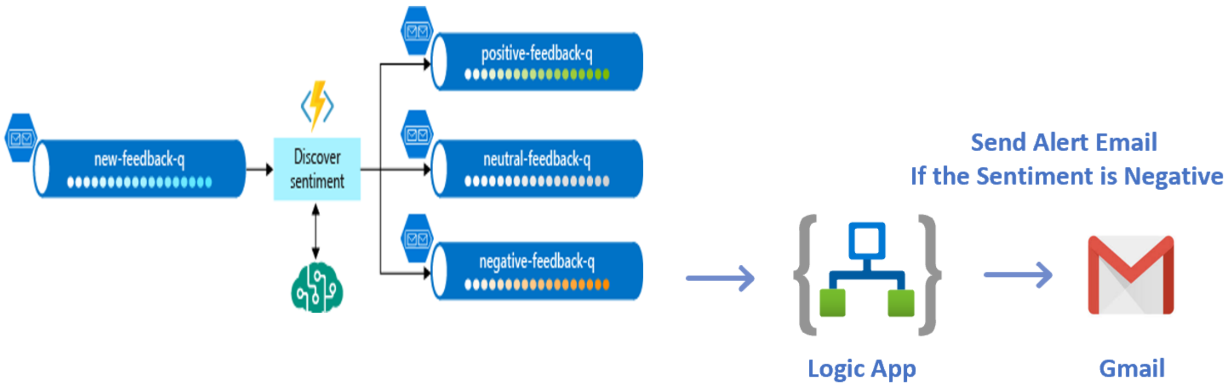
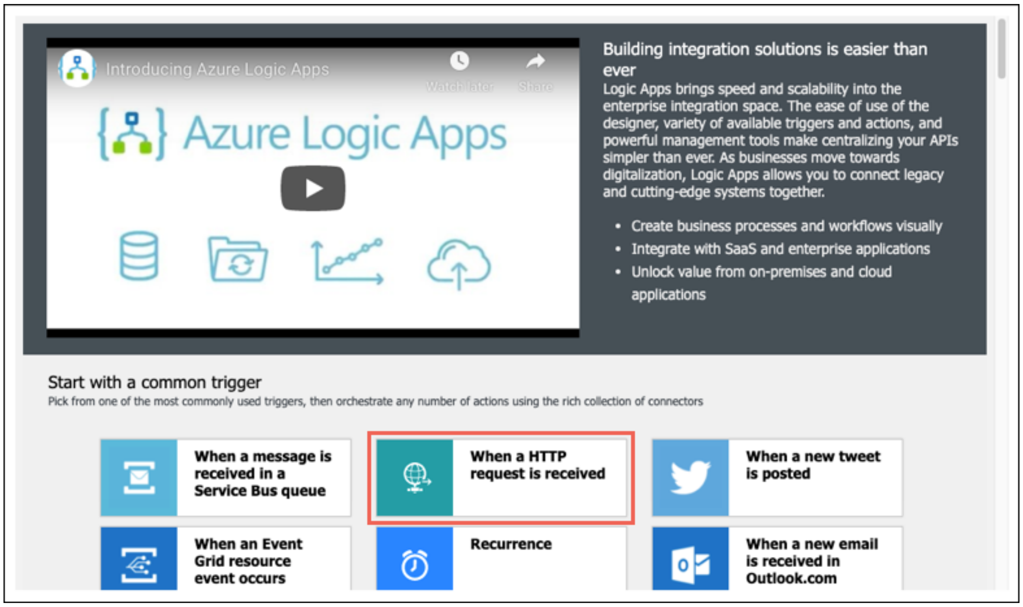
Create the Logic App to send alerts when negative sentiments are received

High Level Overview of the solution



1. In the Azure portal, create an empty logic app by following the instructions in Create your logic app. When you see the **Logic Apps Designer**, return to this tutorial.

2. In the splash page for Logic Apps Designer, select **When an HTTP request is received** under **Start with a common trigger**.



Building integration solutions is easier than ever

Logic Apps brings speed and scalability into the enterprise integration space. The ease of use of the designer, variety of available triggers and actions, and powerful management tools make centralizing your APIs simpler than ever. As businesses move towards digitalization, Logic Apps allows you to connect legacy and cutting-edge systems together.

