Introduction

When you start a **path** at OpenClassrooms, the simplest way not to get lost is to follow the guideline, from project to project, until success!

When you start a **project** at OpenClassrooms, the simplest way not to get lost is to follow the required courses, from course to course, until success!

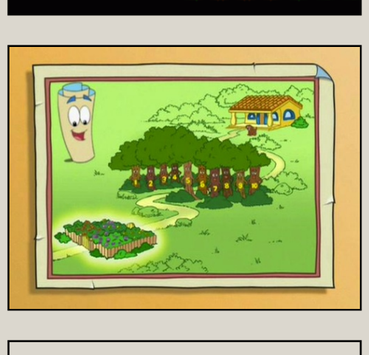
When you start a **course** at OpenClassrooms, the simplest way not to get lost is to follow the guide, chapter after chapter, until success!

However, **courses have prerequisites** : it is recommended to follow courses B,C and D before starting course A. And B,C,D require themselves several courses...

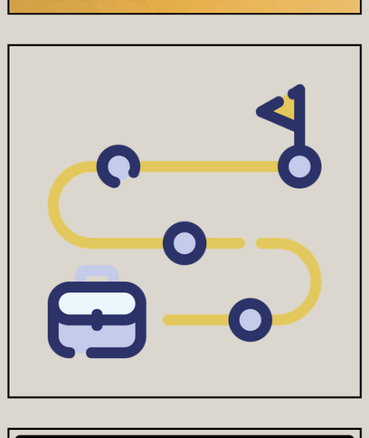
This is where things start to get tricky:

- Even by **reducing healthy learner curiosity** (avoid non-necessary references, skip light requirements), you will still need to keep track of all the required courses to achieve a given course.
- **Cross references, circular references** add some fun to your attempt to follow the links: a paper and a pen become bad companions for this journey
- Moreover, **Scheduling correcly your projects agenda** becomes a **time-consuming task** itself, with few time for that!

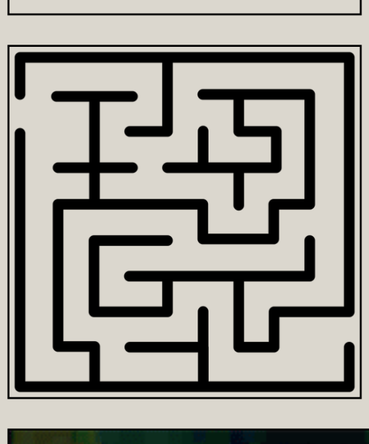
Imagine you start your journey in an unknown place. Somewhere in front of you a few tracks enter *the depths of Knowwood-OC* the forest-of-OpenClassrooms-knowledge, but you won't know where they will take you to!



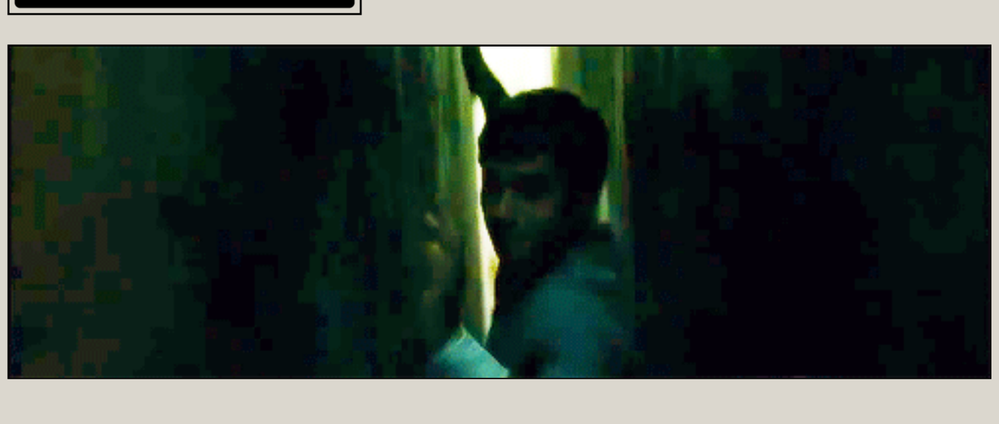
Perhaps it is a **purposeful choice** to let students find their ways by themselves, to let them discover some treasures along the road if curiosity and serendipity are in a good day, but... Even Dora the explorer has a map, hasn't she?



Perhaps the **complexity of paths and references between courses** has grown with time?
Perhaps a given uncertainty regarding the **student's preestablished knowledge of the courses requirements**, added to a given uncertainty regarding the **student's curiosity and available time "to read further"**, has discouraged the desire to build a "step-by-step" schedule for each path?
Would these sources of fuzziness (complexity and uncertainty) explain the **absence of a full path-project-course map** at OC?



When achieving my third project, I was used to find my way along the courses links, and well aware of the interest to self-discover goldnuts along the way.
I started project 4 with same curiosity, interest, will to learn, motivation as before, and began to take note of the prerequisites for this new project.
The cross references between courses, chapters, other courses, other chapters, overwhelmed me progressively with a feeling of being lost again. *As we will see soon below, this feeling was totally predictible and understandable.*

