API GATEWAY REQUEST VALIDATION

Hello! folks its awesome to have you here working towards understanding how to validate the request. So, let’s get started and understand it through an interesting tutorial.

Steps:

1. [Create DynamoDB Table](#_CREATE_DYNAMO_DB)
2. [Create IAM role](#_CREATE_IAM_ROLE)
3. [Create API](#_CREATE_API)
4. [Configure Method](#_CONFIGURE_METHOD)
5. [Configure Request Validation](#_ADD_REQUEST_VALIDATION)
   1. Validating The Query Strings
   2. Validating The Request Body
6. [Test the API](#_TEST_API)

## **CREATE DYNAMO DB TABLE**

Let’s login to AWS and open DynamoDB and start creating a table.

Graphical user interface, application

Description automatically generated

Follow it up by giving the table name and the partition key and opt default settings for other configurations. You’d have a similar screen

Graphical user interface, text, application, email

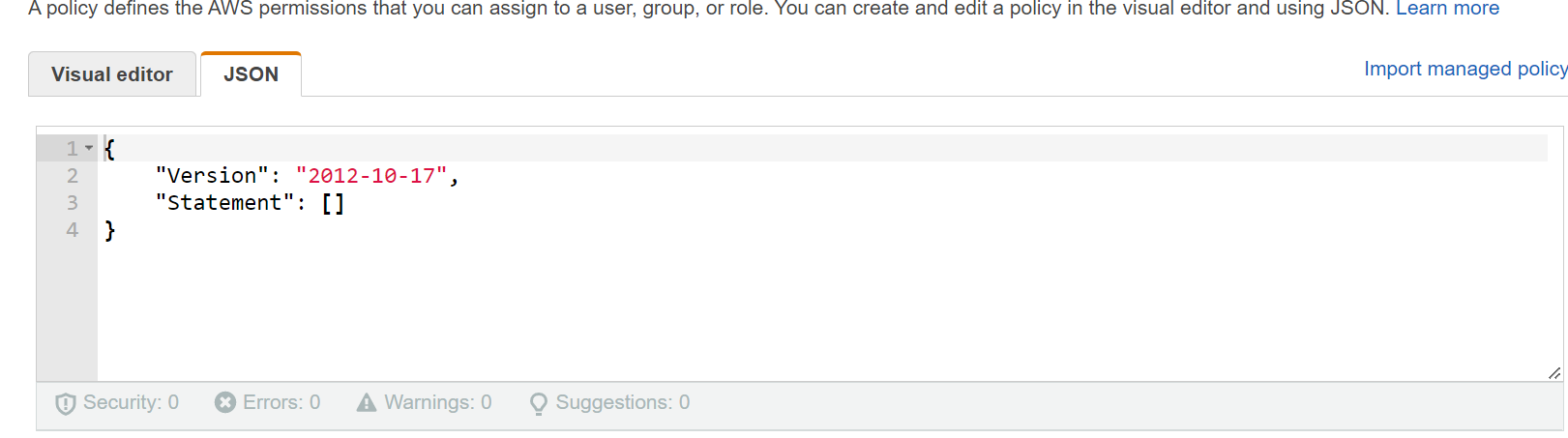
Description automatically generated

Once Done **CREATE TABLE.** This marks the end of this step.

## **CREATE IAM ROLE**

Search for IAM in AWS and then go into **Policies** and create a custom policy so that we can attach it to the role that we’d be creating following up to this.

You’d be having a similar screen



Now to generate the policy we’d be using [AWS policy generator](https://awspolicygen.s3.amazonaws.com/policygen.html) . Once you enter select the policy type as **IAM**, AWS Service as **Amazon DynamoDB**, In Actions select **PutItem**, ARN give the ARN of the DynamoDB table that we created.

