



Rush01

You're almost there!

Summary: Time to tackle a real project!

Contents

I	Ocaml piscine, general rules	2
II	Specific rules of the day	4
III	Preamble	5
IV	Mandatory part	6
IV.1	Basic project	6
IV.1.1	An HTTP configuration	6
IV.1.2	An authentication system	6
IV.1.3	A homepage	7
IV.1.4	Manage the user's profiles	7
IV.2	Choice of functionality	8
IV.2.1	Forum	8
IV.2.2	A messenger system	8
V	Optional part	10
V.0.1	Forum bonus: AJAX	10
V.0.2	Messenger bonus: using websockets	10
V.0.3	Push notifications	10
V.0.4	SSL	11
V.0.5	Package	11
V.0.6	Voting system	11

Chapter I

Ocaml piscine, general rules

- Every output goes to the standard output, and will be ended by a newline, unless specified otherwise.
- The imposed filenames must be followed to the letter, as well as class names, function names and method names, etc.
- Unless otherwise explicitly stated, the keywords `open`, `for` and `while` are forbidden. Their use will be flagged as cheating, no questions asked.
- Turn-in directories are `ex00/`, `ex01/`, ..., `exn/`.
- You must read the examples thoroughly. They can contain requirements that are not obvious in the exercise's description.
- Since you are allowed to use the OCaml syntaxes you learned about since the beginning of the piscine, you are not allowed to use any additional syntaxes, modules and libraries unless explicitly stated otherwise.
- The exercises must be done in order. The graduation will stop at the first failed exercise. Yes, the old school way.
- Read each exercise FULLY before starting it! Really, do it.
- The compiler to use is `ocamlopt`. When you are required to turn in a function, you must also include anything necessary to compile a full executable. That executable should display some tests that prove that you've done the exercise correctly.
- Remember that the special token `";;"` is only used to end an expression in the interpreter. Thus, it must never appear in any file you turn in. Regardless, the interpreter is a powerful ally, learn to use it at its best as soon as possible!
- The subject can be modified up to 4 hours before the final turn-in time.
- In case you're wondering, no coding style is enforced during the OCaml piscine. You can use any style you like, no restrictions. But remember that a code your peer-evaluator can't read is a code he or she can't grade. As usual, big functions are a weak style.
- You will NOT be graded by a program, unless explicitly stated in the subject. Therefore, you are given a certain amount of freedom in how you choose to do the

exercises. However, some piscine day might explicitly cancel this rule, and you will have to respect directions and outputs perfectly.

- Only the requested files must be turned in and thus present on the repository during the peer-evaluation.
- Even if the subject of an exercise is short, it's worth spending some time on it to be absolutely sure you understand what's expected of you, and that you did it in the best possible way.
- By Odin, by Thor! Use your brain!!!

Chapter II

Specific rules of the day

- Today, you can use the following packages (and the packages you'll design and install during the evaluation):

```
asgi-redis==0.14.1
asgiref==0.14.0
autobahn==0.16.0
channels==0.17.3
daphne==0.15.0
Django==1.9.5
django-bootstrap3==7.1.0
gunicorn==19.6.0
msgpack-python==0.4.8
Pillow==3.4.1
redis==2.10.5
six==1.10.0
Twisted==16.4.1
txaio==2.5.1
uWSGI==2.0.14
zope.interface==4.3.2
```

- You can also use JQuery and Nginx as you like.
- You're free to organize your project as you see fit.
- Don't forget your pip files dependencies.
- During the evaluation, the assessor will be able to modify the code on the browser side to try to defeat your repo. You must keep this in mind so you turn-in a very solid project.
- Your repo will be cloned with the `git clone <repo> ~/rush01` command. Make sure your project will work in this location.

Chapter III

Preamble

Here is a selection of Ig Nobel awarded in the last decade:

2013 Probability Award: To Bert Tolkamp, Marie Haskell, Fritha Langford, David Roberts and Colin Morgan, for discovering the probability that a cow gets up increases according to the time it lies down but it is almost impossible to tell when it lies down again.

2013 Peace Award: To Alexander Lukashenko, President of Belarus, for making public applause illegal and the Belarus police for arresting a one-armed man for doing so.

2013 Psychology Award: To Laurent Bègue, Brad Bushman, Oulmann Zerhouni, Baptiste Subra and Medhi Ourabah for experimentally confirming that the people believing they're drunk also believe they're attractive.

2013 Physics award : To Kiyoshi Mabuchi, Kensei Tanaka, Daichi Uchijima and Rina Sakai for studying the friction coefficient between the ground, a banana peel and a sole.

2014 Neuroscience Award : To Jiangang Liu, Jun Li, Lu Feng, Ling Li, Jie Tian and Kang Lee, for trying to understand how some people see the face of the Christ in the face of a toast.

2015 Physics Award : To Patricia Yang and David Hu for demonstrating the mammals weighting more then 3kg emptied their bladder in 21s (+/- 13s).

2015 Biology Award : To Bruno Grossi, Omar Larach, Mauricio Canals, Rodrigo A. Vásquez and José Iriarte-Díaz, for observing a chicken with a stick stuck to its rump probably walked like a non-avian dinosaur.

2015 Litterature Award : To Mark Dingemanse, Francisco Torreira and Nick J. Enfield for discovering the word "huh" existed everywhere in the world, but nobody can tell why.