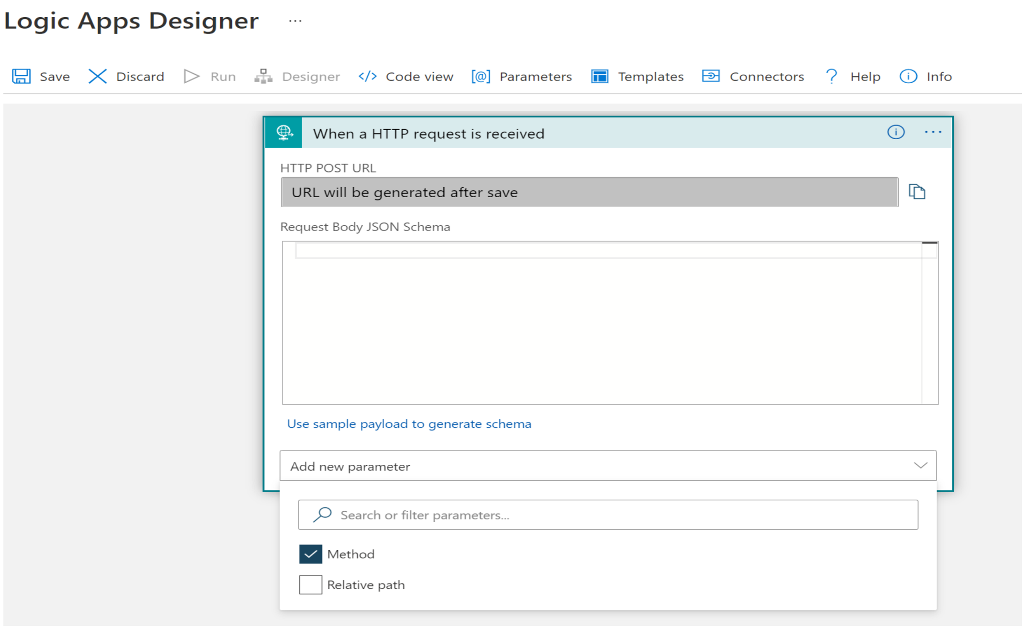
- Create business processes and workflows visually
- Integrate with SaaS and enterprise applications
- Unlock value from on-premises and cloud applications

Start with a common trigger

Pick from one of the most commonly used triggers, then orchestrate any number of actions using the rich collection of connectors

When a message is received in a Service Bus queue
When a HTTP request is received
When a new tweet is posted
When an Event Grid resource event occurs
Recurrence
When a new email is received in Outlook.com