Graphical user interface, text, application, email

Description automatically generated

Then select **Add Statement** follow it up by selecting **Generate Policy.**

Graphical user interface, text, application, email

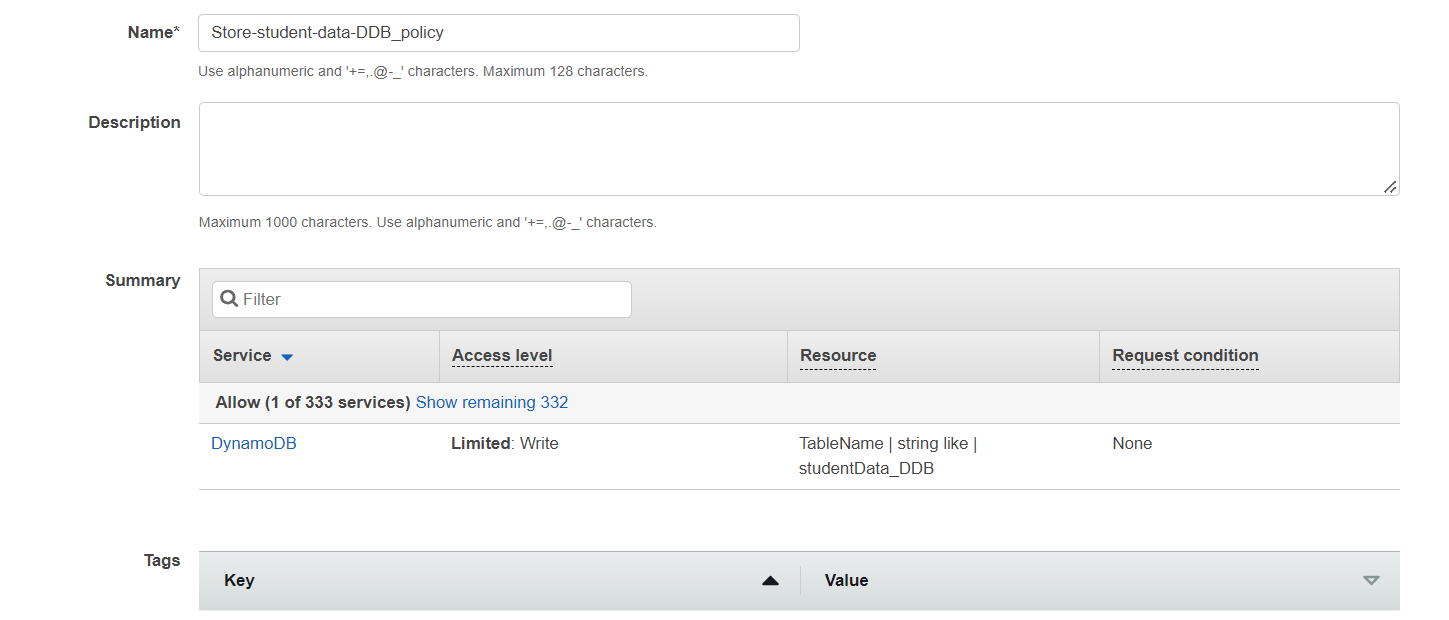
Description automatically generated

Copy this policy and paste in IAM console

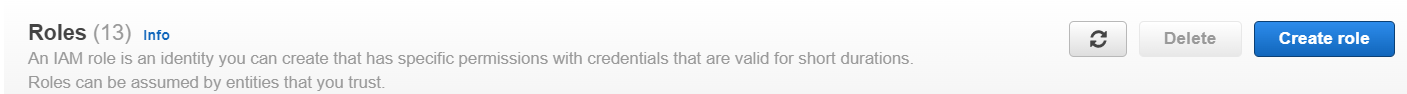
Graphical user interface

Description automatically generated with medium confidence

Once its done click on Next Tags, you don’t have to provide tags just click on **review policy**. In this give a name for the policy and hit **create policy**.



Once the policy is created click on **Roles** and create a new role.



In use cases select API Gateway as we’ll be creating this role for it then click on next **Permissions.**

A picture containing graphical user interface

Description automatically generated

In this section we’ll not be able to select the custom policy so just move to the next section. Here add a name for the role and **create Role**. Once it is created open it under Add Permissions click **Attach Policies,** you’d be taken to the policies section and her just select the custom policy that you have created.

Graphical user interface, application

Description automatically generated

This marks the end of this step.