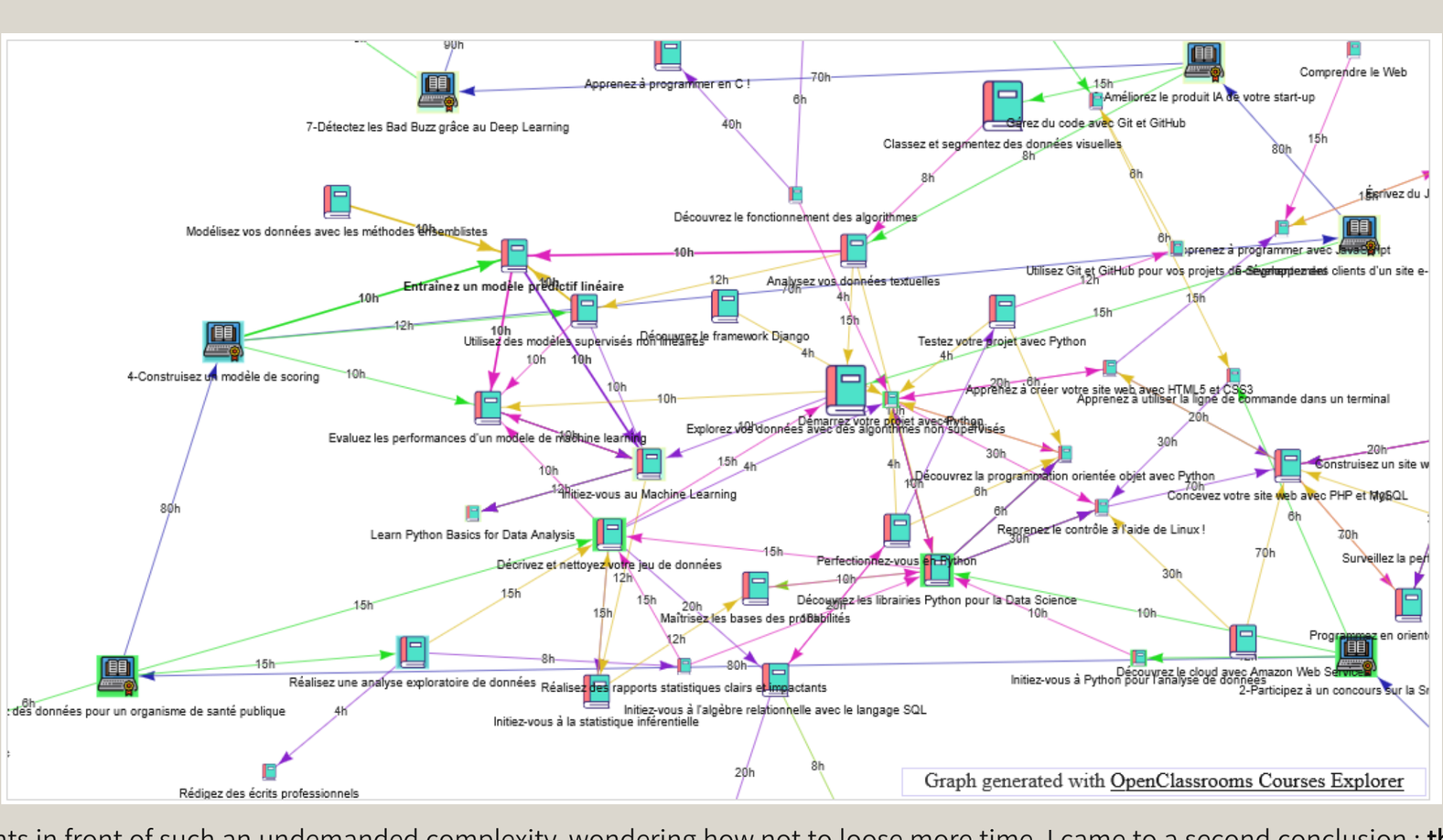


Moreover, the floor, the walls, the doors and windows of this maze seem to be moving slightly: following the OpenClassrooms courses structure refactoring, some courses names are changing, some courses are archived and replaced by other ones. Some of them are even melted in a more compound one.
This is certainly a good thing to update the courses and spring-cleanup the house now and again, but the cost for learners is to find their way in a courses structure that is **changing in time, with no trace of the past structure**. Another loss of precious time is when you wonder why your "achieved" courses are archived, and whether you have to follow or not the new updated courses with (probably better-fresher-more-recent content).

For instance with the best intentions, this is how OpenClassrooms becomes **OpenClassMaze** for the student fellow:

- The course Démarrez votre projet avec Python will be archived in july 2021, so we are invited to follow this one: Découvrez la programmation orienté objet avec Python
- But when landing on the page Découvrez la programmation orienté objet avec Python, we are invited to take note that the course has been updated into a new version that we can find here: Apprenez la programmation orienté objet avec Python
- Finally, reading this new course prerequisites, we are still invited to follow the course Démarrez votre projet avec Python **(The one that will be archived soon and replaced by a one that redirects to the current page? Yeah!)**.
- (re)Finally, reading further this new course prerequisites, we are also invited to follow two new courses Mettez en place votre environnement front-end (with Visual Studio) and Mettez en place votre environnement Python (with Pycharm).

After a few attempts to calm down this growing feeling, I came to a first conclusion : **paper, pen, post'it, calc-sheet, Jira, ... whatever tool you use to draw your learning roadmap is unuseful without a clear sight of the whole links and requirement-dependencies between courses!**.
To understand my mindset when I started following the courses of my 4th project, here is a screenshot of the dense relational network between courses of the 3rd and 4th projects in the path "Ingénieur IA":
In front of such a complex network, one can easily understand that a student can feel a bit lost, and feel a bit uncomfortable when following courses, chapter after chapter.



After very frustrating moments in front of such an undemanded complexity, wondering how not to loose more time, I came to a second conclusion : **this is a problem to be solved by a computer scientist, a data engineer, a motivated database architect, and a person who has 2 weeks in front of him!**.

Then, I looked around and found out I could do the job: I fortunately had achieved my 3rd project 2 weeks before the deadline, had some robust experience of databases, good knowledge about graphs traversal, and a strong need to practice Python programming (started 3 months ago) to strenghten my little knowledge of Pandas, Matplotlib and other tools.

There was a final question : is it possible to collect the courses requirements data without breaking any licence? Quick answer : **all OC courses are provided with a Creative Commons By-SA Licence**. Technical answer : All the data is here, availble and usable for my needs, so let's program an explorer and do some automated things with it.

That was the story , here is OpenClassrooms Courses Explorer!

