

Azure Developer Immersion

In this exercise, you will enable users to ask to get an email each time they are mentioned. You do this by adding another API App that exposes a trigger, then creating an Azure Logic App that receives notifications from this trigger and sends an email.

Before we start, you will need to decide which of two possible mechanisms to use for sending email. If you happen to have an Office 365 account (must be a non-consumer one—the kind where you log in with a “work” account rather than a Microsoft account) then it is easiest to use that in the lab. If you do not have an Office 365 account, you can set up a SendGrid account to send mail. However, be aware that newly-created SendGrid accounts go through a kind of quarantine period in which they are monitored for abuse. In this time, emails can take quite a long time to go through.

There are six exercises in this walkthrough:

1. Update VSTS
2. Configure SendGrid
3. Create a Logic App
4. Continue With SendGrid
5. Continue with Office 365
6. Finish Logic App

Exercise 1: Update VSTS

As in previous walkthroughs, you want to update your Team Project’s task board to show what you are going to work on.

1. Your web browser should still be open with a tab for the Visual Studio Team Services Task Board of your Team Project. If not, open your web browser, access your Team Project, and jump to the Task Board.
2. You’ll want to change the **Complete the 03-VSTS Build Walkthrough** task from **To Do** to **Done**.
3. Drag and drop the card **Complete the 04-Logic App Walkthrough** from the **To do** column to the **In progress** column.

Exercise 2: Configure SendGrid

If you are going to use SendGrid, follow these steps to get set up. If you are using Office 365, you do not need to do this.

1. Use your browser to access the Azure portal at <https://portal.azure.com> if it is not already open in a tab.
2. Click the **+ New** button at the top left of the Azure portal.
3. Select **See All** next to **Marketplace** then type **sendgrid** in the search textbox.

4. You should see **SendGrid Email Delivery** as an option. Select it.

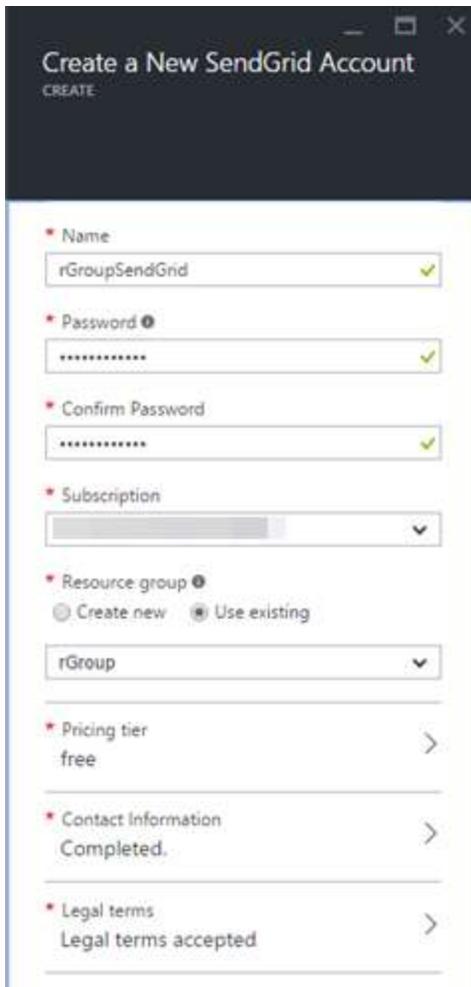
NAME	PUBLISHER	CATEGORY
 SendGrid Email Delivery.	SendGrid	Web + Mobile

5. Click **Create** in the panel that opens.
6. Enter the name **rGroupSendGrid**.
7. Make up a password and enter it in the **Create a New SendGrid Account** panel.

Make sure you make a note of this password you'll need it later.

8. Select the **rGroup** resource group.
9. Under **Pricing tier** select **Free**.
10. Fill in the **Contact Information** section.
11. View the **Legal terms** and click **OK**.