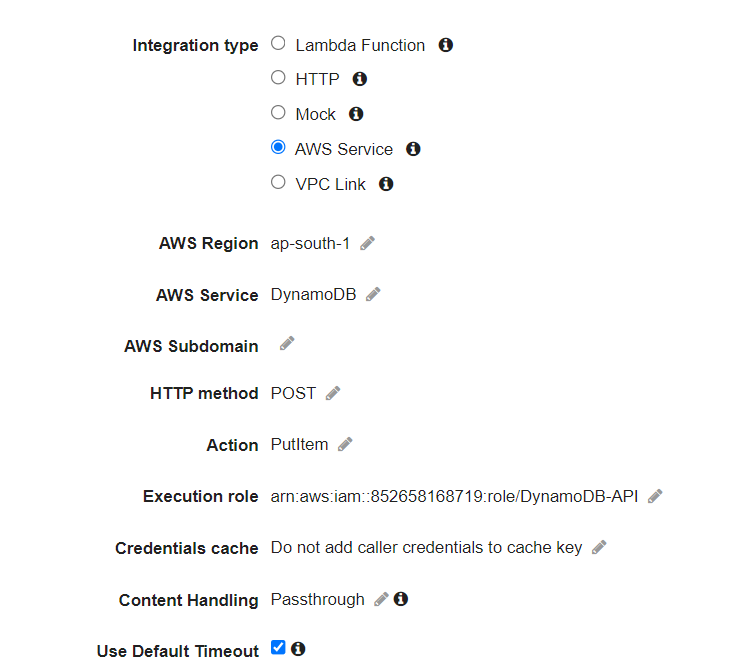
## **CREATE API**

Go to API Gateway and build a Rest API once its created add a new **resource**

Graphical user interface

Description automatically generated with medium confidence

Now add a **method** and choose **POST.** Configure this method’s Integration type as AWS Service and give the configuration as shown below, for execution role copy the arn from IAM console. Check your AWS Region!!.



And then create **Save.** Once done you’d have a similar screen

Graphical user interface, application

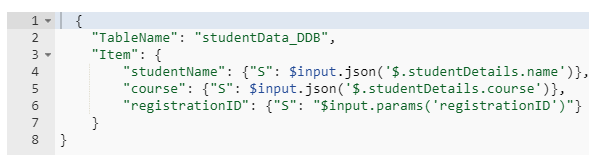
Description automatically generated

## **CONFIGURE METHOD**

Now go to Integration Request and configure the method in this select **mapping template** follow it up by providing the second option in **Request body passthrough**Graphical user interface, text, application

Description automatically generated

Once you tick it then give the information which should be constructed so that DynamoDB would be able to understand.



And then hit **Save.**

## **CONFIGURE REQUEST VALIDATION**

Go to **Method Request** and add request validator

1. Validating Query String Parameters
2. Validating Body

**Validating query string** do the required as shown in the below picture. I have given the same query string as our Partition key for our table, you can do it any way you please.

Graphical user interface, text, application

Description automatically generated

**Validating the body**

There are few more steps such as creating a **Model.** You can create it via online generators to do so click [ [here](https://www.liquid-technologies.com/online-json-to-schema-converter) ]. For this example, I have created and stored it in the student info-Model file. Now copy the json schema to the Models section in the left and paste. In this code I have made **tutionFee not required.** You’ll see it will be testing.

Graphical user interface, text, application

Description automatically generated

After this click save and then go to Resources and go to **Method Request** and click **request body** inside it add the model file

Graphical user interface, application

Description automatically generated

This marks the end of configuring the request validations.