What is OC-C-E ?

- OC-C-E is a tribute to OC : this is the cost I pay to get a full overview of OC courses interconnexions.
- OC-C-E is again a tribute to OC : this is a work motivated by the will to loose less time and embrace as much **good OC Courses content** as possible to avoid missing goldnuts somewhere
- OC-C-E is a submarine project between project 3 and project 4: I used all python data science programming I had learned the 3 last months
- OC-C-E is an OpenClassrooms Courses Explorer

What is *NOT* OC-C-E ?

- OC-C-E is **NOT** a work of OpenClassrooms, it was not demanded by anyone in OC Organization. For this reason, do not ask OC any support or information about this tool.
- OC-C-E is **NOT** a commercial industrial tool : it is free, it is "as is", and there is a licence to explain how to use it.
- OC-C-E is **NOT** for me myself and I (anymore). It was first motivated by my own needs, but the product belongs now to the data-python-scientists and other OC students who will want to do something with it..

Where are the OC-C-E entry points ?

- Launch with Jupyter Lab : several notebooks are provided, one with the full application and some others with tutorial step by step examples.
  - Full application :
    ```
    jupyter lab OCCoursesExplorer.ipynb
    ```
  - Step by step exploration with the datasets:
    ```
    jupyter lab OCCoursesExplorer_stepbystep.ipynb
    ```
  - Courses and paths network exploration:
    ```
    jupyter lab OCCoursesExplorer_courses_and_paths.ipynb
    ```
  - Courses scheduler exploration:
    ```
    jupyter lab OCCoursesExplorer_schedule.ipynb
    ```
- Launch with Jupyter Notebook: same as above, with `notebook` instead of `lab`
- Launch with Voilà: the main full application will start with Voilà!
  ```
  voila --enable_nbextensions=True --VoilaConfiguration.file_whitelist="['.*\.(csv|html|png|jpg)', 'viz.*', 'data.*']" OCCoursesExplorer.ipynb
  ```

See below for installation instructions

Why OC-C-E ?

The personal story leading to develop OpenClassrooms Courses Explorer is now 3 weeks behind. It took me way more time than expected to finally achieve the scheduling tool. There is no "in-between" approach : between building a bunch of scripts and building a clean application with a documented code, I chose the second one. Moreover I wanted to practice Pandas, Matplotlib, Voila, web scrapping... Now it is time to deliver the job and let you do something with it.

Who can use OC-C-E ?

- Someone with Python + Jupyter understanding
- Someone who knows someone who does that

For now, I do not plan to make it a standalone "one-click-install" application.

What can I really do with OC-C-E ?

- Use the datasets provided with the application
- Refresh the datasets from time to time
- Explore Topics and Paths
- Explore Paths and Projects
- Explore Courses and Chapters
- Explore Skills (this will be delivered in a later version)
- Schedule your path and take your progression in account
- Plenty things I do not even know, that you will imagine with the datasets.

OpenClassrooms Courses Explorer : How To install OC-C-E

Download or clone the git repository

Download or clone the git repository in a local folder where you will setup the python environment.

```
git clone https://github.com/TristanV/OCCoursesExplorer.git
```

Requirements

- Install all packages listed in the file requirements.txt + Jupyter Lab and Jupyter Notebook with widgetsnbextension.
  - If running **Anaconda** : create the OCCoursesExplorer environment with the environment.yml file :
    ```
    conda env create -n OCCoursesExplorer -f environment.yml
    ```
  - If running python **VirtualEnv** : use the pip_requirements.txt file
- Alternatively, the following manual setup works with **conda** from the folder where you have downloaded the git content:
  ```
  conda create -n OCCoursesExplorer python
  conda activate OCCoursesExplorer
  conda install -c conda-forge jupyterlab
  conda install -c conda-forge notebook
  conda install -c conda-forge voila
  conda install -c conda-forge IPython
  conda install -c conda-forge ipywidgets
  jupyter nbextension enable --py widgetsnbextension
  conda install -c conda-forge nodejs
  jupyter labextension install @jupyter-widgets/jupyterlab-manager
  jupyter labextension install @jupyterlab/toc
  conda install -c conda-forge matplotlib
  conda install -c conda-forge numpy
  conda install -c conda-forge networkx
  conda install -c conda-forge pandas
  conda install -c conda-forge pip
  conda install -c conda-forge plotly
  conda install -c conda-forge pyvis
  conda install -c conda-forge requests
  conda install -c conda-forge scipy
  conda install -c conda-forge scikit-learn
  conda install -c conda-forge selenium
  conda install -c conda-forge seaborn
  ```
- To use the notebooks with Jupyter and Jupyter Lab, be sure to have setup the widgets as nbextension.
  - By running the following commands, you should see `widgetsnbextension` and `jupyter-widgets` extensions enabled:
    ```
    jupyter nbextension list
    notebook section
    plotlywidget/extension enabled
    - Validating: ok
    voila-gridstack/extension enabled
    - Validating: ok
    voila/extension enabled
    - Validating: ok
    jupyter-js-widgets/extension enabled
    - Validating: ok
    jupyter labextension list
    @jupyter-widgets/jupyterlab-manager v2.0.0 enabled ok
    ```
    If you don't see these extensions enabled, please run the following commands:
  - For Jupyter Notebook :
    ```
    jupyter nbextension enable --py widgetsnbextension
    ```
  - For Jupyter Lab :
    ```
    jupyter labextension install @jupyter-widgets/jupyterlab-manager
    ```
- The **web scrapper** used by this app is Selenium, with a Firefox Driver.
  For convenience, a copy of the geckodriver.exe file is included in this repository, in the "driver" folder, but if you dislike exe files falling from unknown Github repositories, you may also download it manually from here : Last release of the Gecko (Firefox) Driver .
  If you don't want to use Firefox and prefer Chrome, Opera, Safari or whatever else, then you'll have to download the corresponding Web Driver into the driver/ folder. Here is a list of possible web drivers for Selenium
  However, please note that in this early stage of development, the OCCoursesExplorer application is not programmed (yet) to work with another web driver than Firefox. So if you really want to avoid Firefox and prefer your own web driver, you'll also have to replace a piece of Python code in the OCCoursesScrapping.py file.

