Add-ons (/categories/add-ons) All Add-ons (/categories/add-on-documentation) SendGrid (/articles/sendgrid)

This add-on is operated by SendGrid

Email Delivery. Simplified.

SendGrid

Last updated 09 October 2019

Table of Contents

SendGrid (https://elements.heroku.com/addons/sendgrid) is an add-on (https://elements.heroku.com/addons) for providing scalable email delivery and analytics for apps.

SendGrid's cloud-based email infrastructure relieves businesses of the cost and complexity of maintaining custom email systems. SendGrid provides reliable delivery, scalability and real-time analytics along with flexible APIs that make custom integration a breeze.

Provisioning the add-on

SendGrid can be attached to a Heroku application via the CLI:

A list of all plans available can be found **here (https://elements.heroku.com/addons/sendgrid)** .

```
$ heroku addons:create sendgrid:starter
-----> Adding sendgrid:starter to sharp-mountain-4005... done, v18 (free)
```

Once SendGrid has been added a `SENDGRID_USERNAME` , `SENDGRID_PASSWORD` settings will be available in the app configuration and will contain the credentials used to access the newly provisioned SendGrid service instance. This can be confirmed using the `heroku config:get` command.

```
$ heroku config:get SENDGRID_USERNAME
appXYZ@heroku.com

$ heroku config:get SENDGRID_PASSWORD
password
```

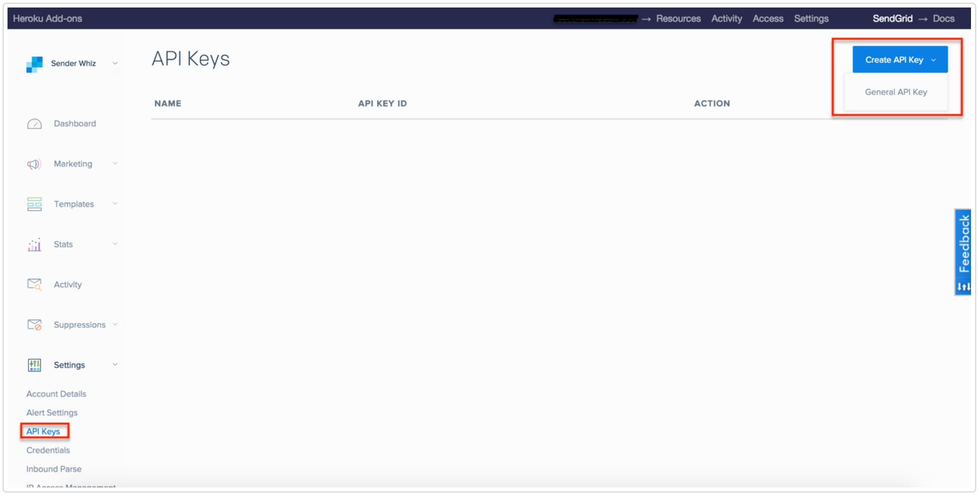
Obtaining an API key

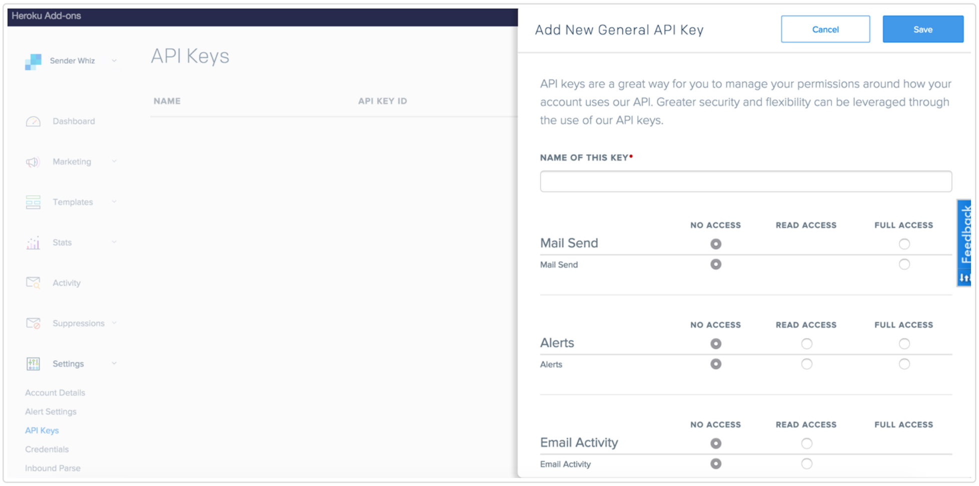
API Keys are used by your application, mail client, or website to authenticate access to SendGrid services. They are the preferred alternative to using a username and password because you can revoke an API key at any time without having to change your username and password. We suggest that you use API keys for connecting to all of SendGrid's services.