Your configured settings should look like the following:



Create a New SendGrid Account
CREATE

- Name: rGroupSendGrid ✓
- Password: [masked] ✓
- Confirm Password: [masked] ✓
- Subscription: [dropdown]
- Resource group: Create new Use existing
rGroup
- Pricing tier: free >
- Contact information: Completed. >
- Legal terms: Legal terms accepted >

12. Click **Create** at the bottom of the blade.

Azure will create the new SendGrid account. This can take a while, but you can move on to the next steps without waiting for it to finish.

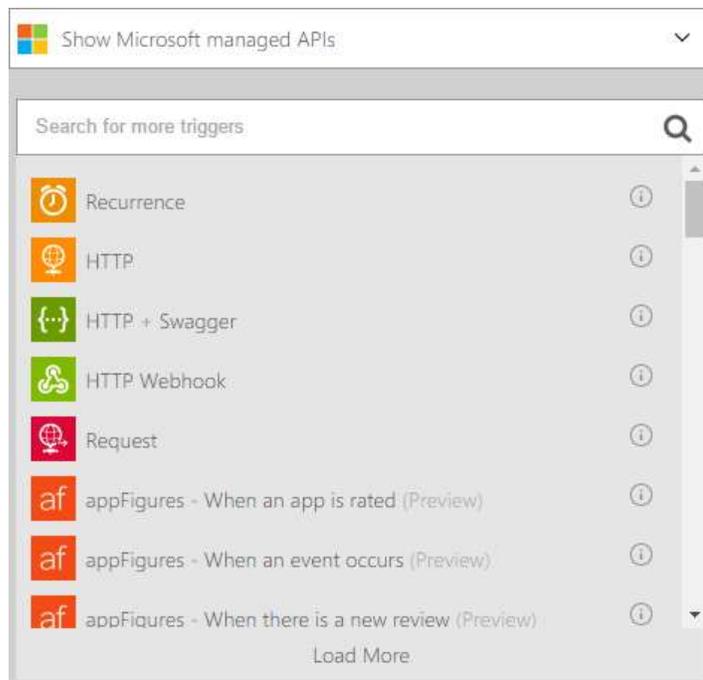
Exercise 3: Create a Logic App

The steps that follow apply regardless of whether you are using SendGrid or Office 365. Logic apps can run workflows in response to various triggers. In our case, we want to trigger work whenever one user mentions another.

1. In the Azure portal click **+ New** at the top left.
2. Select **Web + Mobile** then **Logic App**.
3. In the **Create logic app** pane, name your app **EmailOnMention**.
4. Select the **rGroup** resource group.
5. Click **Create**.

Azure will create the new Logic App. Once it is ready, it should open a blade representing the app. If it does not, open the blade for your **rGroup** resource group and select the **EmailOnMention** logic app.

6. The logic app should open with its **Logic App Designer** view filling most of the web page. Click **Blank Logic App** at the top left of the page.
7. In the middle of the page, it will show a dialog asking you to decide what will trigger the execution of this app's workflow.



8. We're going to use HTTP. Our Logic App will expose an endpoint that a web app can call into each time a user is mentioned. Select the **Request** entry (and not the plain **Http** entry).

9. A box will appear enabling you describe the expected contents of the messages that will be posted to the endpoint. Put the following in the **REQUEST BODY JSON SCHEMA** box:

```
{
  "type": "object",
  "properties": {
    "UserId": {
      "type": "string"
    },
    "Email": {
      "type": "string"
    },
    "MentionedIn": {
      "type": "string"
    },
    "MentionedBy": {
      "type": "string"
    },
    "Message": {
      "type": "string"
    }
  },
  "required": ["UserId", "Email", "MentionedIn", "MentionedBy", "Message"]
}
```

This indicates that each POST must contain a JSON object with 5 string-valued properties with the names shown.

