**The Calendar APIs were deployed on azure here is the link if you would like to access it:**

[**https://my-calendar-be.azurewebsites.net/swagger/index.html**](https://my-calendar-be.azurewebsites.net/swagger/index.html)

**Features:**

1. **Ability to create an event in your calendar.**
2. **Ability to search for an event in your calendar or sort them.**
3. **Ability to delete an event via id in your calendar.**
4. **Ability to delete ALL the events in your calendar. (be careful with that one).**
5. **Implements refresh token logic to handle the expiration of access tokens.**
6. **Implements a search method.**
7. **Allows user to add attachments.**
8. **Implements pagination.**

**Here is a small 2 minutes video to show how to get started using the calendar APIs in case if you feel like not reading the instructions below:**

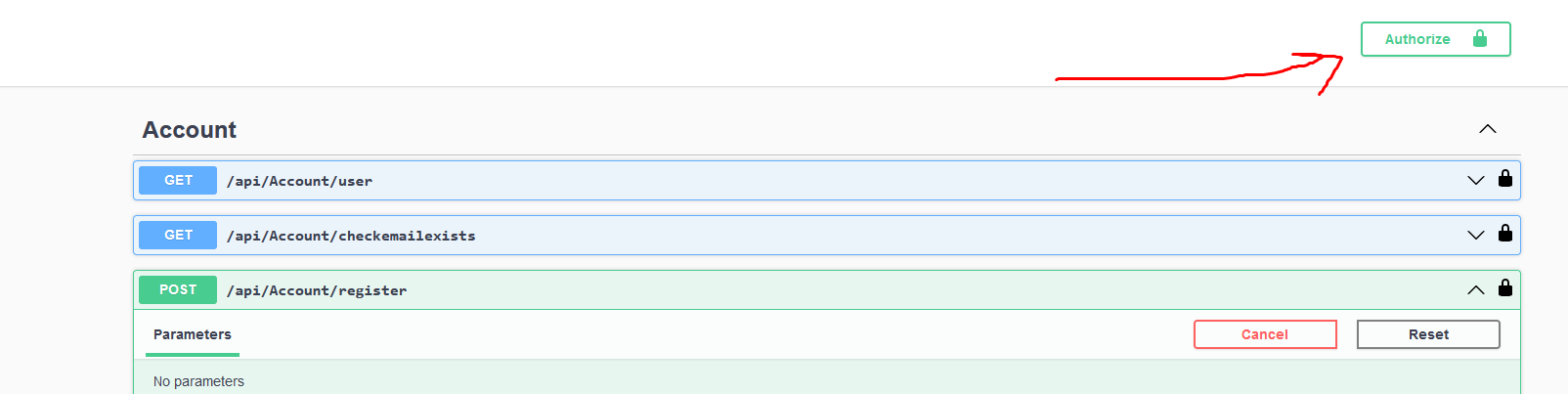
[**https://www.loom.com/share/08e71ebd5645425199af03d46b5bc8c7**](https://www.loom.com/share/08e71ebd5645425199af03d46b5bc8c7)

**The first step to follow to use the Calendar APIs first is Registering:**



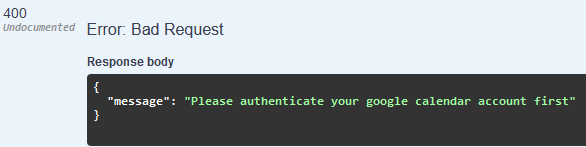
**Note: password must follow Asp.net core criteria (Password must be 6 to 10 characters long and include at least one digit, one lowercase letter, one uppercase letter, one non-alphanumeric character, and no spaces) I suggest making it like this for example: [your-name]\_1. like (Ahmed\_1)**

**Next you will be granted with a token like this one** 

**Please copy that token in the authorize section** 

**Next step is for the google authentication process.**

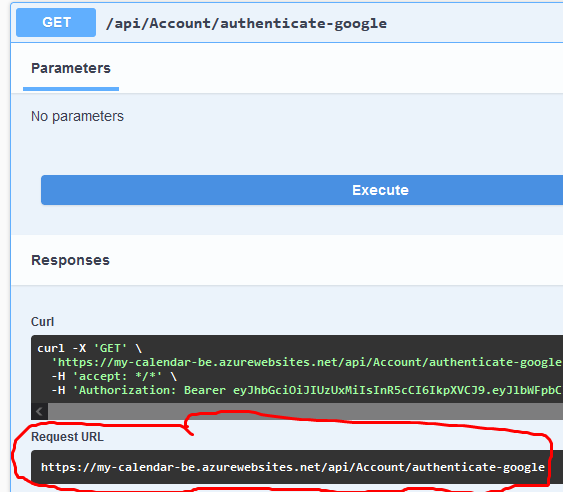
***Warning*: Trying to use any of the calendar APIs in this point will give you this response:**



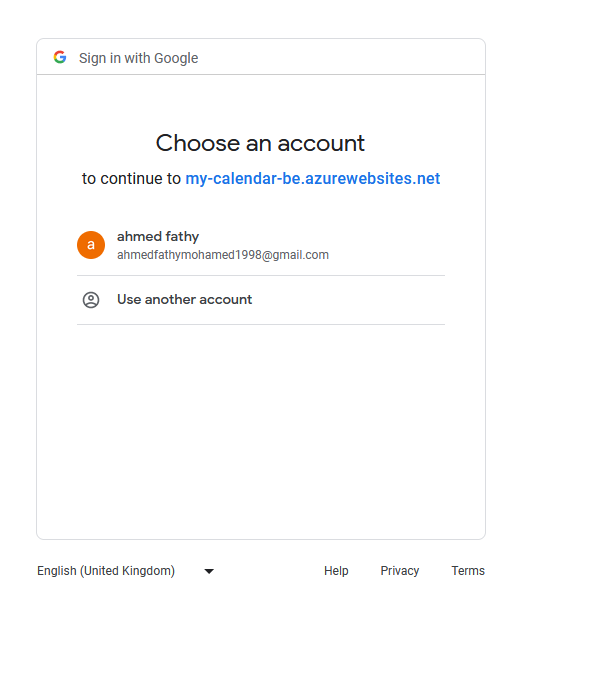
**So we will authenticate google calendar account first by clicking on the API api/account/authenticate-google.**

