# Our Goals and Achievements

We are interacting with the Dialogflow API using Postman. Within Postman, we are testing the API and building tools that will assist Project X Team in implementing it in the front-end code in order to connect it with Dialogflow and interact with it similarly.

# Sprint 3: Interact between Dialogflow API with Postman to solve a problem when trying to update or edit Patient ID

Following the completion of the most recent sprint, in which we included all of the Authorization information from the previous sprint, we are now able to interact with the Dialogflow API.

When attempting to connect from the team Project X website to the chatbot, we encountered the same session problem as occurred when we were trying to modify the patient ID. During this iteration of the sprint, we are going to collaborate closely with Nathan, a member of team Project X, in order to modify the Patient ID by utilising the PATCH command.

**POST command**

use the Dialogflow API to determine intent using a patient id called "test123."

POST: <https://dialogflow.googleapis.com/v2/projects/sit-22t1-team-avengers-c824a5b/agent/sessions/123:detectIntent>

Request Body:

{

"queryInput": {

"text": {

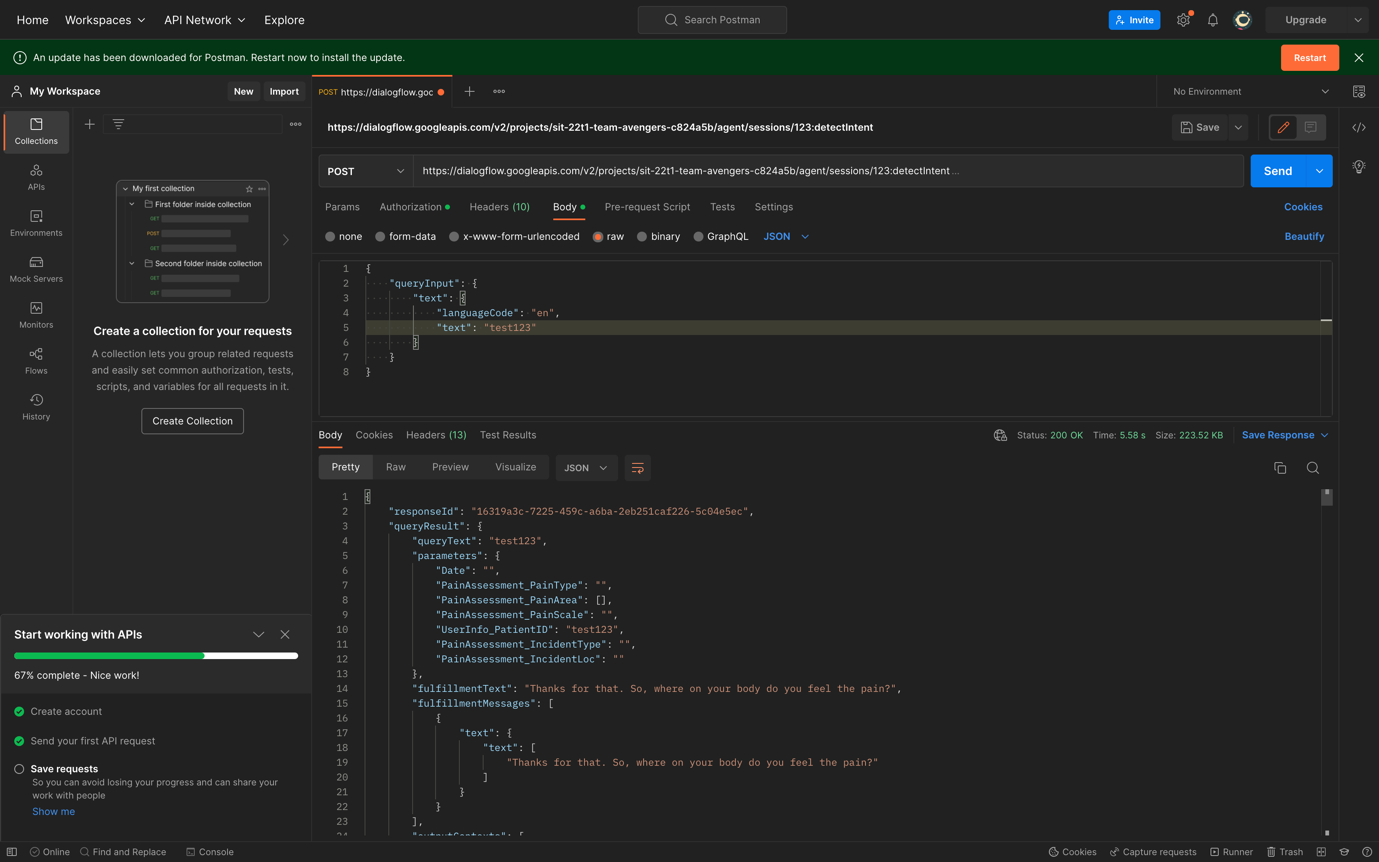
"languageCode": "en",

"text": "I hurt myself"