10. The next steps will depend on whether you are sending email through an existing Office365 account or a SendGrid account.

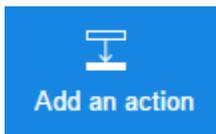
Exercise 4: Connect to Office 365

If you are using Office365, follow these steps. If not, jump to *Exercise 5: Connect to SendGrid*.

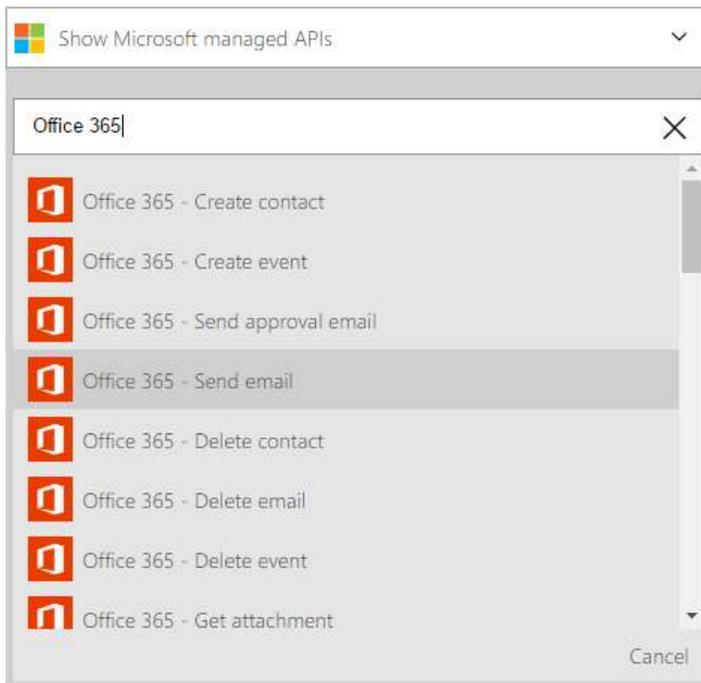
1. In the middle of the designer view, click the plus symbol.



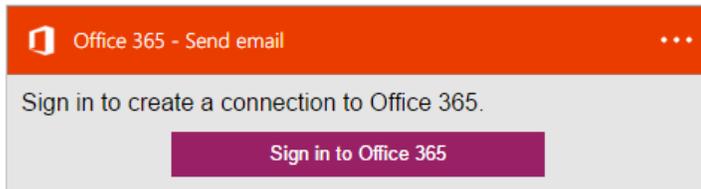
2. Click the **Add an action** button that appears.



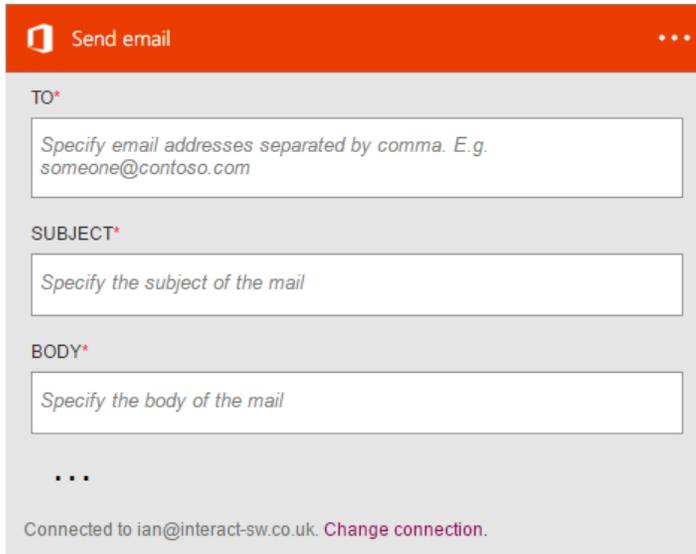
3. A new box will appear asking you what you would like to do next. Type **Office 365** in the search box, and a list of actions will appear, including **Office 365 – Send email**.