3. Click on Add new parameter and select Method option.



When a HTTP request is received

HTTP POST URL

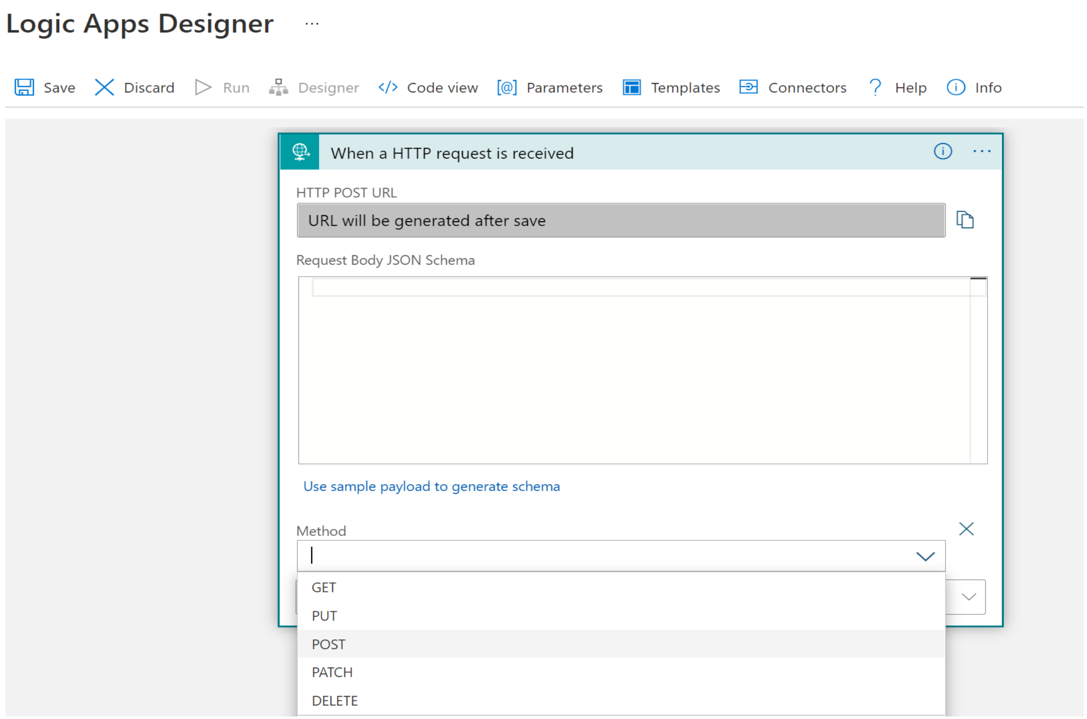
URL will be generated after save

Request Body JSON Schema

☑ Method

□ Relative path

4. Select the Method as POST.



When a HTTP request is received

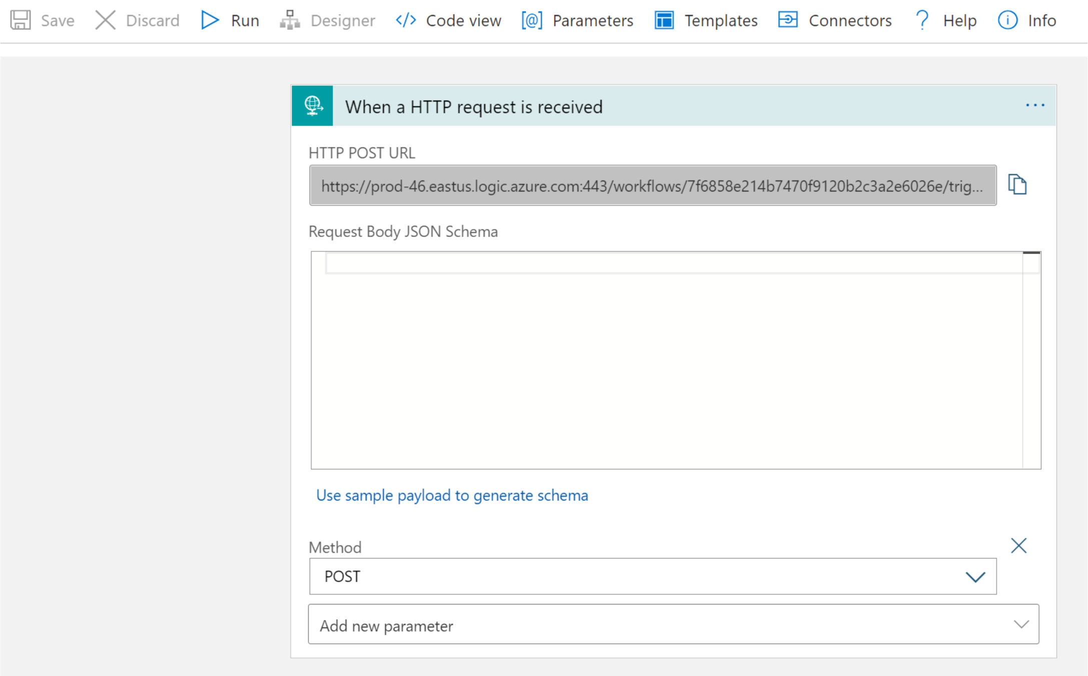
HTTP POST URL

URL will be generated after save

Request Body JSON Schema

Method

5. Save the logic app to complete the HTTP POST request method setup.



HTTP POST URL

https://prod-46.eastus.logic.azure.com:443/workflows/7f6858e214b7470f9120b2c3a2e6026e/trig...