Create an API key

Go here (https://app.sendgrid.com/settings/api_keys). When you click the “Create API Key” button, a dropdown menu will appear allowing you to choose the type of API Key you would like to create. After selecting “General API Key” you will be shown a page allowing you to give your new key a name and permissions.

The API Key name will follow your API key around through the SendGrid customer portal, so it is important that you choose a name that is meaningful to you.





API keys are a great way for you to manage your permissions around how your account uses our API. Greater security and flexibility can be leveraged through the use of our API keys.

Full documentation can be found here (https://sendgrid.com/docs/User_Guide/Settings/api_keys.html).

After installing SendGrid and creating an API Key the application should be configured to fully integrate with the add-on.

Setup API key environment variable

Use the Heroku CLI’s config commands to add your API Key to be used by your applications:

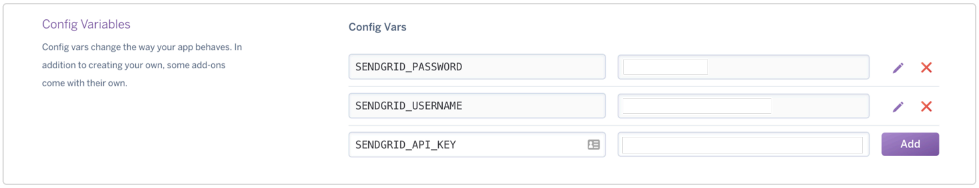
```
$ heroku config:set SENDGRID_API_KEY=xxxx_api_key_xxxx
Adding config vars and restarting myapp... done, v12
SENDGRID_API_KEY: xxxx_api_key_xxxx

$ heroku config
SENDGRID_API_KEY: xxxx_api_key_xxxx
OTHER_VAR:    production

$ heroku config:get SENDGRID_API_KEY
xxxx_api_key_xxxx

$ heroku config:unset SENDGRID_API_KEY
Unsetting SENDGRID_API_KEY and restarting myapp... done, v13
```

You can also edit config vars on your app’s settings tab on Dashboard:


Config Variables
Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.
Config Vars
Add

```
SENDGRID_PASSWORD
SENDGRID_USERNAME
SENDGRID_API_KEY
```

Golang

SendGrid has a Golang helper library that facilitates the adoption of SendGrid in Golang applications. The module's source code can be found at GitHub (http://github.com/sendgrid/sendgrid-go).

Install Package

Install sendgrid locally with the following command.

```
$ go get github.com/sendgrid/sendgrid-go
```

Example

See here how to add an API-KEY in your App Environment .

The following is the minimum needed code to send an email with the /mail/send Helper (https://github.com/sendgrid/sendgrid-go/tree/master/helpers/mail) (here (https://github.com/sendgrid/sendgrid-go/blob/master/examples/helpers/mail/example.go#L32) is a full example):