4. Select the **Office 365 – Send email** action. A box will appear with a button asking you to sign in to Office 365.



5. Click the **Sign in to Office 365** button. Sign in using the Office 365 account from which you'd like emails to be sent.
6. Once you have signed in, the **Send Email** box will show various fields enabling you to configure the email to be sent.



Send email

TO*

Specify email addresses separated by comma. E.g. someone@contoso.com

SUBJECT*

Specify the subject of the mail

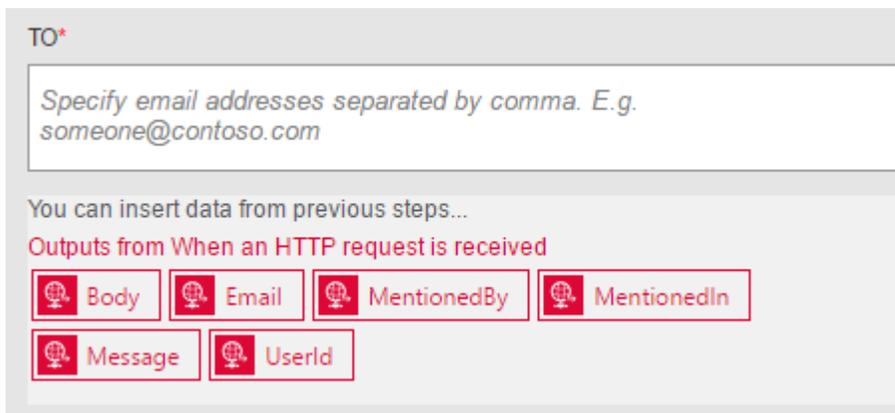
BODY*

Specify the body of the mail

...

Connected to ian@interact-sw.co.uk. [Change connection.](#)

7. Click in the **TO** box. The designer will show various properties from the HTTP input that can be used.



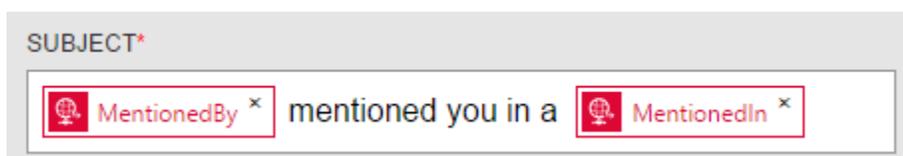
TO*

Specify email addresses separated by comma. E.g. someone@contoso.com

You can insert data from previous steps...

Outputs from When an HTTP request is received

8. Select the **Email** property.
9. Click in the **Subject** box, select **MentionedBy**, then click in the **Subject** box again after the item you just added and type **mentioned you in a** (with spaces before and after this), and then click on **MentionedIn**.



SUBJECT*

mentioned you in a

10. Click in **BODY** and then select **Message**.
11. At the bottom of the box, there are three dots. Click these to show some extra fields.
12. Click in **FROM** and type **donotreply@rgroup.us**.
13. Click the **Save** button at the top of the editor.
14. Skip Exercise 5 and continue to Exercise 6.

Exercise 5: Connect to SendGrid

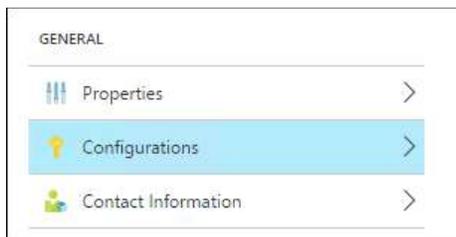
If you are using SendGrid and NOT Office 365, follow these steps.

1. You need your SendGrid user name so open a new tab in your browser.
2. Navigate to <https://portal.azure.com>.

For the **User Name**, you will need to find out what name SendGrid gave you.

This will not be the name you entered when creating the account; that is just the name Azure uses for your SendGrid account in the resource group.

