TP2

For today’s lab, the objectives are the following:

* Be able to see logs in both the frontend and backend of your app
* Manipulate cookies of your browser
* Test Oauth mechanism

# Test the python logger

1. Custom you Flask application by adding a new page **/logger** that does the following
   1. Prints a log on python
   2. Prints a log on the browser
2. Deploy your app
3. Activate logger on **Deta**
4. Go to your website and check on the browser console if you have any log
5. Go to Deta micro page and check if you can see your python log
6. Commit your changes in github
7. Check that the deployment went well by going on your application
8. Modify your python to be able to print a message in textbox

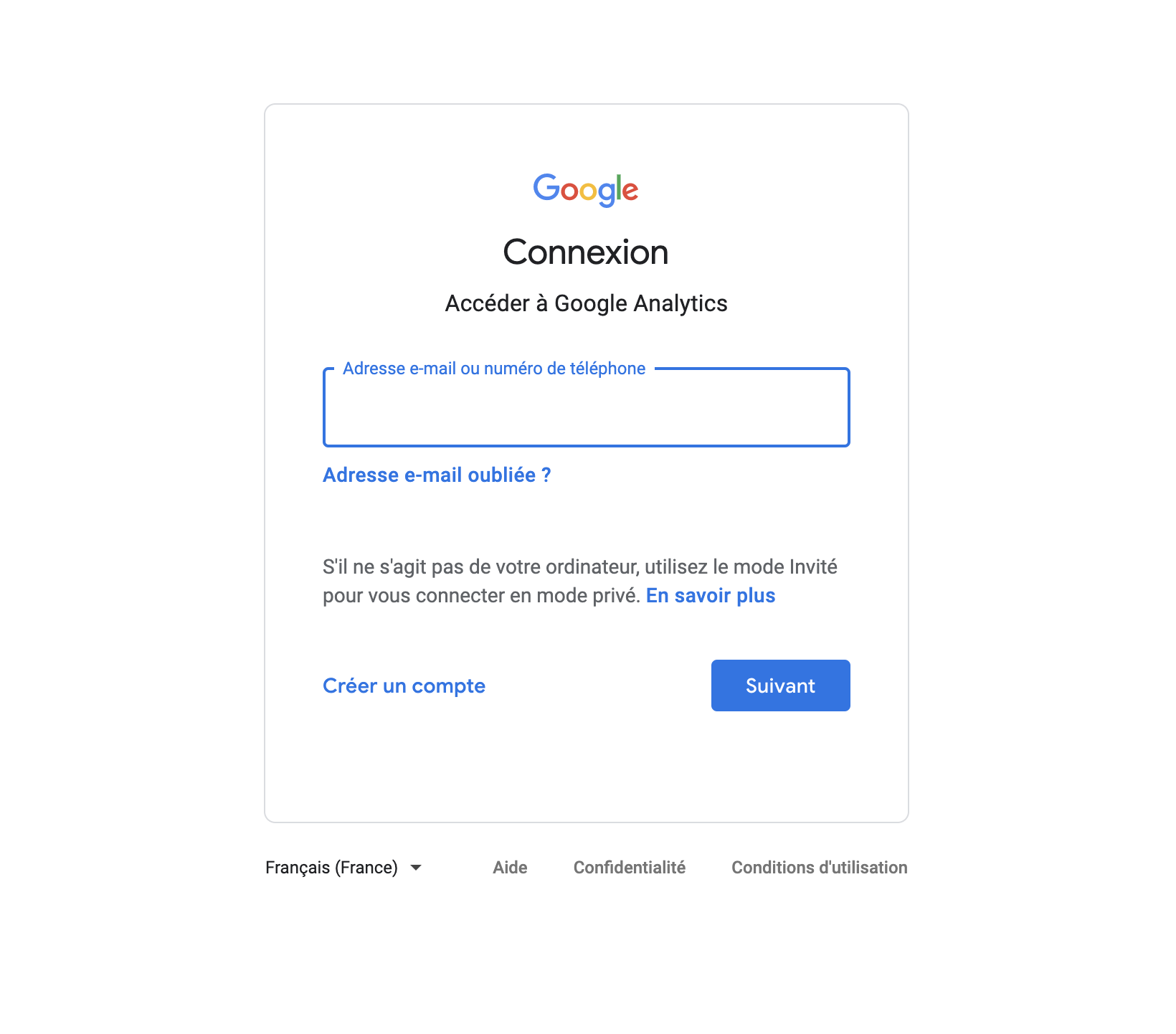
# Manipulate cookies 🍪

1. Login onto your ganalytics page
2. In your app, add a button that will make a google request when you click on it [(documentation here](https://docs.streamlit.io/en/stable/api.html#display-interactive-widgets)). The request must be done as follow: req = requests.get("<https://www.google.com/>")
3. Then get the cookies and display them in the app: st.markdown(req.cookies.\_cookies)
4. Now copy and paste your ganalytics URL and request it (careful, the following example correspond to MY ganalytics URL): req2 = requests.get(<https://analytics.google.com/analytics/web/#/report-home/a164062586w272485488p243020933>)
5. Display it in streamlit:
   1. st.text(req2.status\_code)
   2. st.markdown(req2.text)

Bonus: if you want to html to be interpreted in streamlit (dirty way), you can do:

| from html import unescape import streamlit.components.v1 as components components.html(unescape(req.text)) |
| --- |

1. Remark that the last part is a html that ask you to login to google (like if you try to paste the ganalytics URL in an private mode tab):



The goal of this part was only to see how we can access the cookies from a request via python. This is possible via many other techniques (command line, browser extension, etc.)

In the next part, we’ll see that it is possible to get specific cookies with login information.

# Request with oauth

1. For certain website it is possible to login with a post request and then get the cookies from this request to make other requests (the following lines are just an example of how it can be done):
   1. payload = {'inUserName': 'USERNAME, 'inUserPass': 'PASSWORD'}
   2. url = 'http://www.example.com'
   3. r = requests.post(url, data=payload)
   4. requests.get(other\_url, cookies=r.cookies)
2. However, I did not manage to login that way in ganalytics. There are other ways to login with google through your python code. Find it, login and fetch the “number of visitor” information
3. Finally display this information in your app

Deploy once again your app on heroku (be careful, don’t push any password in github, use env variables instead).

In the mail, add the URL of your application and the repository where your code is (if the mode is private, please share the access to: Saxamos).

At the end of the day you should have something that looks a bit like this (feel free to tune the frontend as you wish):