Configuration

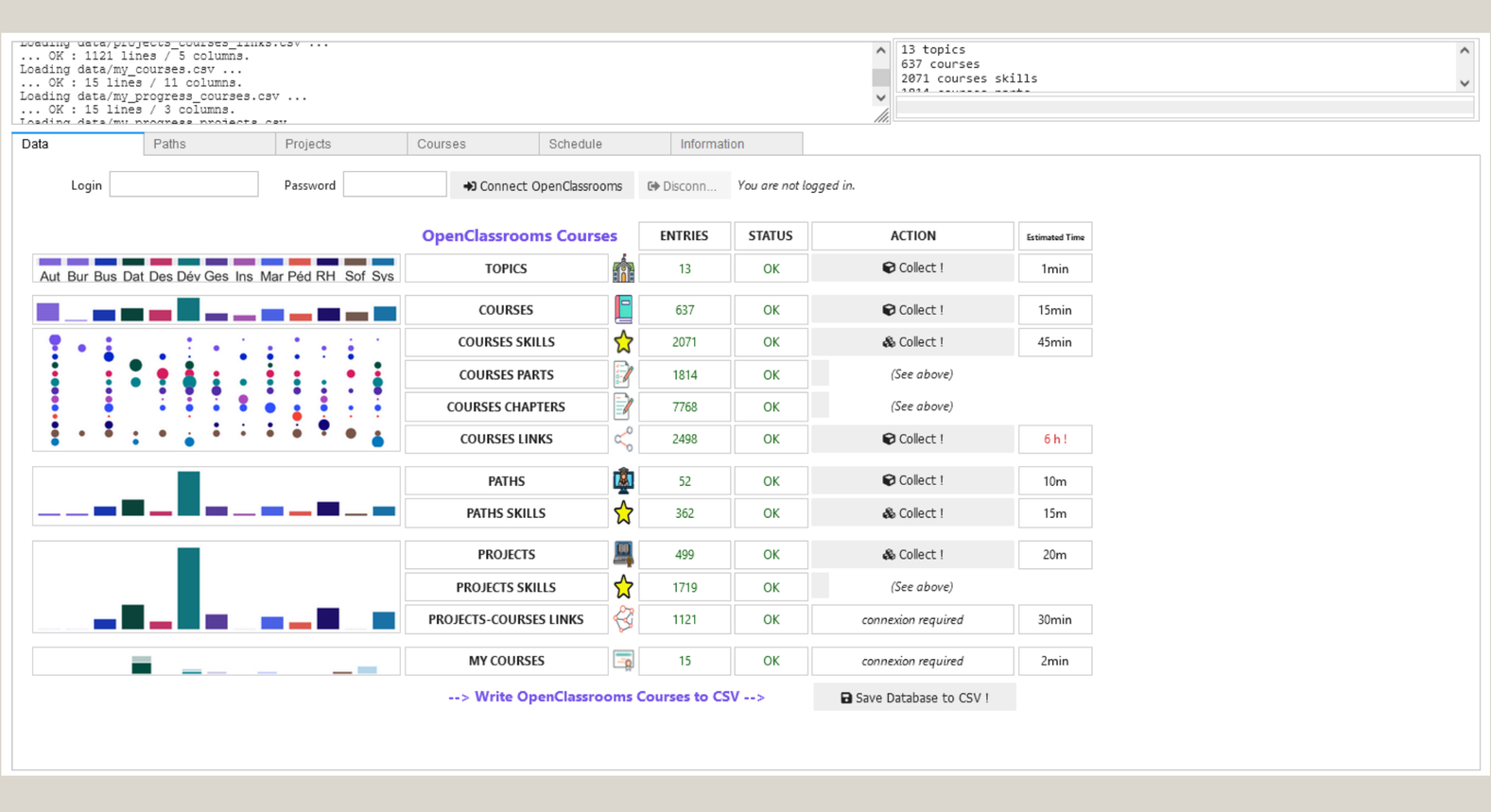
Once installed, there are few manual configuration steps.

- The web scrapper used by this app is Selenium, with a Firefox Driver.
  Selenium needs the **absolute path** towards this Firefox Driver to be set: open `OCCoursesConfig.py` in a text editor, then find the line where the variable `FirefoxWebdriverPath` is set, and change the String to correspond with the absolute path towards your Firefox web driver!
  ```
  FirefoxWebdriverPath = r"C:\path\towards\OCCoursesExplorer\driver\geckodriver.exe" # the path to the webdriver must be absolute
  ```

