Travail effectue pour Neurosys

1. Get the local developpement environment working(PyCharm, Visual Studio Code)
   1. The procedure to get labguru working locally can be found in <https://github.com/BioData/LabguruPython>.
   2. In addition to make this procedure working, it is also necessary to change one python file in order to get everything up and running. Running the install procedure as described on Github will raise one problem: your local install will point on **my.labguru** (which not work for us) and you must point on **eu.labguru**.
   3. Once you carry on the install procedure: Edit the file Labguru/labguru/api.py and replace **my.labguru.com** by **eu.labguru.com**.
   4. We provided labguru with what was requested to make local dev working on: MacOS, Windows10 and Windows11, at <https://app.slack.com/client/TA2H35MB3/C0293J133LH> in python section.
2. Prise en main du Logiciel Labguru

Les besoins de Neurosys avec LabGuru

1. Dupliquer les experiences avec les stocks. La duplication doit avoir lieu à la fin d’une experience.
   1. L’unite n’est pas l’experience mais un ensemble d’experiences ou projets
2. Solution Prepare
   1. Acces dans LabGuru : Inventory
      * SP.Bioch
      * SP.Culture
   2. Neurosys souhaite nome la solution prepare avec le nom de tous les constituants de l’experience.
3. Dates de peramption
   1. Acces dans LabGuru : Inventory
      * SP.cult met automatiquement 1 date d’expiration 2 semaines apres la date de creation.
      * Est-il possible de laisser la possibilite de customiser la date d’expiration
4. Dans inventory shopping list
   1. Groupper par fournisseur et par jour
   2. Is it possible to have an Approved order for all folders?
   3. Order Price per supplyer

Meeting Labguru 19/04/2023

1. Present:
   * Tom, Julien Palleau
   * Amir Yampel
2. Questions
   * NEUROSYS QUESTONS:

I’d like to understand in what extent labGuru is customizable. Neurosys requests:

* + 1. **In Storage/Stock**: Stock duplication when you clone an experience. The way Neurosys is working: They work at project level and all the stock is made at the project level, when you duplicate an experience the stock at the experience level must remain unchanged however at the project level it must be updated.
    2. **In Inventory/SP**: Neurosys is looking for a way to rename the SP (Prepared Solution) with what is in the SP.
    3. **Always in Iventory/SP**: Neurosys is looking for automatically adding an expiration date with a default value of 2 weeks after the creation date and the possibility to freely modify the expiration date.
    4. **In Inventory/Shopping List**:
       1. Possibility to group by: provider and by day, the shopping list.
          1. Possibility to sum the order by provider
          2. Approved order for the all folder.
    5. **In E-Notebook/Projects:** 
       1. Neurosys is looking for elligable tasks only
  + How to clean the logs in flows, it is possible to archive them but how to delete it?
  + Is there a documentation with a description of all the objects available as well as their methods. The link <https://github.com/BioData/LabguruPython> is a begining but it doesn’t seem to be 100% accurate.
  + Setting up a dev environment to work from linux.

Les premiers scripts pythons

*# ##################################*

*# # List all the modules installed #*

*# ##################################*

*# help('modules')*

*# #-------------------------------------------------------------------------------------*

*# # List folders id and then print each folder by its id: OK*

*# for index, value in enumerate(lab.list\_folders(project\_id=None, page\_num=1)):*

*# print(value)*

*# # print(value.project\_id)*

*# # print(lab.get\_folder(folder\_id = value.project\_id).title)*

*# #-------------------------------------------------------------------------------------*

*# # List Inventory items name and id: OK*

*# for index, value in enumerate(lab.list\_inventory\_items(item\_type='cell\_lines', page\_num=1)):*

*# print(f"item name: {lab.list\_inventory\_items(item\_type='cell\_lines', page\_num=1)[index].name}")*

*# print(f"item id: {lab.list\_inventory\_items(item\_type='cell\_lines', page\_num=1)[index].id}")*

*# #-------------------------------------------------------------------------------------*