Chapter IV

Mandatory part

Here is the second rush that will round off those two piscine weeks.

Here is the opportunity to show your Django skills with the creation of an intranet.

IV.1 Basic project

You must start creating a basic project with Django including several parts, amongst which:

IV.1.1 An HTTP configuration

A server configuration that will serve your project with Nginx (You must *NOT* use the development server, `DEBUG` *MUST* be set at `False`).

All the functionalities must work with this configuration. (If one functionality works with `./manage.py runserver` but not with Nginx, this functionality is not valid).

IV.1.2 An authentication system

The only available elements when disconnected will be:

- A registration page.

This page's form must feature a field for the user name and two fields for the password (that must not appear when typed). Validation will require that the username is not already used and entered values and password are identical.

Once the validation executed, a user is created with the data entered and validated. The visitor is connected with this user and redirected to the homepage.



Of course, the password must not be stocked in clear in the database.

- A connection page.

This page's form must feature a field for the user name and a field for the password (which must not appear when typed). Validation will require the user to be real, and its authentication to be possible.

Once the form is validated, the visitor is connected with this user and is redirected towards the homepage.

Once connected, the user must be redirected towards the homepage.

Once connected, you cannot access these pages anymore. The user can now access a disconnection button.

IV.1.3 A homepage

A homepage allowing to access all the functionalities that you will be able to implement.

IV.1.4 Manage the user's profiles

You can associate some informations to each user. For instance;

- Their name.
- Their surname.
- Their email.
- Their description.
- Their profile pictures.

Each user (and an administrator) must be able to check and update their information on their profile page.

The administrators must also be able to offer or cancel the administrator rights to a user (he must be able to know if a user has the administrator authorizations).

An administrator can also be a super user with a set of permissions or being part of a group. It's up to you.

IV.2 Choice of functionality

Once this project achieved, you will implement at least one either one of the following functionalities (if you do both, you'll get additional points).

IV.2.1 Forum

This forum, although basic, must implement a post system.

These posts must be listed by groups of 10 per page (design a pagination system when there are more than 10 messages).

Links towards the listed posts are present.

A post will have to feature a title, an author, a content, a creation date and time as well as a group of linked comments.

The comments will feature a content, a creation date and time and an author.

The detail of a post will display all informations of this post as well as the comments (and their respective informations).

Commenting a post or a comment will be possible (Your comments will be set as trees). The display will clearly indicate what the comment relates to (a post or a comment).

Any user must be able to create posts or comments.

IV.2.2 A messenger system

Your messenger system will allow the user to discuss with other users through direct messages.

A discussion will always be held between two *different* users (And these users are the only ones who can see the content of this discussion).

Your messenger system will include two pages:

- A page listing the *ongoing* discussions with they current user. You must show the informations of the last sent/received message (date/time, content cut down to 30 characters, contact...) for each discussion.

They will appear by groups of 10 and you will create a pagination system if there are too many messages for one page.

For each listed discussion, you will also implement a link to its details.

You will also add a link to start a discussion with a *different* user on the profile page (if the discussion has not been created yet).

- Detail on each discussion.

You will list all the messages chronologically along with their content and their emission's date and time.

The display should also clearly show if a message has been sent or received.

The user the discussion is shared with must also appear.

Chapter V

Optional part

Once the mandatory part has been achieved, you can choose one of the following bonuses:

V.0.1 Forum bonus: AJAX

implementation.

Reloading the page must not be necessary to display a comment form, fill it, send it and see the changes operate after the server validation and do it again.

Of course, this bonus is possible only if you have designed the forum.

V.0.2 Messenger bonus: using websockets

Using websockets in you messenger system:

The messages sent by another user must appear instantly in the discussion's thread without having to reload the page.

Of course, this bonus is possible if you have implemented the messenger system.

V.0.3 Push notifications

With the help of the websockets, you must implement push notifications system. Two occurrences can appear:

- You must have a notification appear when a new post is created if you have designed the forum. This notification must at least show the post title cut down to 30 characters.
- You must have a notification appear when a new message is sent to you if you have implemented the messenger system. You must at least display the sender's name and the message content cut down to 30 characters.

If you have validated both the messenger system and the forum, either one will be enough to validate this bonus.

V.0.4 SSL

Serve your site in HTTPS and use potential websockets in WSS with the help of some SSL key and certificate.

When your site is accessed with the HTTP protocol HTTP, the redirection towards the HTTPS version is automatic.

V.0.5 Package

Turn-in an application as a package that will be built and installed with `pip` during the evaluation.

V.0.6 Voting system

Implement a voting system that will allow your users to vote for some elements of your website.

These votes will be positive (upvotes) or negative (downvotes).

This functionality will unlock the display of specific informations:

- Vote buttons (to downvote or upvote the element).
- A vote recap (positive as well as negative) for each element.
- The current user's vote for each element (upvoted, downvoted, or no vote)

A user can only vote once for each element. They will not be able to upvote and downvote at once.

If the user clicks the same button twice, the previous vote is cancelled.

If the user clicks to downvote an element after upvoting it, the element is downvoted and the previous upvote is cancelled. It also works the other way.

Once the button has been clicked, the informations are updated on the display.

Once you have achieved your system, test it on your forum (On posts and comments). Make sure having several elements submitted to a vote on a similar page doesn't disrupt the different vote buttons and that each of them properly behaves (for instance, a single upvote for an element doesn't upvote all the page's elements).