With Mail Helper Class

```
package main

import (
    "fmt"
    "github.com/sendgrid/sendgrid-go"
    "github.com/sendgrid/sendgrid-go/helpers/mail"
    "os"
)

func main() {
    from := mail.NewEmail("Example User", "test@example.com")
    subject := "Hello World from the SendGrid Go Library"
    to := mail.NewEmail("Example User", "test@example.com")
    content := mail.NewContent("text/plain", "some text here")
    m := mail.NewV3MailInit(from, subject, to, content)

    request := sendgrid.GetRequest(os.Getenv("SENDGRID_API_KEY"), "/v3/mail/send", "https://api.sendgrid.com"
    request.Method = "POST"
    request.Body = mail.GetRequestBody(m)
    response, err := sendgrid.API(request)
    if err != nil {
        fmt.Println(err)
    } else {
        fmt.Println(response.StatusCode)
        fmt.Println(response.Body)
        fmt.Println(response.Headers)
    }
}
```

The `NewV3MailInit` constructor creates a personalization object (https://sendgrid.com/docs/Classroom/Send/v3_Mail_Send/personalizations.html) for you. Here (https://github.com/sendgrid/sendgrid-go/blob/master/examples/helpers/mail/example.go#L28) is an example of how to add to it.

Without Mail Helper Class

The following is the minimum needed code to send an email without the /mail/send Helper (here (https://github.com/sendgrid/sendgrid-go/blob/master/examples/mail/mail.go#L47) is a full example):

```
package main

import (
    "fmt"
    "github.com/sendgrid/sendgrid-go"
    "os"
)

func main() {
    request := sendgrid.GetRequest(os.Getenv("SENDGRID_API_KEY"), "/v3/mail/send", "https://api.sendgrid.com"
    request.Method = "POST"
    request.Body = []byte(` {
    "personalizations": [
        {
            "to": [
                {
                    "email": "test@example.com"
                }
            ],
            "subject": "Hello World from the SendGrid Go Library!"
        }
    ],
    "from": {
        "email": "test@example.com"
    },
    "content": [
        {
            "type": "text/plain",
            "value": "Hello, Email!"
        }
    ]
}`)
    response, err := sendgrid.API(request)
    if err != nil {
        fmt.Println(err)
    } else {
        fmt.Println(response.StatusCode)
        fmt.Println(response.Body)
        fmt.Println(response.Headers)
    }
}
```

Full documentation of all the features of SendGrid's Go helper library can be found on GitHub (http://github.com/sendgrid/sendgrid-go).

Java

SendGrid has a Java library that facilitates the adoption of SendGrid in Java applications. The module's source code can be found at GitHub (http://github.com/sendgrid/sendgrid-java).

Install Package

Choose your installation method - Maven w/ Gradle (recommended), Maven or Jar file.

via Maven w/ Gradle

Add the following to your build.gradle file in the root of your project.

```
...
dependencies {
  ...
  compile 'com.sendgrid:sendgrid-java:3.0.9'
}

repositories {
  mavenCentral()
}
...
```

Then import the library - in the file appropriate to your Java project.

```
import com.sendgrid.SendGrid;
```

via Maven

```
mvn install
```

via jar file

You can just drop the jar file in. It's a fat jar - it has all the dependencies built in.

sendgrid-java.jar (http://repo1.maven.org/maven2/com/sendgrid/sendgrid-java/3.0.9/sendgrid-java-3.0.9-jar.jar)

Example

See here how to add an API-KEY in your App Environment .

The following is the minimum needed code to send an email with the /mail/send Helper (https://github.com/sendgrid/sendgrid-java/tree/master/src/main/java/com/sendgrid/helpers) (here (https://github.com/sendgrid/sendgrid-java/blob/master/examples/helpers/mail/Example.java#L30) is a full example):

With Mail Helper Class

```
import com.sendgrid.*;
import java.io.IOException;

public class Example {
  public static void main(String[] args) throws IOException {
    Email from = new Email("test@example.com");
    String subject = "Hello World from the SendGrid Java Library!";
    Email to = new Email("test@example.com");
    Content content = new Content("text/plain", "Hello, Email!");
    Mail mail = new Mail(from, subject, to, content);

    SendGrid sg = new SendGrid(System.getenv("SENDGRID_API_KEY"));
    Request request = new Request();
    try {
      request.method = Method.POST;
      request.endpoint = "mail/send";
      request.body = mail.build();
      Response response = sg.api(request);
      System.out.println(response.statusCode);
      System.out.println(response.body);
      System.out.println(response.headers);
    } catch (IOException ex) {
      throw ex;
    }
  }
}
```

See here how to add an API-K

The `Mail` constructor creates a personalization object (https://sendgrid.com/docs/Classroom/Send/v3_Mail_Send/personalizations.html) for you. Here (https://github.com/sendgrid/sendgrid-java/blob/master/examples/helpers/mail/Example.java#L221) is an example of how to add to it.

