Azure Functions SendGrid bindings

2017-3-16 • 1 min to read • Contributors

In this article

This article explains how to configure and work with SendGrid bindings in Azure Functions. With SendGrid, you can use Azure Functions to send customized email programmatically.

This article is reference information for Azure Functions developers. If you're new to Azure Functions, start with the following resources:

Create your first Azure Function. C#, F#, or Node developer references.

function.json for SendGrid bindings

Azure Functions provides an output binding for SendGrid. The SendGrid output binding enables you to create and send email programmatically.

The SendGrid binding supports the following properties:

- `name` : Required - the variable name used in function code for the request or request body. This value is `$return` when there is only one return value.
- `type` : Required - must be set to "sendGrid."
- `direction` : Required - must be set to "out."
- `apiKey` : Required - must be set to the name of your API key stored in the Function App's app settings.
- `to` : the recipient's email address.
- `from` : the sender's email address.
- `subject` : the subject of the email.
- `text` : the email content.

Example of **function.json**:

JSON Copy

```json
{
    "bindings": [
        {
            "name": "$return",
            "type": "sendGrid",
            "direction": "out",
            "apiKey" : "MySendGridKey",
            "to": "{ToEmail}",
            "from": "{FromEmail}",
            "subject": "SendGrid output bindings"
        }
    ]
}
```

Note

Azure Functions stores your connection information and API keys as app settings so that they are not checked into your source control repository. This action safeguards your sensitive information.

C# example of the SendGrid output binding

C# Copy

```
#r "SendGrid"
using System;
using SendGrid.Helpers.Mail;

public static Mail Run(TraceWriter log, string input, out Mail message)
{
     message = new Mail
    {
        Subject = "Azure news"
    };

    var personalization = new Personalization();
    personalization.AddTo(new Email("recipient@contoso.com"));

    Content content = new Content
    {
        Type = "text/plain",
        Value = input
    };
    message.AddContent(content);
    message.AddPersonalization(personalization);
}
```

Node example of the SendGrid output binding

JavaScript Copy

```
module.exports = function (context, input) {
    var message = {
        to: "recipient@contoso.com",
        from: "sender@contoso.com",
        subject: "Azure news",
        content: [{
            type: 'text/plain',
            value: input
        }]
    };

    context.done(null, message);
};
```

Next steps

For information about other bindings and triggers for Azure Functions, see

- Azure Functions triggers and bindings developer reference
- Best practices for Azure Functions Lists some best practices to use when creating Azure Functions.
- Azure Functions developer reference Programmer reference for coding functions and defining triggers and bindings.