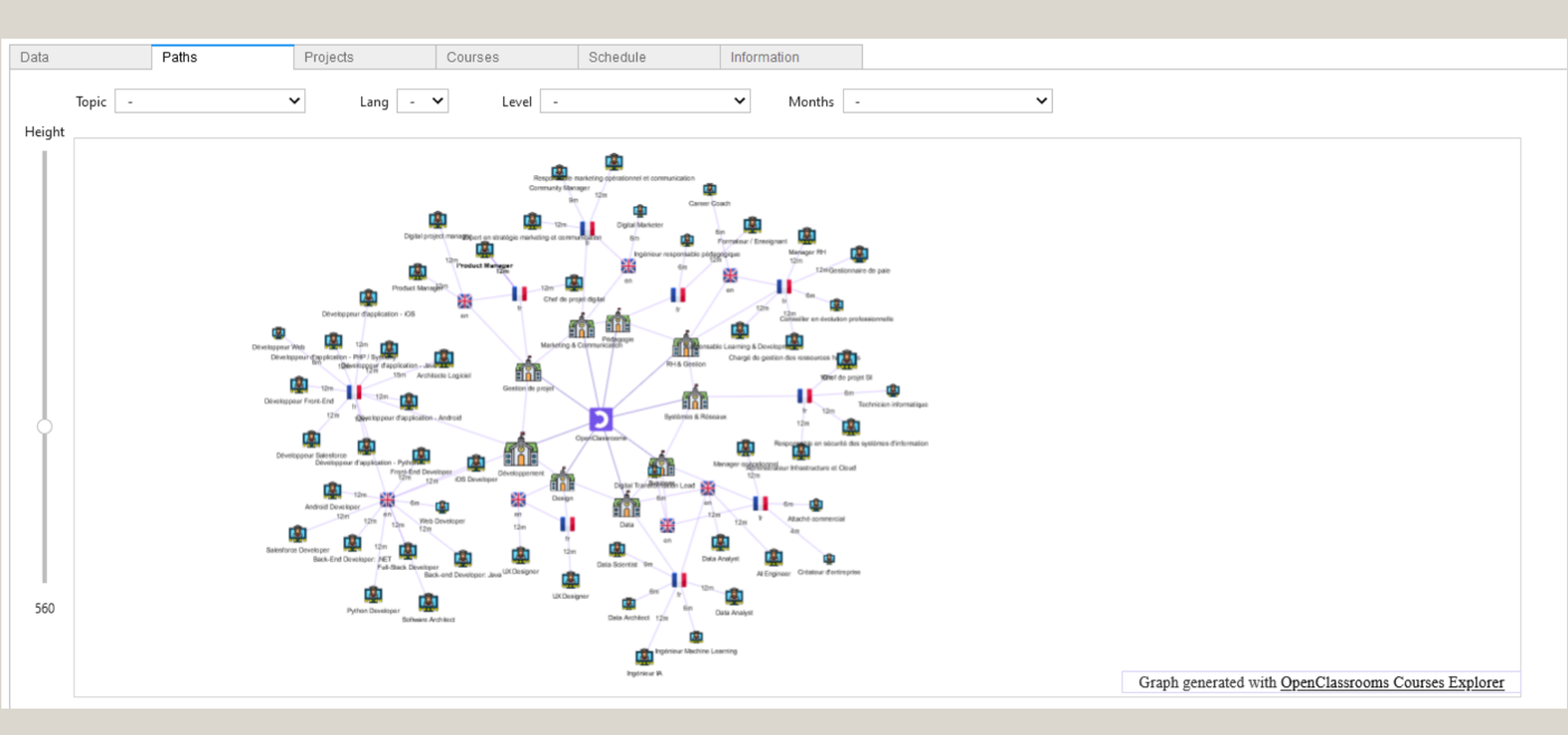
How to explore Topics, Paths, Projects and Courses

Here are some captures of the application. The widgets are self-explanatory.