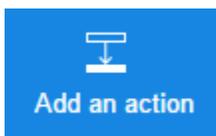
3. You need to go to the **rGroupSendGrid** item which you should find pinned to your home page. If not, it will be a Resource in the **rGroup** resource group.
4. Click the **All Settings** link, then click **Configurations**.



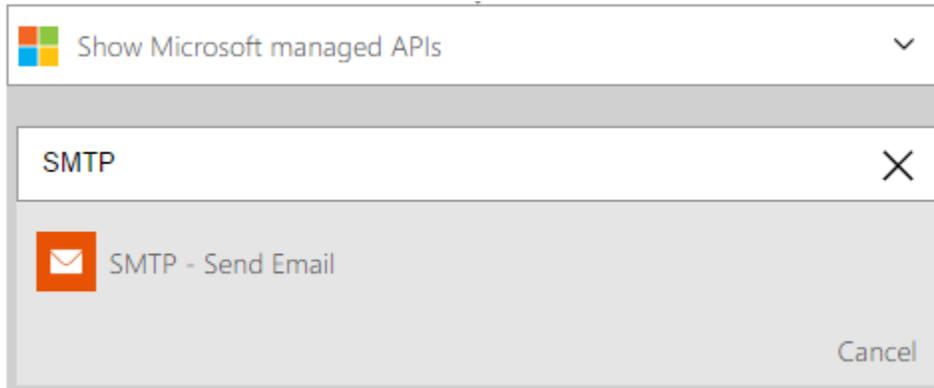
5. From the **Configurations** blade, copy the **USERNAME** field.
6. Switch back to your other Azure tab, which should still be showing your Logic App.
7. In the middle of the designer view, click the **New step** button.



8. Click the **Add an action** button that appears.



- A new box will appear asking you what you would like to do next. Type **SMTP** in the search box. A list of actions will appear, including **SMTP – Send email**.



- Select **SMTP – Send email**. A box will appear with a button asking for details.

A screenshot of the "SMTP - Send Email" configuration dialog box. The title bar is orange and contains an envelope icon, the text "SMTP - Send Email", and a close button (three dots). The main area is light gray and contains several input fields and a checkbox. The fields are: "CONNECTION NAME*" with a placeholder "Enter name for connection"; "SMTP SERVER ADDRESS*" with a placeholder "SMTP Server Address"; "USER NAME*" with a placeholder "User Name"; "PASSWORD*" with a placeholder "Password"; and "SMTP SERVER PORT" with a placeholder "SMTP Port Number (example: 587)". Below these fields is a checkbox labeled "ENABLE SSL?" with the text "Enable SSL? (True/False)" next to it. At the bottom center, there is a "Create" button.

- Enter **SendGrid** as the **CONNECTION NAME**.
- Enter **smtp.sendgrid.net** as the **SMTP SERVER ADDRESS**.
- Enter the SendGrid user name you discovered a few moments ago as the **USER NAME** and the password you specified when creating the SendGrid account as the **PASSWORD**.
- Click the **Create** button.

15. The box will change so that it shows various fields you can use to configure the email to be sent.

16. Click in the **TO** box. The designer will show various properties from the HTTP input that can be used.

17. Select the **Email** property.

18. Click in the **SUBJECT** box, select **MentionedBy**, then click in the **Subject** box again after the item you just added and type **mentioned you in a** (with spaces before and after this), and then click on **MentionedIn**.

19. Click in **BODY** and then select **Message**.

20. At the bottom of the box, click **Show advanced options**.

21. Click in the **FROM** box and type donotreply@rgroup.us.

22. Click the **Save** button at the top of the editor.

23. Continue on.

Exercise 6: Continue with Logic Apps

You will now finish configuration of your Logic App.