Request Body JSON Schema

Use sample payload to generate schema

Method: POST

Add new parameter

Make a note of the HTTP POST URL

Example:

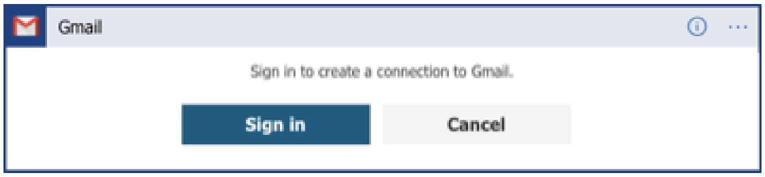
```
https://prod-78.eastus.logic.azure.com:443/workflows/947cc849f439437a923bdb6914ef4ed0/triggers/manual/paths/invoke?api-version=2016-10-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=ymIutpGGLYrRxuocT1hvGqxB-uyAEjQJCQ_DKNM_s3c
```

You'll notice that this HTTP POST URL can be broken up into three distinct pieces:

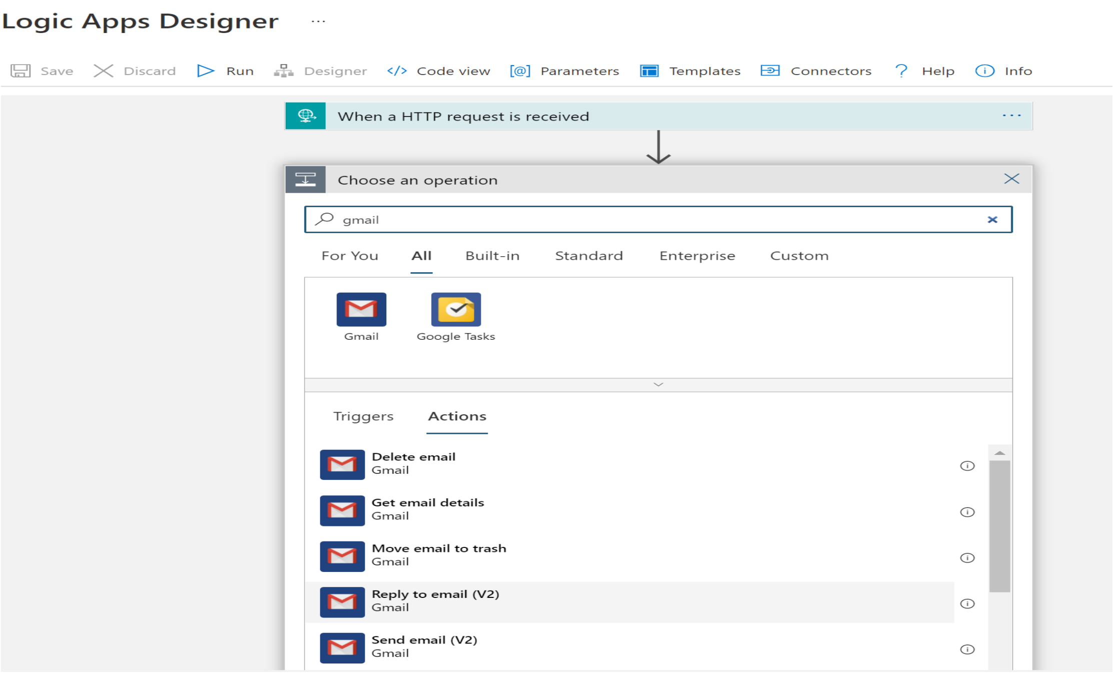
a) HOST: `'prod-78.eastus.logic.azure.com'`
b) PORT: `443`
c) PATH: `'/workflows/947...........DKNM_s3c'`

This becomes important later on when you want to programmatically issue an HTTP POST.

6. At the bottom of the designer, click New step, type Gmail in the actions search box and find and select Send email (V2).