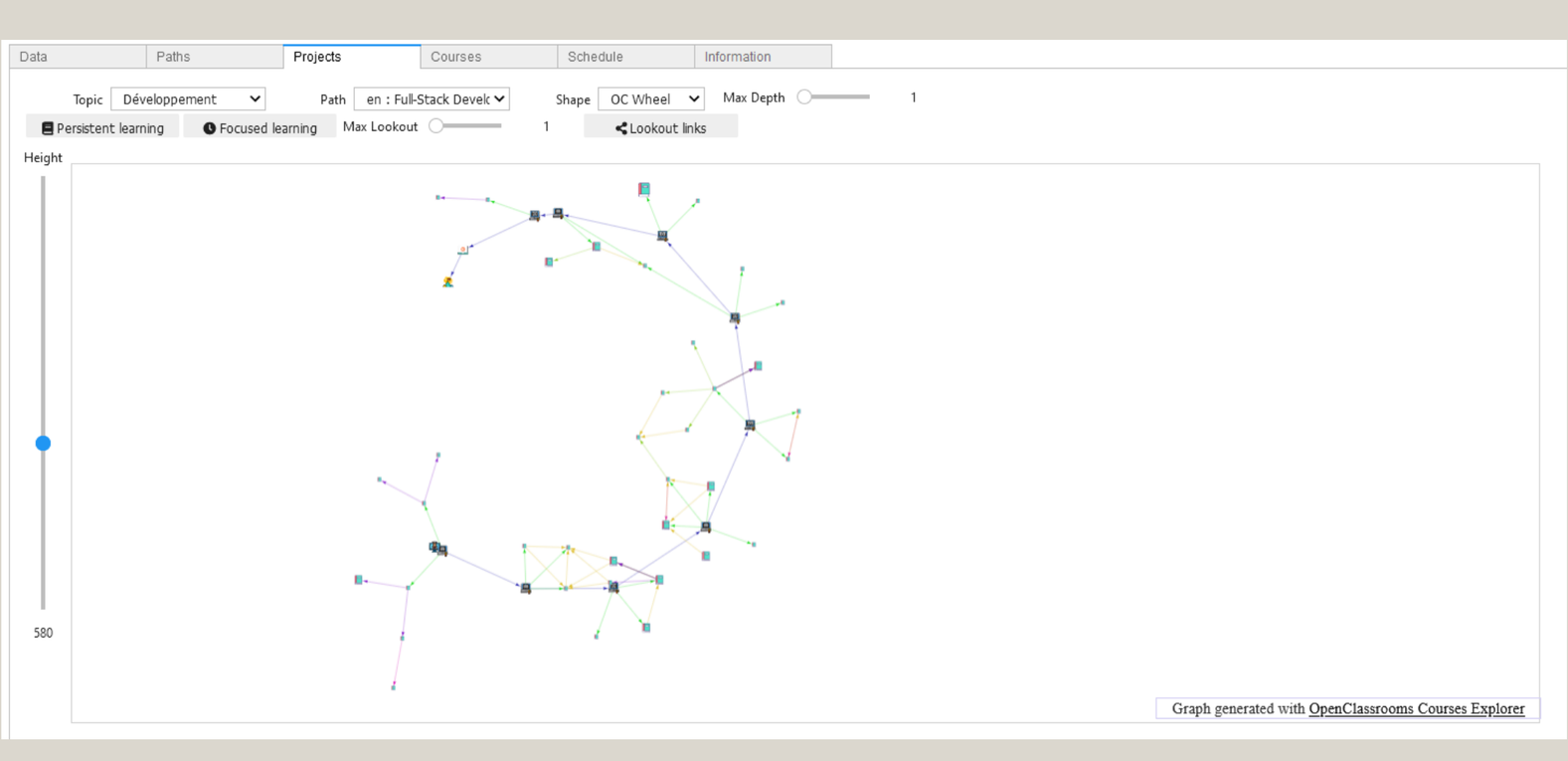
The Data Interface



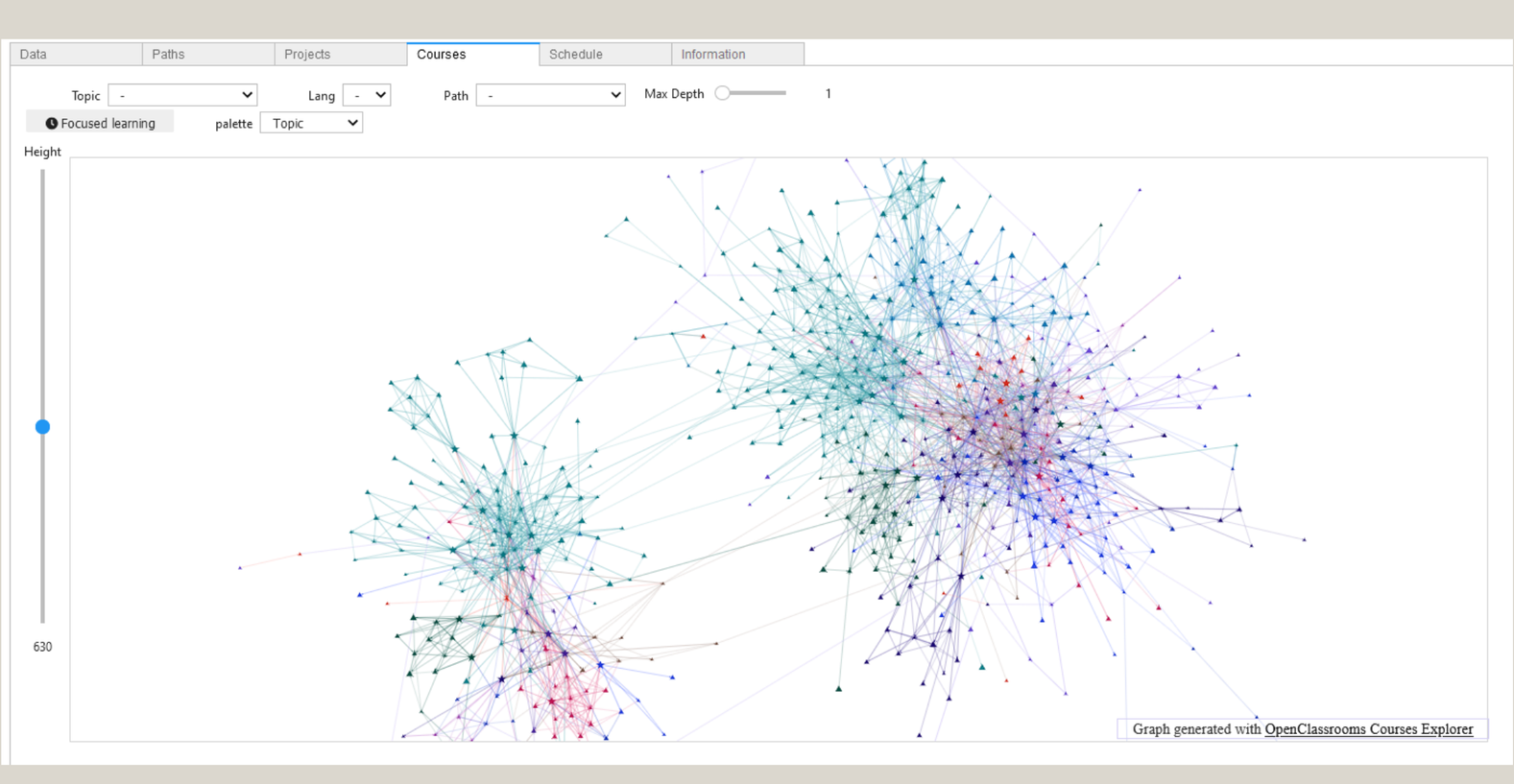
The Paths tab



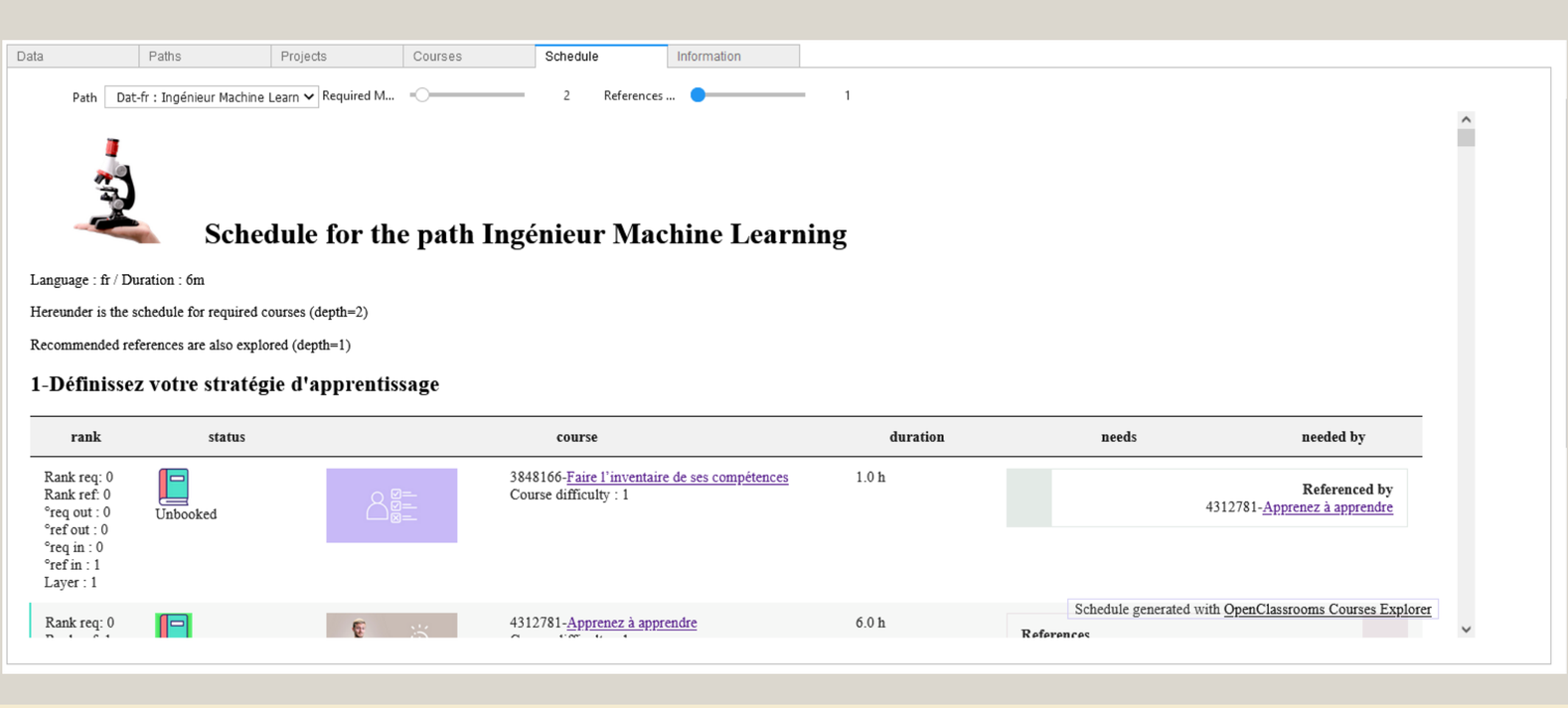