Then go to Actions and **Deploy API**

Logo

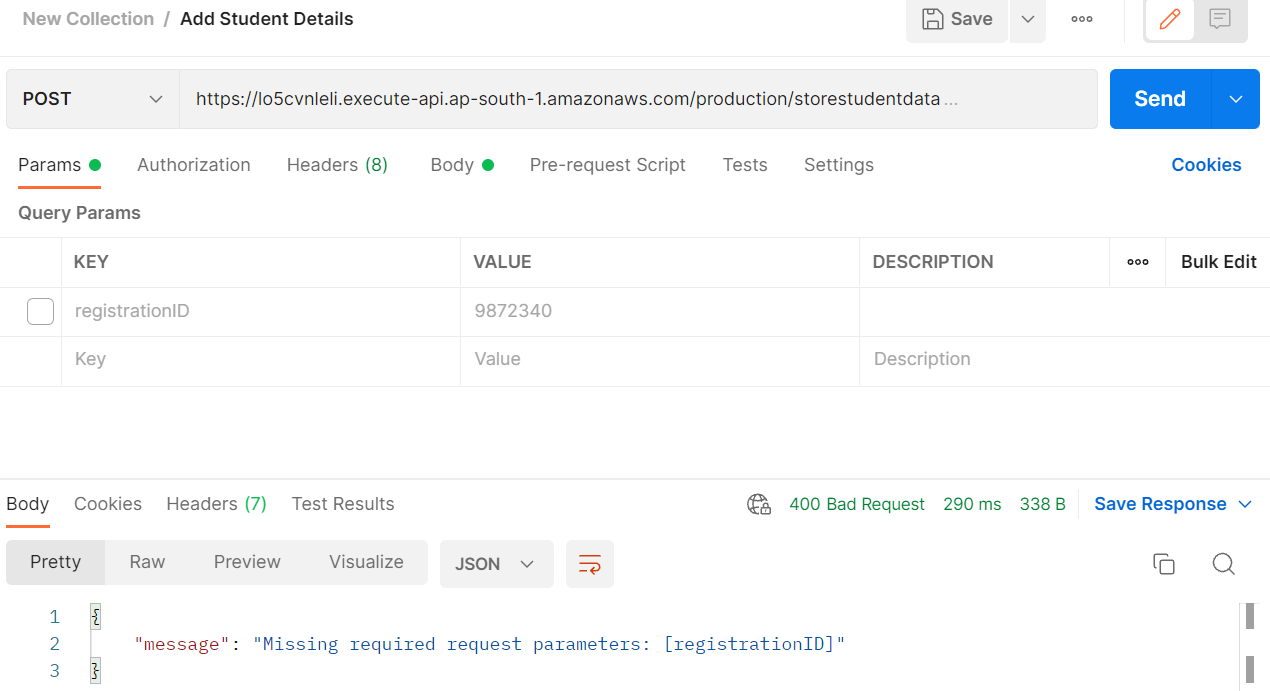
Description automatically generated with low confidenceGraphical user interface, table