}

}

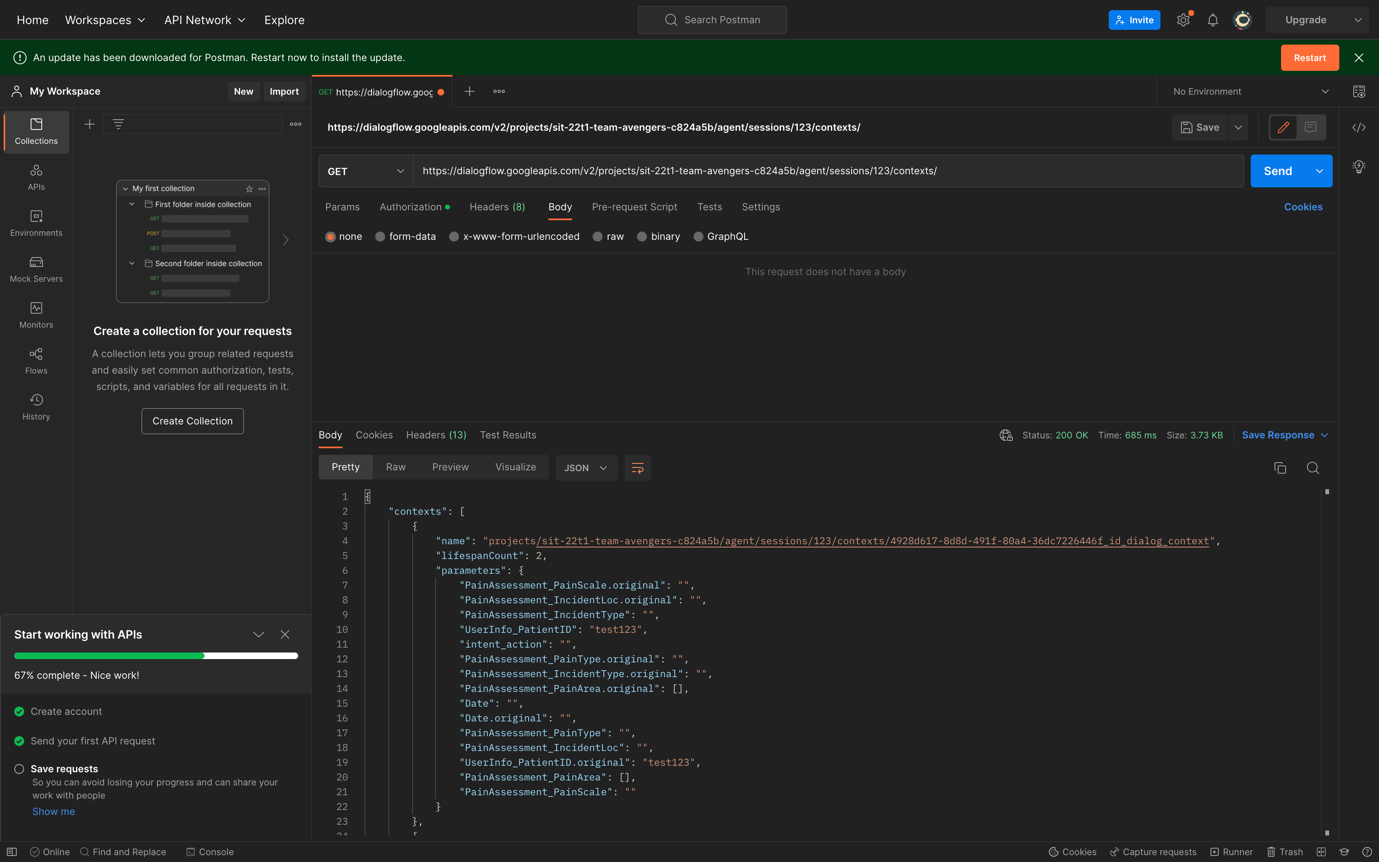
}



**GET command**

obtain the context information as well as the context ID, both of which will be placed to use in the process of updating the Patient ID.

GET: <https://dialogflow.googleapis.com/v2/projects/sit-22t1-team-avengers-c824a5b/agent/sessions/123/contexts/>



**PATCH command**

update the Patient ID to from “test123” to “newid”.

PATCH: <https://dialogflow.googleapis.com/v2/projects/sit-22t1-team-avengers-c824a5b/agent/sessions/123/contexts/context-id>

Request body:

{

"parameters": {

"PainAssessment\_IncidentLoc.original": "",

"intent\_action": "",

"Date.original": "",

"PainAssessment\_PainType": "",

"PainAssessment\_IncidentType.original": "",

"PainAssessment\_PainArea": [],

"PainAssessment\_PainArea.original": [],

"PainAssessment\_PainScale.original": "",

"UserInfo\_PatientID.original": "newid",

"PainAssessment\_IncidentLoc": "",

"Date": "",

"PainAssessment\_PainScale": "",

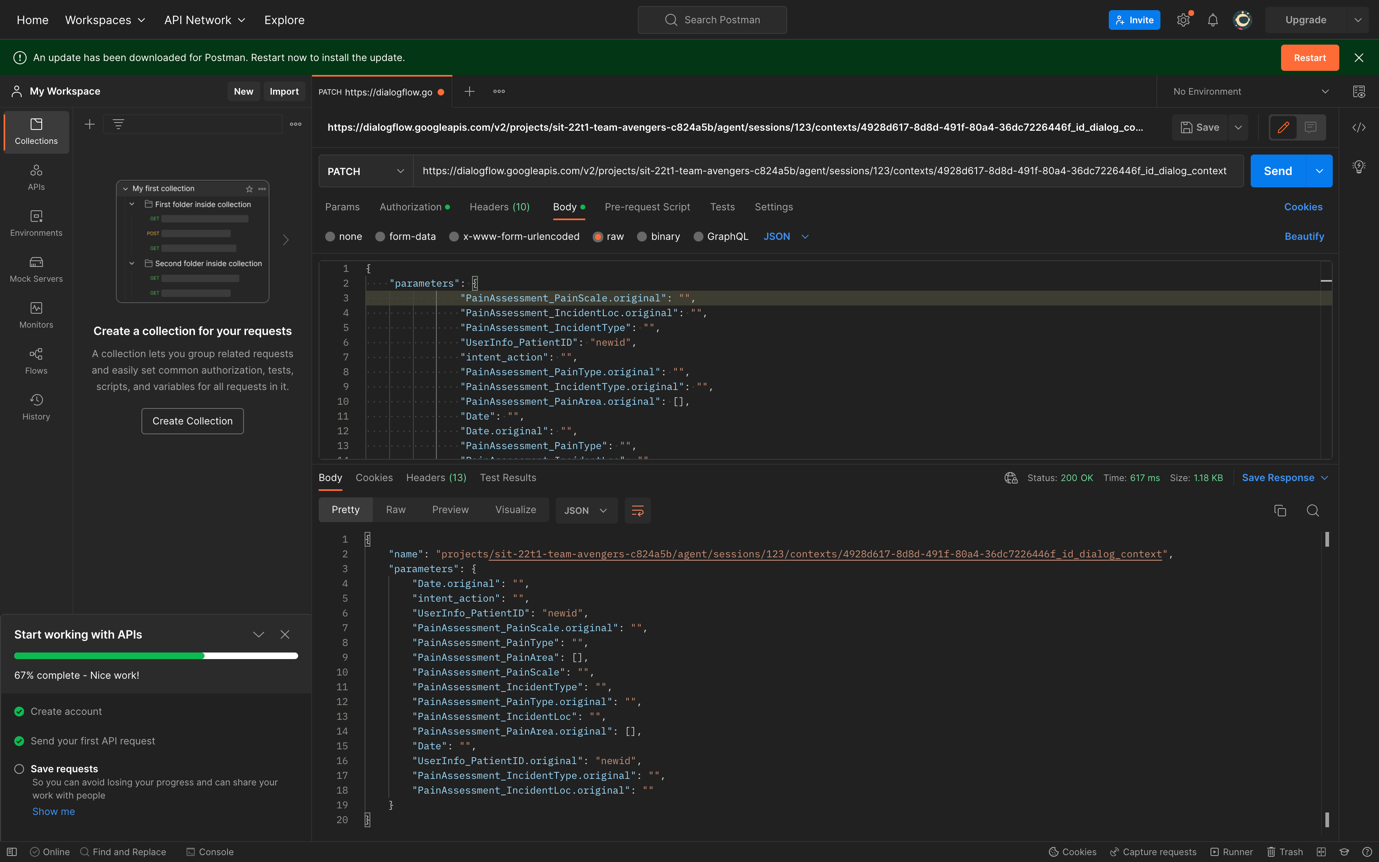
"UserInfo\_PatientID": "newid",

"PainAssessment\_PainType.original": "",

"PainAssessment\_IncidentType": ""

}

}



**POST command**

After the Patient ID has been successfully updated, we will now attempt to create a new context in the expectation that the API will recognise another intent on our part without prompting the user to provide their Patient ID.

POST: <https://dialogflow.googleapis.com/v2/projects/sit-22t1-team-avengers-c824a5b/agent/sessions/123:detectIntent>

Request body:

{

"queryInput": {

"text": {

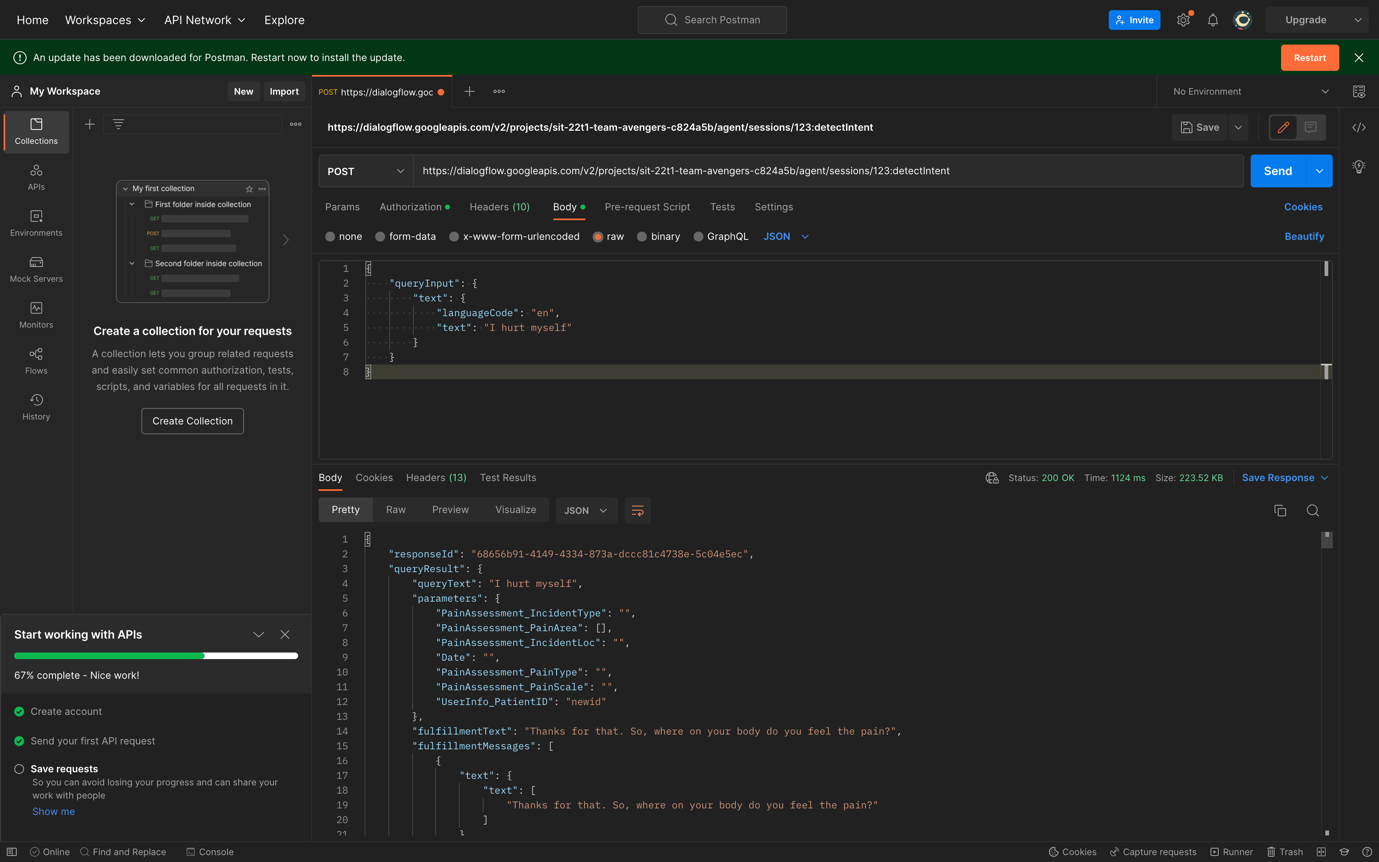
"languageCode": "en",

"text": "I hurt myself"

}

}

}



As we can see now, the new Patient ID is now updated, and the chatbot is asking a new question when the user has their Patient ID. This is an update that has just been implemented.

# Sprint 4: Interact between Dialogflow API to create a context using Postman

Before we can use Postman to create a context, we need to first perform the POST command to initiate a session. After that, we are able to test the context using the detect intent in order to confirm the Patient ID.

**POST Command**

We create a session called 123 and in the request body we named the context as “hello”. In the parameters, we named the Patient\_ID as “newid”.

POST: <https://dialogflow.googleapis.com/v2/projects/sit-22t1-team-avengers-c824a5b/agent/sessions/123/contexts>

Request body:

{

"name": "projects/sit-22t1-team-avengers-c824a5b/agent/sessions/123/contexts/hello",