The Projects tab



The Courses tab



The Schedule tab

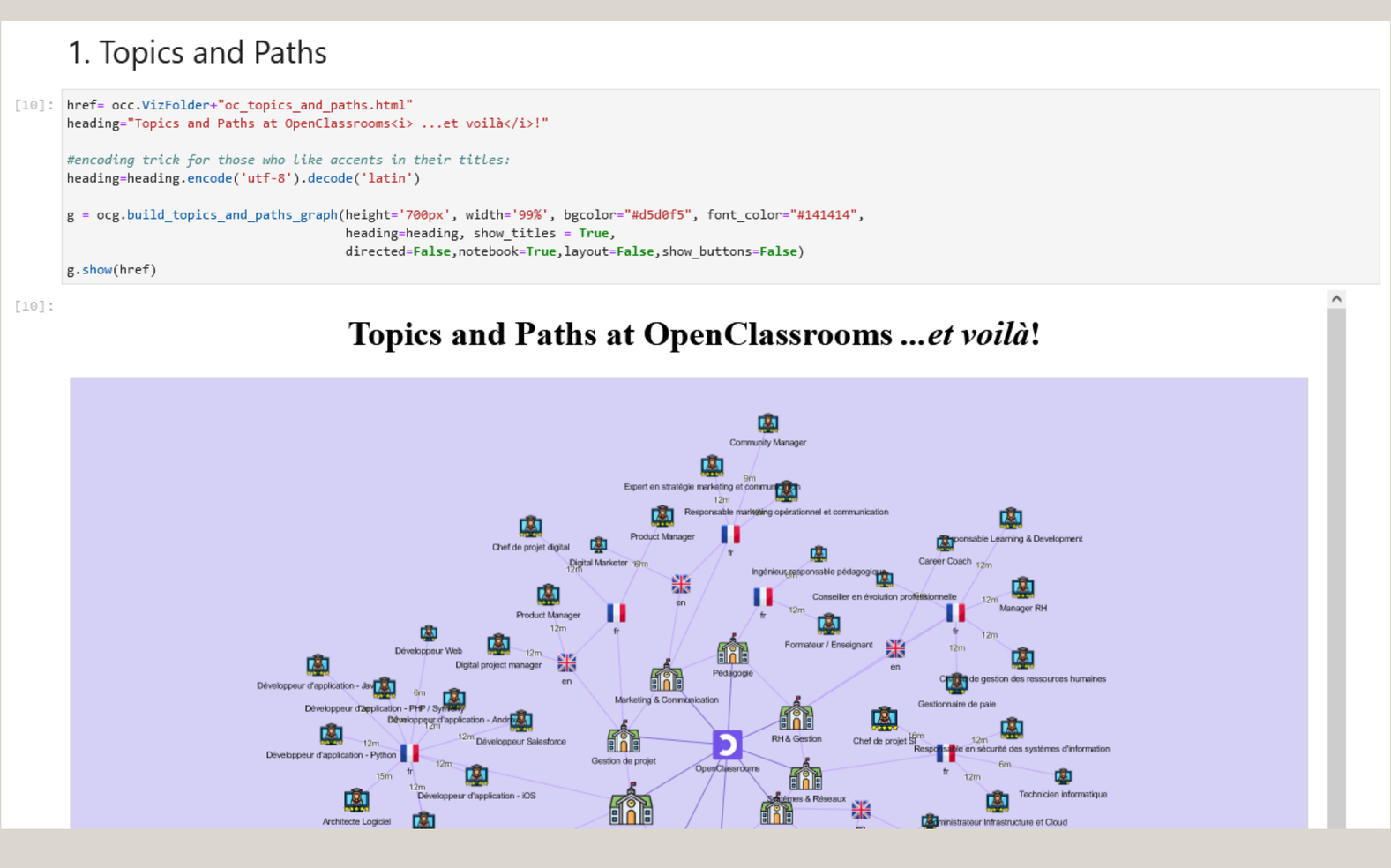


How to use OC-C-E with Notebooks?