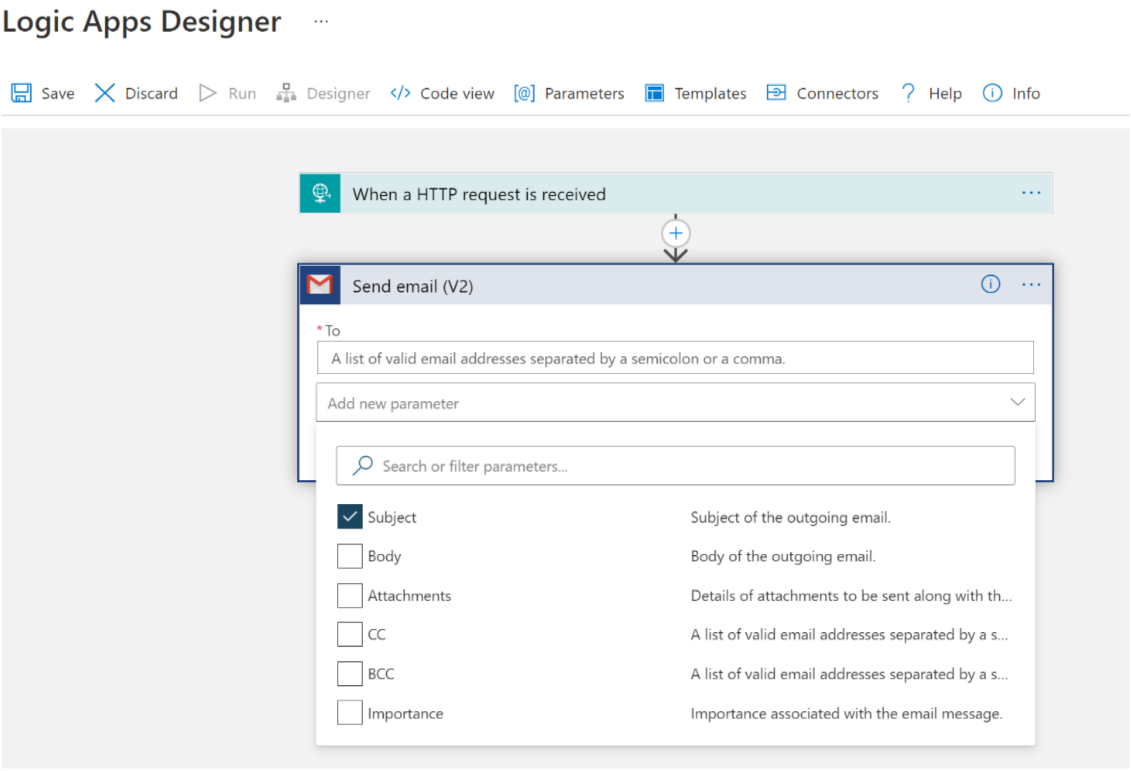


Sign in to create a connection to Gmail.



gmail

7. click on the Add new Parameter box and select the Subject option.