"lifespanCount": 2,

"parameters": {

"PainAssessment\_IncidentLoc.original": "",

"intent\_action": "",

"Date.original": "",

"PainAssessment\_PainType": "",

"PainAssessment\_IncidentType.original": "",

"PainAssessment\_PainArea": [],

"PainAssessment\_PainArea.original": [],

"PainAssessment\_PainScale.original": "",

"UserInfo\_PatientID.original": "newid",

"PainAssessment\_IncidentLoc": "",

"Date": "",

"PainAssessment\_PainScale": "",

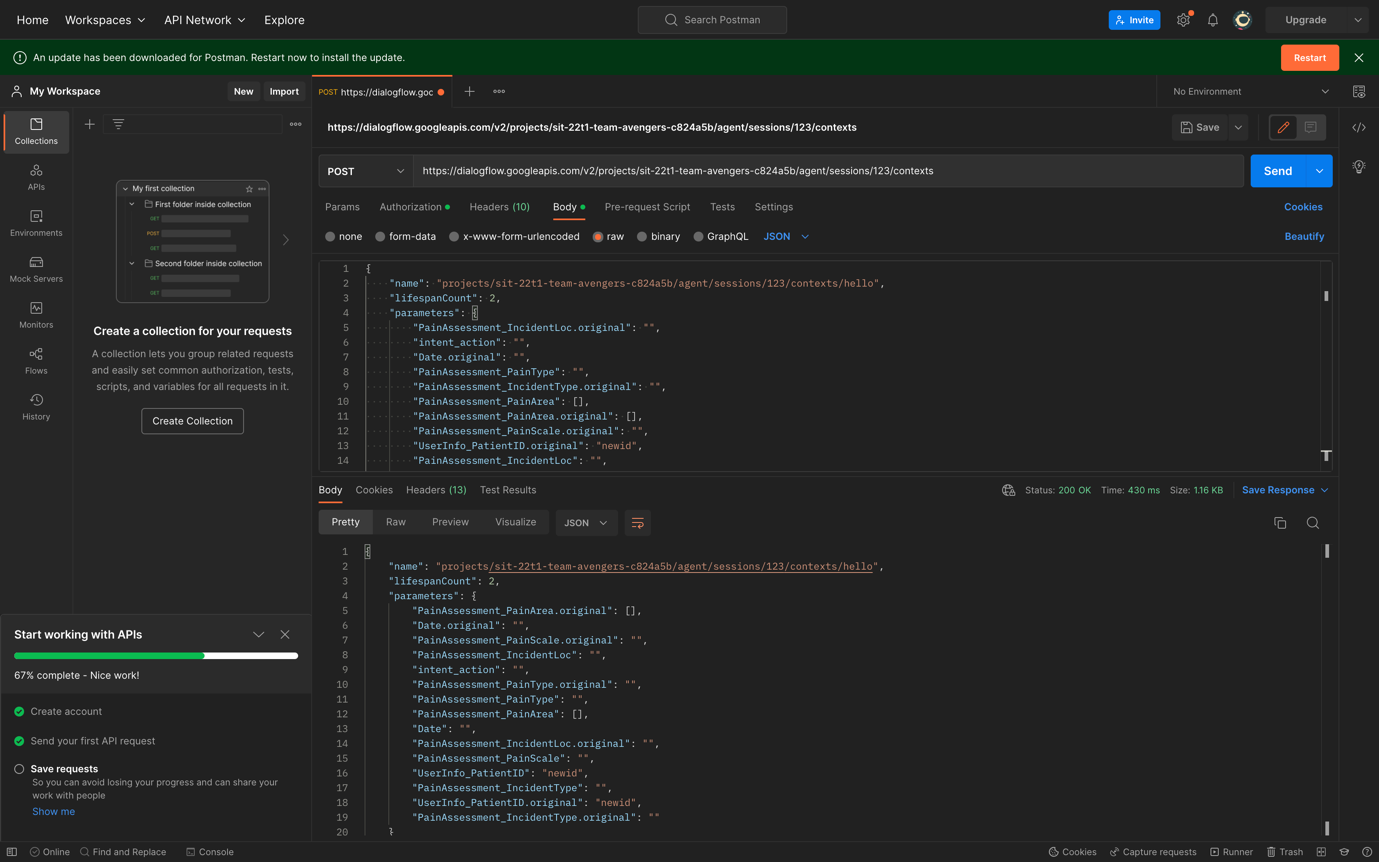
"UserInfo\_PatientID": "newid",

"PainAssessment\_PainType.original": "",

"PainAssessment\_IncidentType": ""

}

}



While we are in the process of sending the request, we are able to see that there is a context with the name “hello” and that the patient ID starts with “newid”.

**POST Command**

Next, we will validate the context that is referred to as “hello” by using the detect intent to check the patient ID.

POST: <https://dialogflow.googleapis.com/v2/projects/sit-22t1-team-avengers-c824a5b/agent/sessions/123:detectIntent>

Request body:

{

"queryInput": {

"text": {

"languageCode": "en",

"text": "Hi"

}

}

}

Graphical user interface, text

Description automatically generated

The Medi Chatbot will respond with the fulfilment message after the request has been sent, and the Patient ID will continue to be referred to as "newid."

