.

`I'll spare you the VAT. My pleasure.`

Back-office:
You will always calculate on a weekly granularity (abscissa axis) of the sums (ordinate axis) on selectable time ranges via the 'pick-a-date' plugin in the header. There are three pages: by `client`, by `company` and for `all`, that feature the following graphs:

- Some sums (out-going invoices) minus the expenses (in-going invoices)
- Amount of orders

Only a manager can create a project.

IV.2.3 Sprint 3: Survey

Each operator can create `one` survey. Surveys are public and you must register the surveyed users' mails (syntactically valid), that were not registered.

Surveys feature:

- A title
- An intro in 'rich-text'
- Yes/No questions
- A thank you message in 'rich- text'

Back-office :

- creation of multiple surveys
- survey validation (publication/visible)
- import of new surveys by CSV
- export of survey results by CSV
- collected mails listing and export by CSV

IV.3 Bonus

As you may have guessed, bonus will only be evaluated `only` if every mandatory element is perfectly achieved.
Many of the 'smart-features' have been omitted in the subject. Find relevant and useful elements to add as long as they don't use `any` additional Gem.

`ONE` bonus is an enhancement of `ONE` sprint and layout. This is serious business.