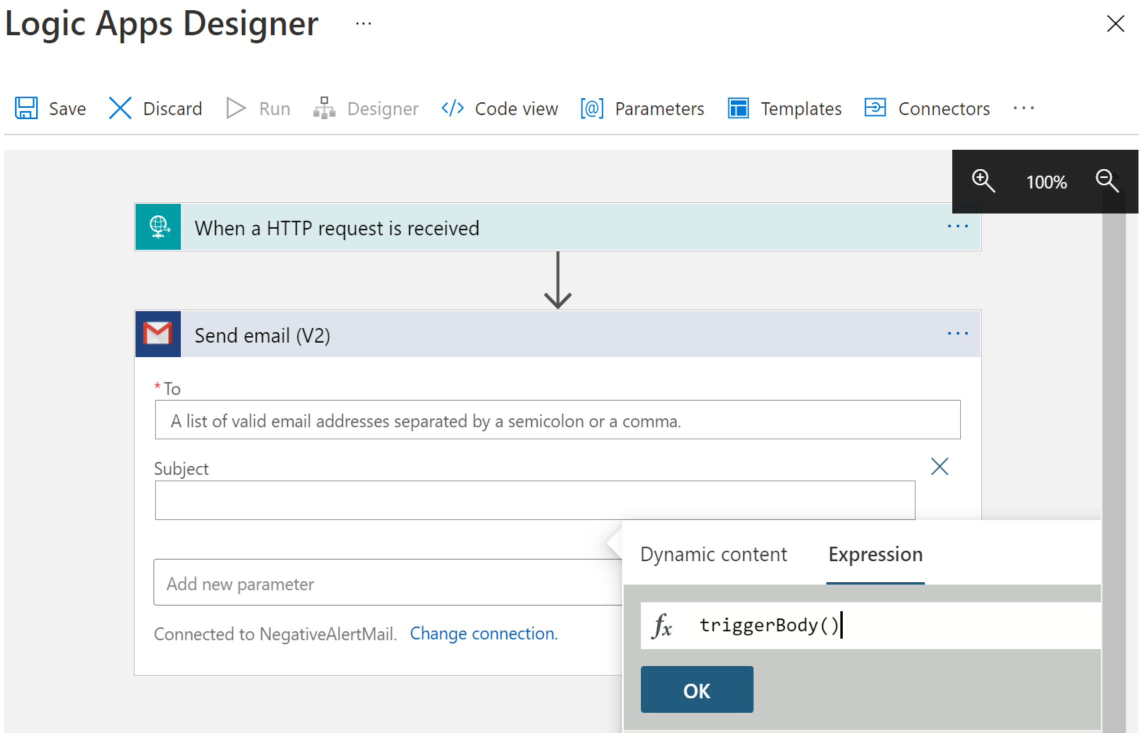


*To

A list of valid email addresses separated by a semicolon or a comma.

☑ Subject	Subject of the outgoing email.
□ Body	Body of the outgoing email.
□ Attachments	Details of attachments to be sent along with th...
□ CC	A list of valid email addresses separated by a s...
□ BCC	A list of valid email addresses separated by a s...
□ Importance	Importance associated with the email message.

8. Click on the Subject textbox, and the Expression content dialog is automatically opened. Search for “triggerBody()” expression and click OK.



When a HTTP request is received

Send email (V2)

* To

A list of valid email addresses separated by a semicolon or a comma.

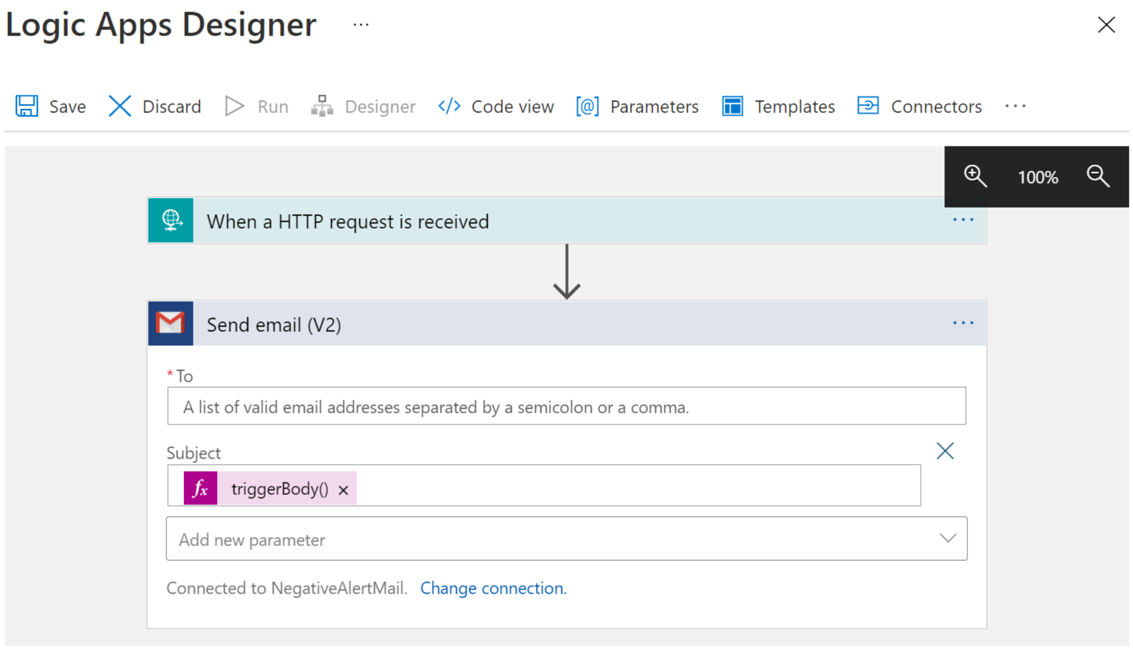
Subject

Add new parameter

Connected to NegativeAlertMail. Change connection.

fx triggerBody()

9. Save the logic app to complete the email setup.



When a HTTP request is received

Send email (V2)

* To

A list of valid email addresses separated by a semicolon or a comma.