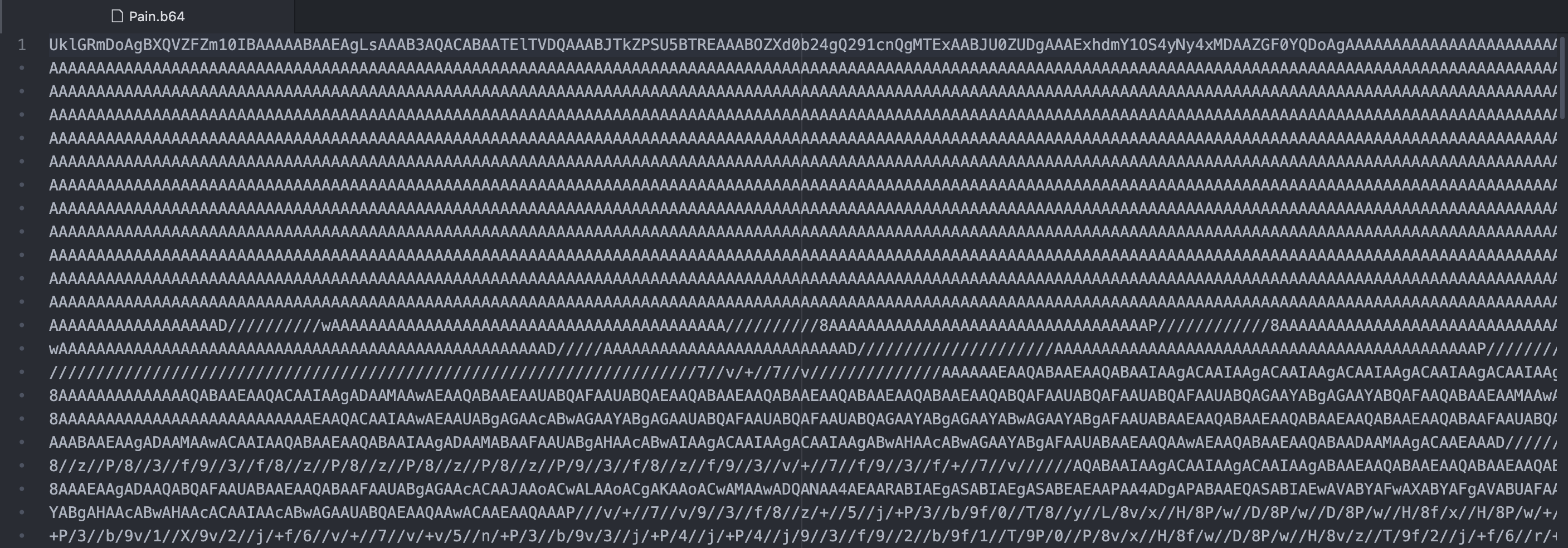
# Sprint 5: Interact between Dialogflow API to test the Speech-to-text function using Postman

In this sprint, we going to work on how to send audio input to a detect intent request using the API. Dialogflow processes the audio and converts it to text before attempting an intent match. This conversion is known as audio input, speech recognition, speech-to-text.

First, we need to have a sample input audio file called “Pain.wav”, which says “I am in pain”, and we have to encoding the audio file to base64 using this command:

* For Mac user: **$ base64 -i Pain.wav -o Pain.b64**
* For Window user: **C:> Base64.exe -e Pain.wav > Pain.b64**
* For Linux user: **$ base64 Pain.wav -w 0 > Pain.b64**

This is the output as shown below:



**POST command**

After successfully encoding the audio file, we use “detectIntent” command to detect the audio file from Postman, and we need to copy the base64 encoded file to the “inputAudio” in the JSON request body.