**That section needed a front-end to be implemented in a better way but it can be managed by doing the following:**

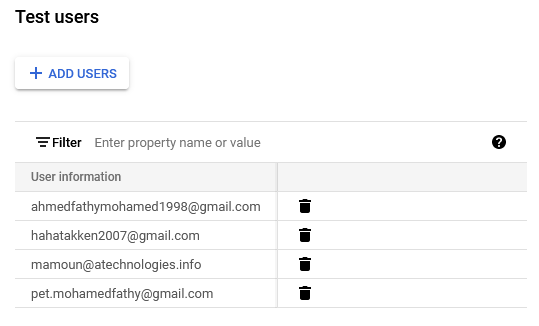
**Copy the URL circled in a new tab**



**You will be met by a screen like this:**

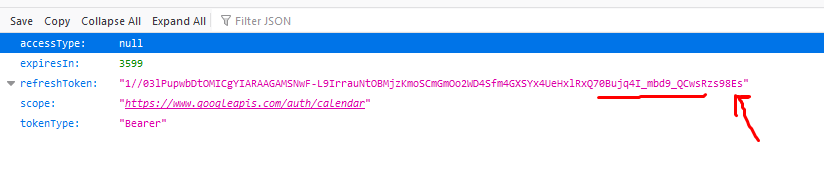


**Warning: you must be among these test users in order for this to work:**

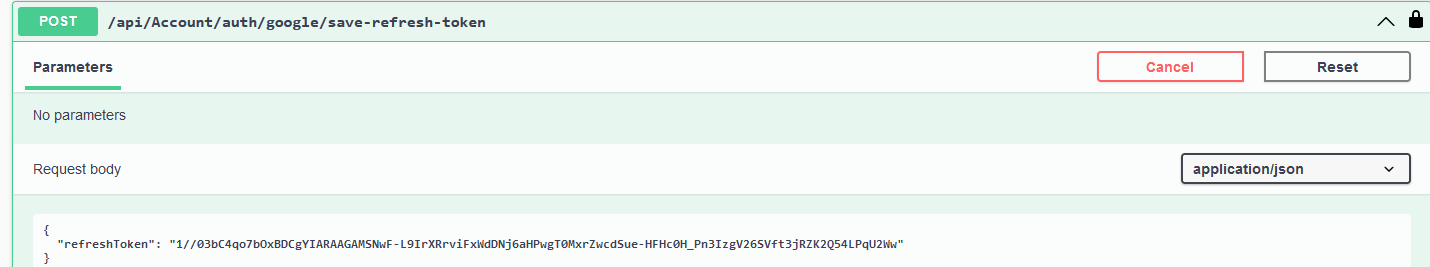


**I have added** [**mamoun@atechnologies.info**](mailto:mamoun@atechnologies.info) **among the authorized testers.**

**The result of the aforementioned step will be a Json response like this one:**



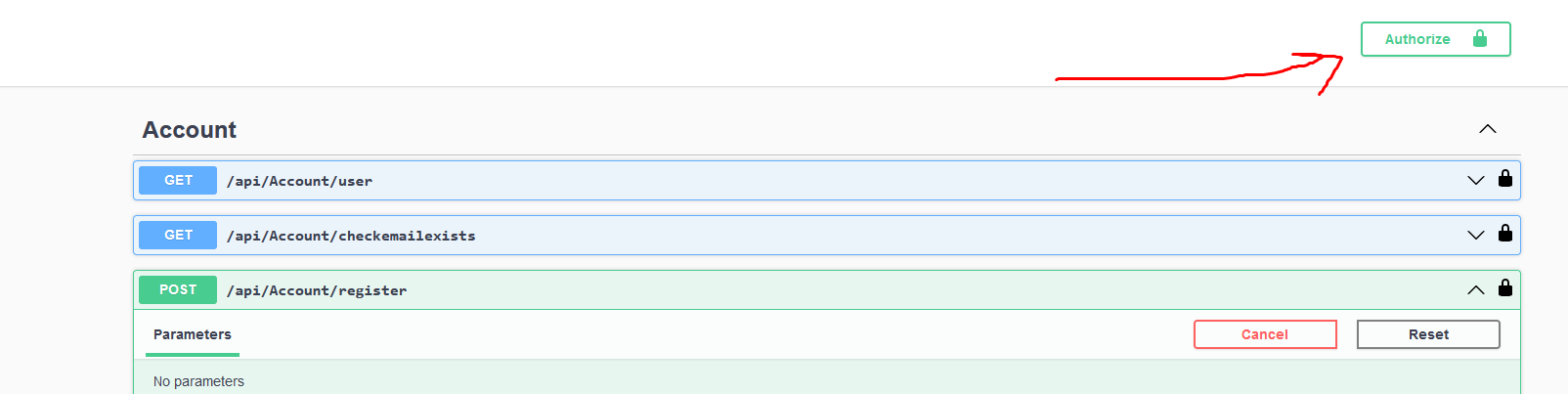
**The last step you need to follow is to copy that refresh token and put it inside the google/save-refreshtoken-api:**



**The response should be a new jwt token:**



**Then we insert that token in the authorize:**



***Now you can use all the calendar APIs***