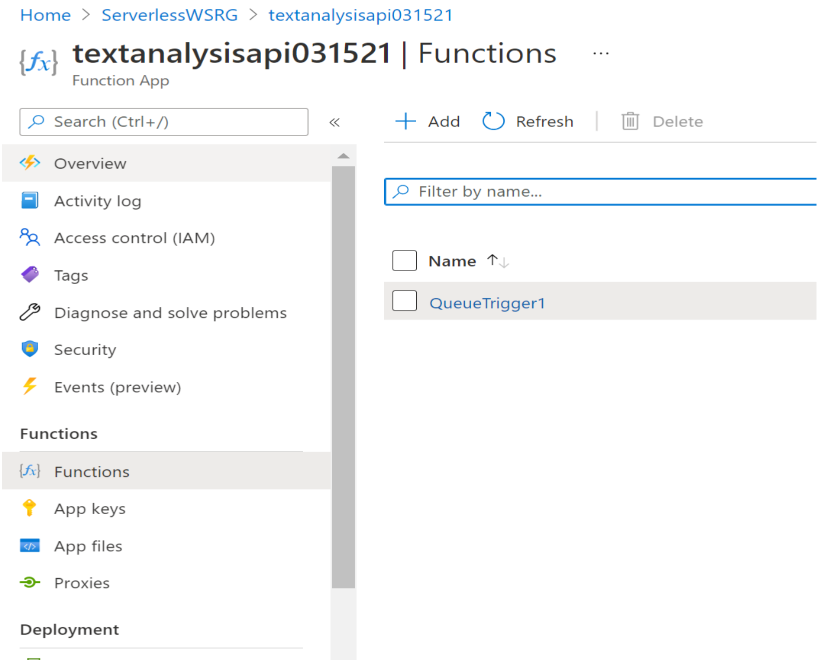
Subject

fx triggerBody() ×

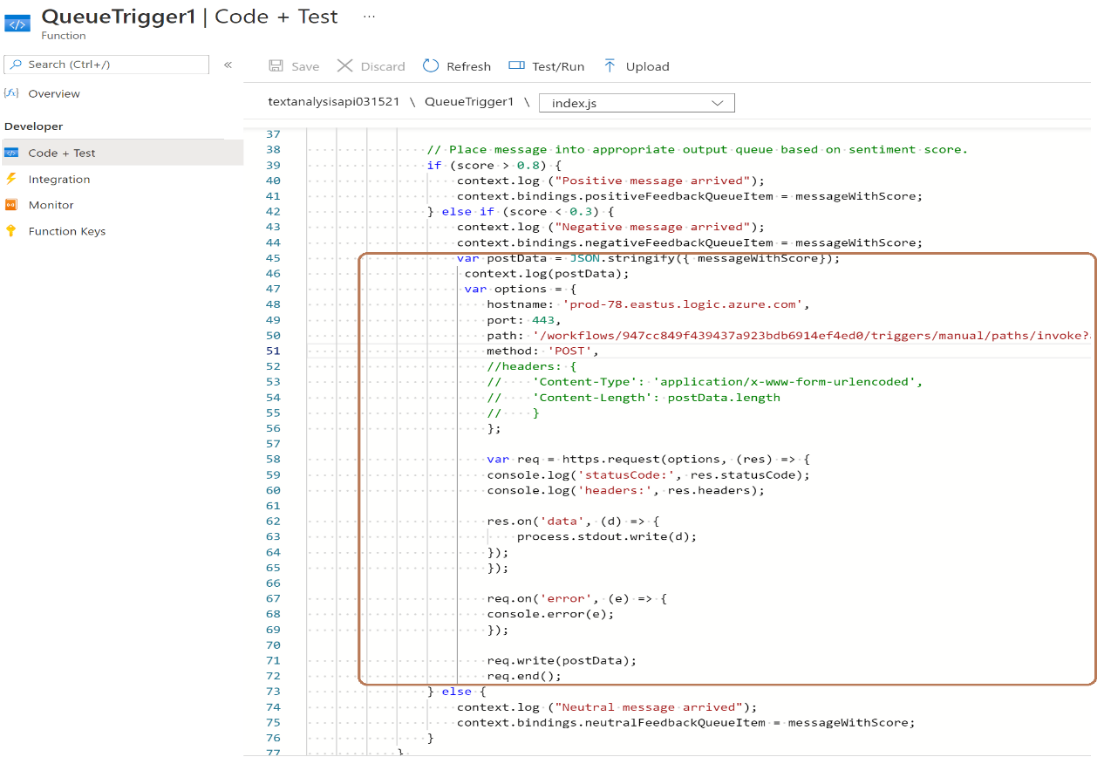
Add new parameter

Connected to NegativeAlertMail. Change connection.

10. Go to your function app.



11. Open the Code editor and paste the logic to send negative sentiment email alert under **“else if (score < 0.3)”** block.



```

            // Place message into appropriate output queue based on sentiment score.
            if (score > 0.8) {
                context.log ("Positive message arrived");
                context.bindings.positiveFeedbackQueueItem = messageWithScore;
            } else if (score < 0.3) {
                context.log ("Negative message arrived");
                context.bindings.negativeFeedbackQueueItem = messageWithScore;
                var postData = JSON.stringify({ messageWithScore});
                 context.log(postData);
                 var options = {
                    hostname: 'prod-78.eastus.logic.azure.com',
                    port: 443,
                    path: '/workflows/947cc849f439437a923bdb6914ef4ed0/triggers/manual/paths/invoke?
                    method: 'POST',
                    //headers: {
                    //    'Content-Type': 'application/x-www-form-urlencoded',
                    //    'Content-Length': postData.length
                    //    }
                    };

                    var req = https.request(options, (res) => {
                    console.log('statusCode:', res.statusCode);
                    console.log('headers:', res.headers);

                    res.on('data', (d) => {
                        process.stdout.write(d);
                    });
                    });

                    req.on('error', (e) => {
                    console.error(e);
                    });

