### **Context**

The [Office of Foreign Assets Control](https://en.wikipedia.org/wiki/Office_of_Foreign_Assets_Control) (OFAC) publishes a list of sanctioned companies, individuals, and vessels that are prohibited from conducting any business with counterparties in the United States. If an American company violates this sanction regime, they’re likely to be heavily fined by the US government.

There are a number of other sanctions lists produced by foreign governments. The UK Treasury publishes a consolidated list of "asset freeze targets" that's somewhat similar to OFAC's list.

Processing semi-structured data like entries in these sanctions lists is an important part of what Sayari does.

### **Task**

The raw OFAC list can be found in XML [here](https://www.treasury.gov/ofac/downloads/sdn.xml). The raw UK Treasury list can be found in a number of formats [here](https://www.gov.uk/government/publications/financial-sanctions-consolidated-list-of-targets/consolidated-list-of-targets).

For this assignment we're also providing somewhat normalized versions of these datasets. The OFAC list corresponds to "ofac.jsonl.gz" and the UK Treasury list corresponds to "gbr.jsonl.gz". These files are gzipped json-lines format.

Although these files have mostly the same schema, there are some differences to watch out for (for example in how dates are formatted).

Your task:

1. Download the normalized versions of both lists. Take a look at the available fields.
2. Load each list into Spark.
3. Using Spark, find entities (people, companies, vessels, etc) that are present in both lists, and store these entities in an output file.

Some details about what we'd like to see:

1. For the output format, we'd like a dataset where each row represents a single entity found in both input datasets. There should be at least two columns that indicate the matched entity from the original datasets. The first column should be the id from OFAC's list under the name ofac\_id, and the second column should be the id from the UK list under the name uk\_id.
2. In addition to these ID fields, please provide some kind information about why the two entities are the same. You can use whatever schema you find easiest to convey this information. Just give us something that indicates why an entity from the OFAC list was matched with an entity from the UK list.
3. In addition to your Spark application, please send us your final output file.
4. For the Spark application, we'll accept either a PySpark (.py) file, or a .jar.

We'll be testing your application in local mode with something like the following command:  
spark-submit --master local[\*] app.(py|jar)

1. Please give us a list of third-party dependencies that you used. For Python dependencies we'd like a requirements.txt file. For Spark a list of packages to pass under the --packages option would be great.
2. Let us know the exact spark command we should use to run your application.
3. You are encouraged to use whatever third-party libraries you like, as long as we can easily reproduce your solution. No need to reinvent the wheel.

Final note: Finding entities in both datasets is meant to be challenging, since there is no perfect shared unique identifier in both datasets. There's not a perfect answer for this task, but we are interested in seeing a sensible approach to the problem with a good balance between quality and quantity of matches. While we aren't expecting a production-ready unit-tested repository, we will be looking for fundamentally good code quality and understandable code.