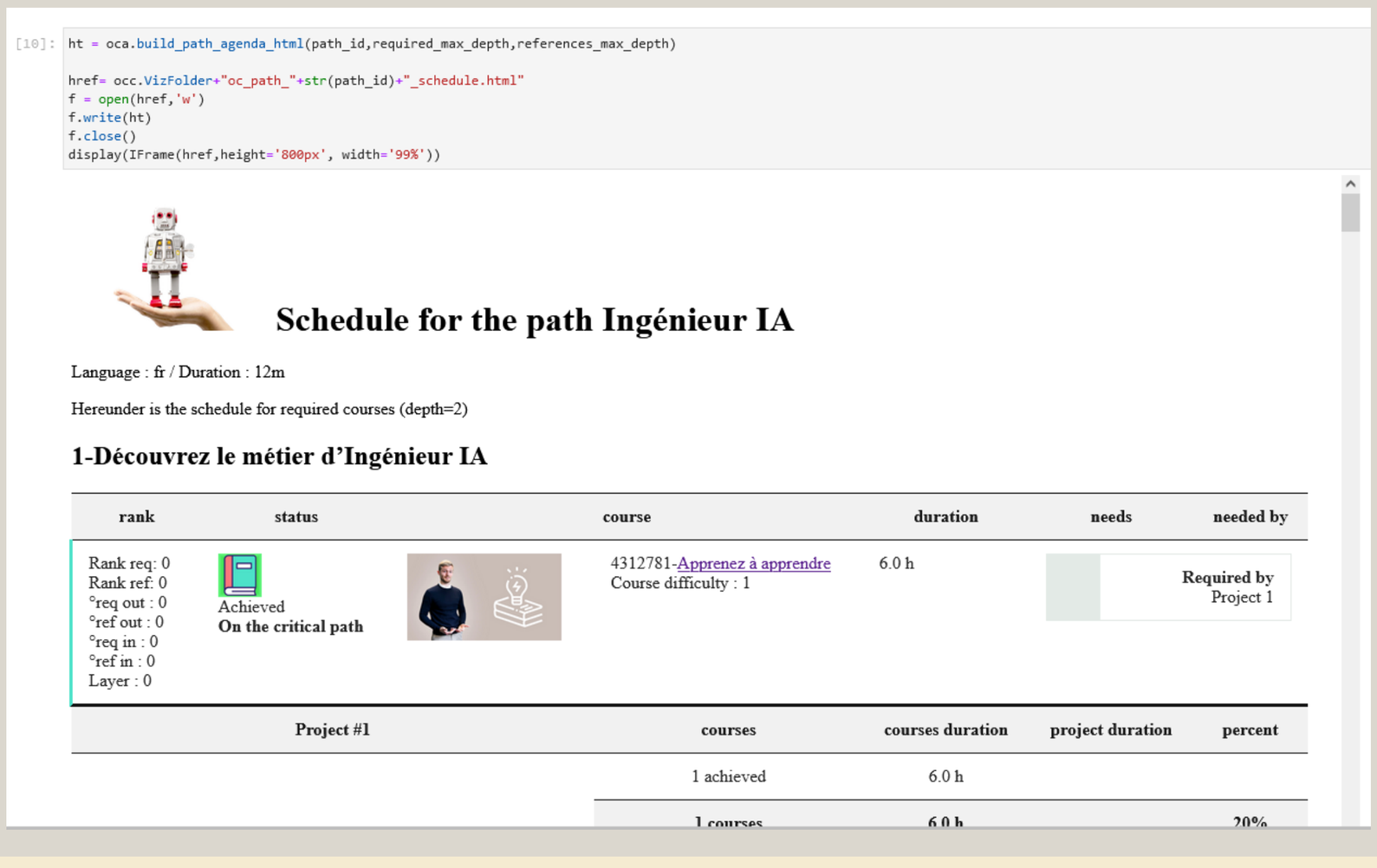
Feel free to use and adapt the tutorial notebooks provided : step by step, courses and paths, scheduler ...

Launch some queries on the datasets

Create some dynamic graphs



Create a schedule



How to analyze my Courses progression?

In the "my_courses" folder, there is a notebook analyzing your progression: it is largely using an exercise correction given in a Course about exploratory analysis.
Here is the course analyzing OC courses progression.
Launch the notebook with the command:

```
jupyter lab my_courses/my_courses_analysis.ipynb
```

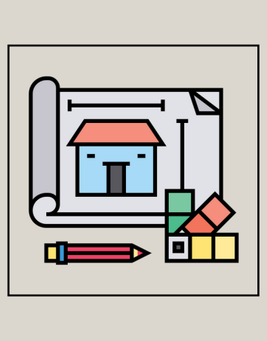
How to export my Path Schedule towards CSV or Jira?

In a next version of the application.

How to contribute?

People interested, mostly OC students following the DATA paths but also "Python dev" and any other curious dev, are welcome to contribute: contact me via the OC Workplace!

Checkpoint



Remember all of this is just a Proof of Concept, a Work in Progress. I will be very happy if someone gets something useful from this tool. Please let me know!