*# # Number of Inventory items: OK*

*# print(f"number items in inventory: {len(lab.list\_inventory\_items(item\_type='cell\_lines', page\_num=1))}")*

*# #-------------------------------------------------------------------------------------*

*# # List all projects id, projects name and for each one, all folders title: OK*

*# for proj in lab.list\_projects(page\_num=1):*

*# print(f'PROJECT - project id: {proj.id}, project name: {proj.name}')*

*# for folder in lab.list\_folders(project\_id=proj.id, page\_num=1):*

*# print(f'Folder title: {folder.title}')*

*# Look for what is get\_future\_folders() and what is get\_past\_folders() ???*

*# if proj.id:*

*# print(f'project members: {proj.member}')*

*# # # print(proj.register('New Folder 2'))*

*# for f in proj.get\_folder(folder\_id)*

*# # print('current folders')*

*# # for f in proj.get\_current\_folders():*

*# # print(f)*

*# # print('future folders')*

*# # for f in proj.get\_future\_folders():*

*# # print(f)*

*# # print('pass folders')*

*# # for f in proj.get\_past\_folders():*

*# # print(f)*

*# #-------------------------------------------------------------------------------------*

*# # List in all experiments all elements: OK*

*# for exp in lab.list\_experiments(page\_num=1):*

*# print(exp.id)*

*# for index, value in enumerate(lab.get\_elements\_by\_type(experiment\_id=exp.id, element\_type=None)):*

*# print(index, value)*

*# #-------------------------------------------------------------------------------------*

*# # List all sections: OK*

*# # The script doesn't return anything. Is there any section created?*

*# sections = lab.list\_experiment\_procedures(experiment\_id=None, page\_num=1)*

*# print(sections)*

*# #-------------------------------------------------------------------------------------*

*# # List all projects: OK*

*# # print out project info*

*# for project in lab.list\_projects(page\_num=1):*

*# print(f'Project id: {project.id}, Project title: {project.title}, Project owner: {project.owner}')*

*# #-------------------------------------------------------------------------------------*

*# # Download project information*

*# project = lab.get\_project(project\_id='1')*

*# print(f'Project id: {project.id}, Project title: {project.title}')*

*# #-------------------------------------------------------------------------------------*

*# # Find a project by name: NOK it returns an empty string ??? Must be investigated*

*# for project in lab.list\_projects(page\_num=1):*

*# projects = lab.find\_projects(name=project.title)*

*# print(projects)*

*# #-------------------------------------------------------------------------------------*

*# # Start new project*

*# project\_new = lab.add\_project(title="My new test project - JP", description="Dev test project")*

*# print(project\_new)*

*# #-------------------------------------------------------------------------------------*

*# Update a project*

*# for project in lab.list\_projects(page\_num=1):*

*# print(f'Project id: {project.id}, Project title: {project.title}, Project owner: {project.owner}')*

*project\_update = lab.update\_project(project\_id='7', title="Update my new test project - JP")*

*print(project\_update.id, project\_update.title)*

*# #-------------------------------------------------------------------------------------*

*# # List all experiments: OK*

*# for exp in lab.list\_experiments(page\_num=1):*

*# print(f'experiment id: {exp.id}, experiment name: {exp.title}, experiment description: {exp.description}')*

*# #-------------------------------------------------------------------------------------*

*# # Firt test with usual "hello world!": OK*

*# #log("Hello world this is a log!")*

*# #print("Hello World this is a print!")*

Labguru Software Understanding

Folders

Experiments

Project(Internal project or customer)

1. Glossary
   1. Inventaire -> catalogue
   2. Item -> 1 element du catalogue (matiere non physique)
   3. Stock -> c’est physique, Information sur chaque copie (ex : P30->Lucie, P30->Tom, P30->Eliott, P30->Julien)
      1. Available to use
      2. Consumed

Dans mon catalogue(inventaire) j’ai une liste d’item (elements du catalogue). Chez Labguru le stock c’est la quantite pour 1 item (ou stock par item). Dans le catalogue j’ai 1 item P30 et 4 stocks (P30, Lucie, Tom, Eliott, Julien) + 1 emplacement physique pour chacun.