In the last lesson, you created a timed lambda function that looks at all the speeders files, and aggregates them into a file with the key speeders-full/full.csv.

Cody has asked you to create an API letting her trigger an update to that file at any time. She wants to be able to have the most up to date data for council meetings.

In this exercise, you will create a lambda function called triggerFullreportBuilder that will invoke the fullReportBuilder function from the previous lesson. We will do so asynchronously so the function can execute quickly!

**File to edit**

You will be editing the triggerFullReportBuilder/lambda\_function.py file in the editor.

**Slides**

Don't forget - you can always click on the **slides** icon in the top right to see the slides - they'll be pretty helpful!

**Instructions**

**100XP**

* Edit the triggerFullReportBuilder/lambda\_function.py file.
* Grab the async parameter from the query string to determine execution type.
* Invoke the lambda function passing that parameter, and return.
* In the terminal, run python3 run\_lambda.py to deploy and test your lambda function.

Code:

import boto3

lambda\_client = boto3.client('lambda', aws\_access\_key\_id="12345",

aws\_secret\_access\_key="12345",

region\_name='us-east-1',

endpoint\_url="http://localhost:4574")

def trigger\_full\_report\_builder(event, context):

async = event['queryStringParameters']['async']

if async == 'true':

exec\_type = 'Event'

else:

exec\_type = 'RequestResponse'

result = lambda\_client.invoke(

FunctionName='fullReportBuilder',

InvocationType=exec\_type)

return {

'statusCode': 200,

'headers': {

"content-type" : "application/json"

},

'body': "Successfully Invoked"

}