POST: <https://dialogflow.googleapis.com/v2/projects/sit-22t1-team-avengers-c824a5b/agent/sessions/123:detectIntent>

Request body:

{

"queryInput": {

"audioConfig": {

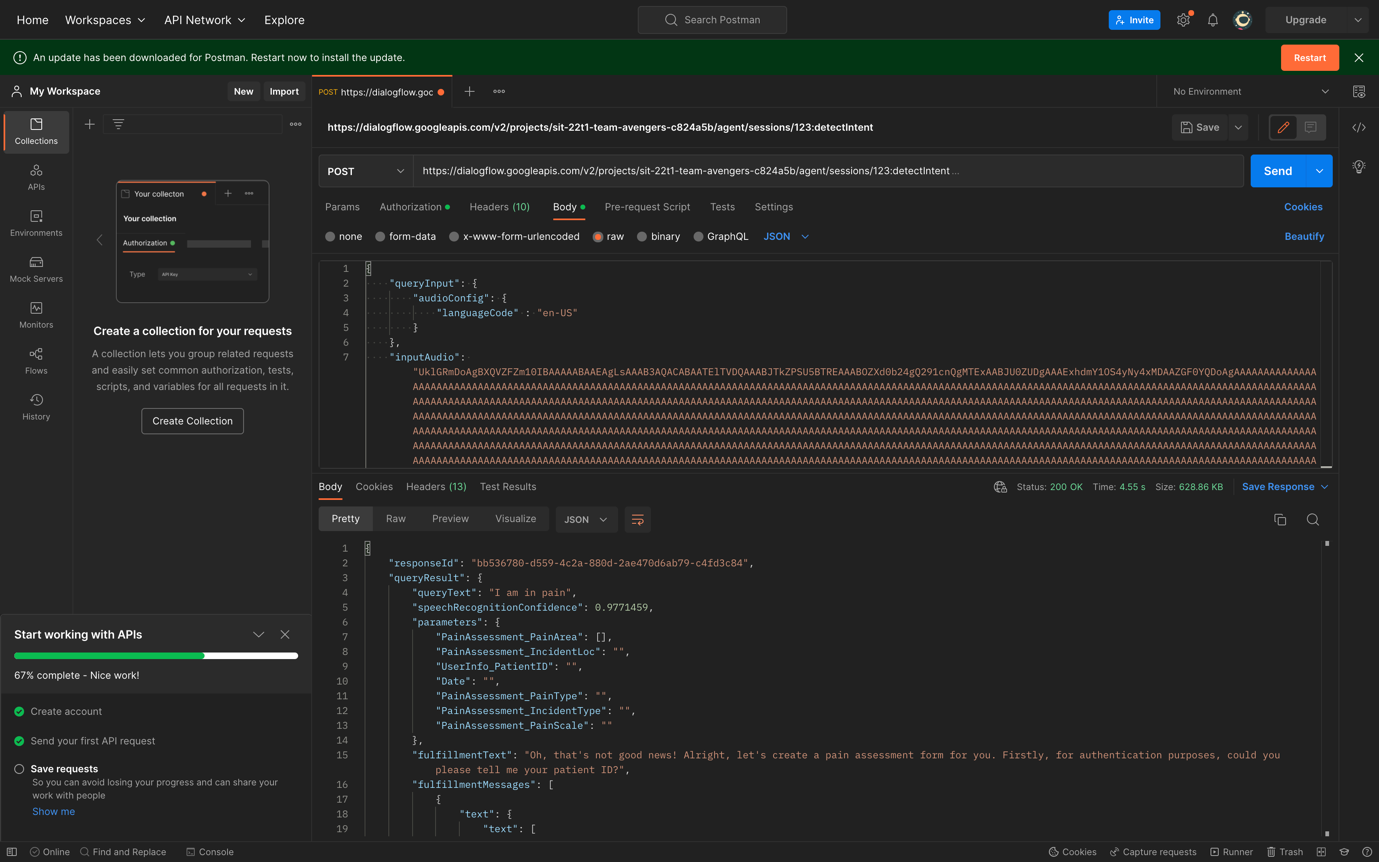
"languageCode" : "en-US"

}

},

"inputAudio": "UklGRmDoAgBXQVZFZm10IBAAAAABAAEAgLsAAAB3AQAC….”

}



After the request has been sent, the MediChatbot will detect the audio file in the request body and provide a response with the fulfilment message.