Description automatically generated

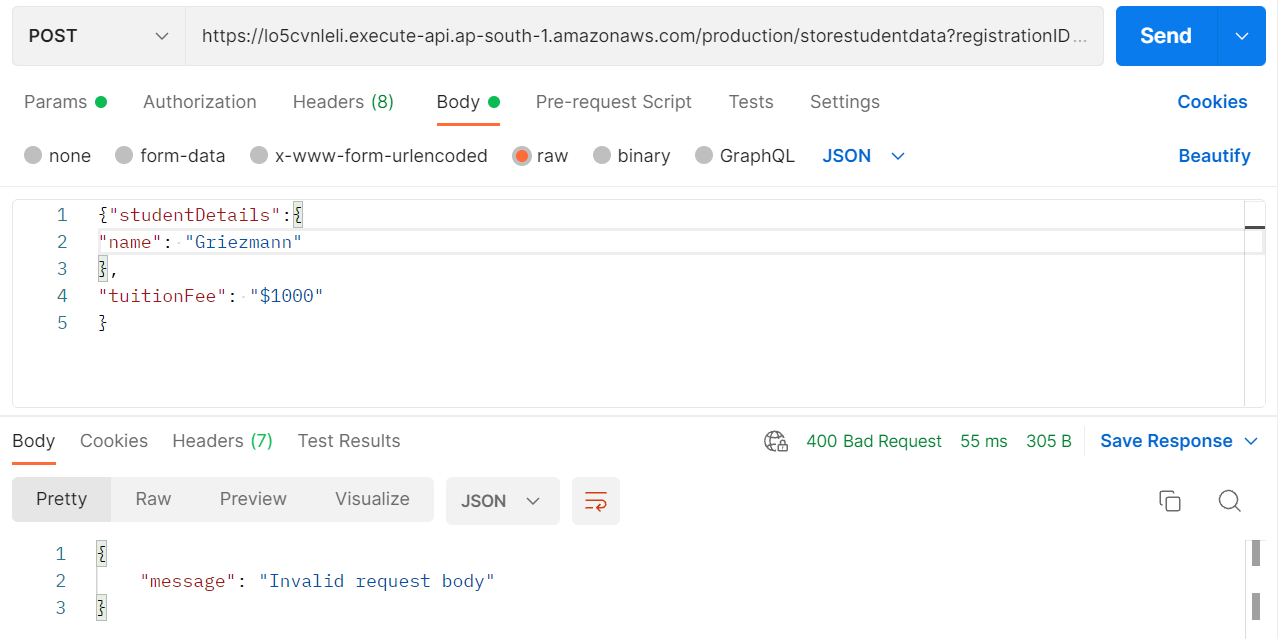
## **TEST API**

Use Postman to do the testing

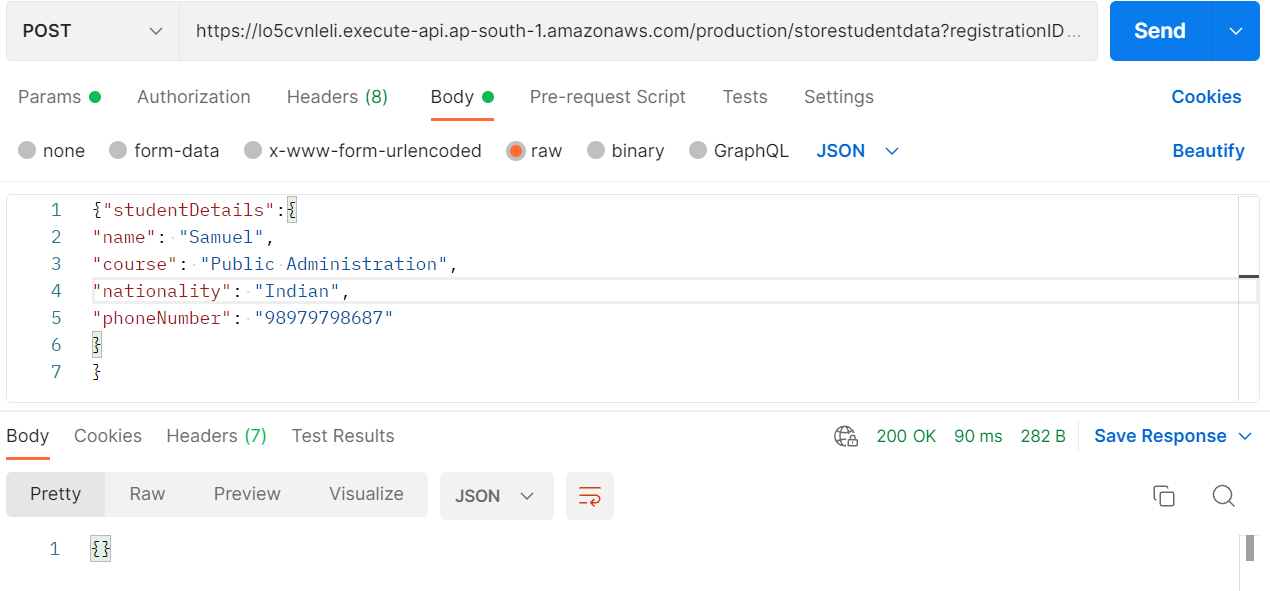
**Scenario 1**: Check by not giving the query string [registrationID], body already has the request payload.



**Scenario 2**: Give the query string but in payload try not giving any key in studentDetails



Scenario 3: Try not giving tuitionFee



It worked and you can see it in DynamoDB Table as well Graphical user interface, text, application, email

Description automatically generated

This marks the end of the tutorial, thank you for sticking till the end and finishing the tutorial, hope you understood the concepts of implementing request validator, models, DynamoDB Table and how to integrate with an API. **Happy Learning!!**

References:

* <https://docs.aws.amazon.com/>
* https://www.youtube.com/watch?v=HP29MR8RVus