1. In the **Logic Apps Designer**, expand the **Request** box and copy the URL in the **HTTP POST TO THIS URL** box. Save this copied URL somewhere (e.g., in a text editor).
2. Return to **Visual Studio**.
3. In the **Rg.Web** project, in the **Operations** folder, open the **UserOperations.cs** file.
4. Near the bottom, you will find a **NotifyMentionsAsync** method.

This already gets called any time any user is mentioned, but it does not do anything yet. Let's fix that.

5. At the top of the file, add:

```
using System.Linq;
using System.Net.Http;
using System.Net.Http.Formatting;
```

6. In the **NotifyMentionsAsync** method, inside the **if** statement you'll find a **// TBD** comment. Replace that with this code:

```
string mentioningUserName = await dbContext.UserInfos
    .Where(u => u.UserInfoId == mentioningUserId)
    .Select(u => u.Name)
    .SingleOrDefault();

using (var client = new HttpClient())
{
    var format = new JsonMediaTypeFormatter();
    foreach (UserInfo mentionedUser in text.MentionsUser)
    {
        if
(mentionedUser.MentionNotificationSettings.HasFlag(NotifyOptions.Email))
        {
            var notification = new MentionNotification
            {
                UserId = mentionedUser.UserInfoId,
                Email = mentionedUser.Email,
                MentionedIn = mentionedIn,
                MentionedBy = mentioningUserName,
                Message = text.Content
            };

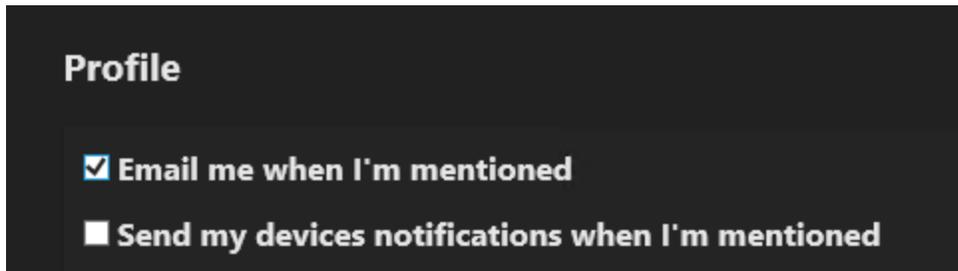
            var request = new HttpRequestMessage(HttpMethod.Post, "LOGIC APP URL
HERE");
            request.Content = new ObjectContent<MentionNotification>(
                notification, format, JsonMediaTypeFormatter.DefaultMediaType);
            await client.SendAsync(request);
        }
    }
}
```

Change the **LOGIC APP URL** text to the URL you copied earlier from your Logic App.

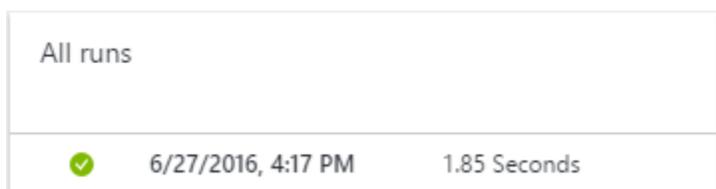
For each user mentioned by a message, this code checks to see if that user has opted into receiving email notifications. If so, it makes an HTTP request to the URL for your Logic App to trigger its execution.

You are going to post messages using the web UI to test this.

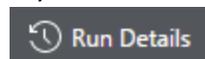
7. In **Solution Explorer**, right-click on **Rg.Web** project node, not the solution, select **Publish**, and click **Publish**.
8. Once the publish is complete, log in to your web application.
9. Expand the menu at the top and click on **Profile**.
10. Ensure the **Email me when I'm mentioned** checkbox is checked.



11. Expand the menu again and click **Home**.
12. In the timeline message entry textbox, type **Hello**, then a space, then the '@' symbol. When you type **@**, a popup will appear listing all the site's users. (If you've only logged in as yourself so far, it will be just you at this point.) Select yourself. Click **Post**.
Right now, you are the only user who has opted into emails. Normally you would not mention yourself, but you are doing this just to test the functionality.
13. Back in the Azure portal, go to your Logic App. You may need to close the designer.
14. You should now see an entry under **All runs**.



