Challenge 4

In this challenge, we had to subscribe to the events for adding, modifying, and deleting users. When these events occurred, a log record (with Info level) containing certain information had to be generated.

Optionally, we were asked to verify the content of the input JSON and check for possible GAM errors. If there were any, they had to be logged.

To get started, let's first create the procedure that will be executed when the events are triggered.

The first thing we will do is to define the rules it will have. As mentioned in the theory session, the event name and input JSON will have IN, and the output JSON will have OUT.

Going to the code itself, first we will check that the event is one of those indicated in the instructions. If it is not, we will log an error. If it is, we are going to divide the code in two Subs only to make it more organized and easier to understand. The first one will be in charge of obtaining the variables that we were asked to log. The second one will impact the previous data in the log.

Let's start to see them.

We will obtain the username from the input JSON through the GAMUser.

As for the IP address, we will obtain it through the GAM session, and for that we use the primitive one that it provides. Once the session is obtained, the IP is contained in the *InitialIPAddress* attribute.

Checking for GAM errors was optional according to the challenge instructions, so before saving the IP we will do that to avoid possible errors. For this, we run through the collection of errors returned by GAMSession when using it, and in case we find any, we save it and then log with *Error* level.

Finally, the other Sub will be executed, and it will log the information requested in the challenge. Obviously, we took the name of the event from the input variable.

The other optional point was to verify the content of the input JSON, and we do it in the following way. If it is empty, we log again with *Error* level.

With this, we have finished the programming part, and now we have to subscribe to the events through the GAM Backoffice.

As we said in the theory session, to do this we must go to Settings/Event Subscription.

After clicking on Add we start to fill in the data.

We choose the first event (which is to modify in this case) and enter:

- The file name (this was the name of the file that will listen to the execution of the event)
- The name of the class (this was the name of the program), and finally
- The name of the method (which is "execute").

Once this is done, we confirm.

We repeat this process with the other two events.

Given the limitations of using the cloud, we will not show that the log record is actually generated with the data when a user is added, removed, or modified. You can confirm this by testing it yourself in your own work environment.