                    req.write(postData);
                    req.end();
            } else {
                context.log ("Neutral message arrived");
                context.bindings.neutralFeedbackQueueItem = messageWithScore;
            }
        }
```

Sample Code Fragment for negative sentiment (ie Score < 0.3):

```
var postData = JSON.stringify({ messageWithScore});
  context.log(postData);
  var options = {
                 hostname: '<YOUR_HOSTNAME_FROM_STEP_5>',
                 port: 443,
                 path: '<YOUR_WORKFLOW_STRING_FROM_STEP_5>',
                 method: 'POST',
                };
   var req = https.request(options, (res) => {
   console.log('statusCode:', res.statusCode);
   console.log('headers:', res.headers);

   res.on('data', (d) => {
       process.stdout.write(d);
   });
});

req.on('error', (e) => {
  console.error(e);
});

req.write(postData);
req.end();
```

Replace the hostname and the path parameters using the information from your logic app HTTP POST method. (For example, here is mine. YOURS IN YELLOW WILL BE DIFFERENT!)

```
var postData = JSON.stringify({ messageWithScore});
context.log(postData);
var options = {
                 hostname: 'prod-78.eastus.logic.azure.com',
                 port: 443,
                 path:
'/workflows/947cc849f439437a923bdb6914ef4ed0/triggers/manual/paths/invoke?api-version=2016-10-01&sp=%2Ft
riggers%2Fmanual%2Frun&sv=1.0&sig=ymIutpGGLYrRxuocT1hvGqxB-uyAEjQJCQ_DKNM_s3c',
                 method: 'POST',
                };
 var req = https.request(options, (res) => {
 console.log('statusCode:', res.statusCode);
 console.log('headers:', res.headers);
 res.on('data', (d) => {
 process.stdout.write(d);
});
});
req.on('error', (e) => {
console.error(e);
});
req.write(postData);
req.end();
```