15. You can click the entry in this list to get more detail. This is a very simple workflow so it will not say much. You can click the Run Details button for more info.



It just reports that the only operation it performed succeeded, and provides some links to details about the information processed. The first image shows the Office 365 Connector. The second shows SendGrid.

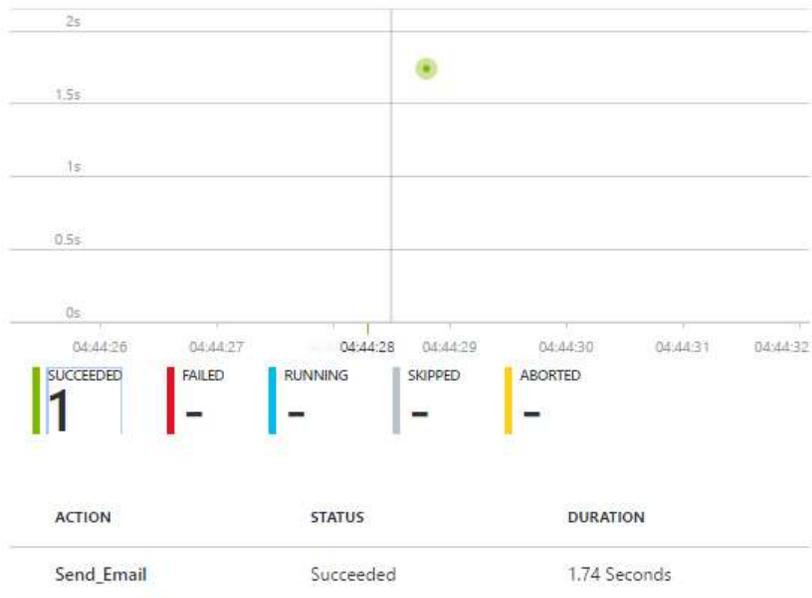


TRIGGER

NAME [manual \(08587345662201099000\)](#)
 OUTPUTS LINK <https://flowprodu05db01.blob.core.windows.net/flow..>

DETAILS

START DATE Monday, June 27, 2016, 4:17:45 PM
 END DATE Monday, June 27, 2016, 4:17:47 PM
 CORRELATION ID 07da652e-768a-4d6e-a7f9-d21ecc22c864 



TRIGGER

NAME [manual \(08587345646167880801\)](#)
 OUTPUTS LINK <https://flowprodcu05db01.blob.core.windows.net/flow...>

DETAILS

START DATE Monday, June 27, 2016, 4:44:28 PM
 END DATE Monday, June 27, 2016, 4:44:30 PM
 CORRELATION ID 
 OUTPUTS No outputs

16. You can click on that action to see further details. Click on the links and you will be able to see the subject, content, and recipient that were passed in, as well as detailed information about the result of the network invocation it performed to send the email.
17. If you have not already, you should soon receive an email. It could take a while if you are using SendGrid because your new account is not yet fully trusted.

Note check your Junk / Spam folders.

18. While you wait, like you've done before, commit and push your changes to Visual Studio Team Services from Visual Studio, making sure to associate your work item. Use a commit message like **Add code to integrate with Logic App and support e-mail notifications.**
19. Add your work item as before and then **Commit All and Push.**
20. In Visual Studio Team Services, mark your task as **Done.**
 Let's review where you are.

You have created a Logic App that uses one of the built-in API adapters (either Office365 or SMTP, depending on which way you did this). You then invoked this logic app from your web app to send emails to users who have asked to be notified when they are mentioned. To keep things quick, this was a pretty simple Logic App, but you can create more advanced workflows involving multiple actions and various flow control constructs to perform more complex work.

Last Updated: September 23, 2016.