Without Mail Helper Class

The following is the minimum needed code to send an email without the /mail/send Helper (here (https://github.com/sendgrid/sendgrid-java/blob/master/examples/mail/mail.java#L54) is a full example):

```
import com.sendgrid.*;
import java.io.IOException;

public class Example {
  public static void main(String[] args) throws IOException {
    try {
      SendGrid sg = new SendGrid(System.getenv("SENDGRID_API_KEY"));
      Request request = new Request();
      request.method = Method.POST;
      request.endpoint = "mail/send";
      request.body = "{\"personalizations\":[{\"to\":[{\"email\":\"test@example.com\"}],\"subject\":\"Hello W
      Response response = sg.api(request);
      System.out.println(response.statusCode);
      System.out.println(response.body);
      System.out.println(response.headers);
    } catch (IOException ex) {
      throw ex;
    }
  }
}
```

Full documentation of all the features of SendGrid's Java module can be found on GitHub (http://github.com/sendgrid/sendgrid-java).

Node.js

SendGrid has a Node.js package that facilitates the adoption of SendGrid in Node.js applications. The package's source code can be found at GitHub (http://github.com/sendgrid/sendgrid-nodejs).

Install Package

The following recommended installation requires npm (https://npmjs.org/). If you are unfamiliar with npm, see the npm docs (https://npmjs.org/doc/). Npm comes installed with Node.js since node version 0.8.x therefore you likely already have it.

Add the following to your `package.json` file:

```
{
  ...
  "dependencies": {
    ...
    "sendgrid": "^4.2.0"
  }
}
```

Install sendgrid-nodejs and its dependencies:

```
$ npm install
```

Alternative installation

You can also install sendgrid locally with the following command:

```
$ npm install sendgrid
```

Example

See here how to add an API-KEY in your App Environment .

The following is the minimum needed code to send an email with the /mail/send Helper (https://github.com/sendgrid/sendgrid-nodejs/tree/master/lib/helpers/mail) (here (https://github.com/sendgrid/sendgrid-nodejs/blob/master/examples/helpers/mail/example.js#L15) is a full example):

With Mail Helper Class

```
var helper = require('sendgrid').mail;
var from_email = new helper.Email('test@example.com');
var to_email = new helper.Email('test@example.com');
var subject = 'Hello World from the SendGrid Node.js Library!';
var content = new helper.Content('text/plain', 'Hello, Email!');
var mail = new helper.Mail(from_email, subject, to_email, content);

var sg = require('sendgrid')(process.env.SENDGRID_API_KEY);
var request = sg.emptyRequest({
  method: 'POST',
  path: '/v3/mail/send',
  body: mail.toJSON(),
});

sg.API(request, function(error, response) {
  console.log(response.statusCode);
  console.log(response.body);
  console.log(response.headers);
});
```

The `Mail` constructor creates a personalization object (https://sendgrid.com/docs/Classroom/Send/v3_Mail_Send/personalizations.html) for you. Here (https://github.com/sendgrid/sendgrid-nodejs/blob/master/examples/helpers/mail/example.js#L10) is an example of how to add to it.

Without Mail Helper Class

The following is the minimum needed code to send an email without the /mail/send Helper (here (https://github.com/sendgrid/sendgrid-nodejs/blob/master/examples/mail/mail.js#L31) is a full example):

```
var sg = require('sendgrid')(process.env.SENDGRID_API_KEY);
var request = sg.emptyRequest({
  method: 'POST',
  path: '/v3/mail/send',
  body: {
    personalizations: [
      {
        to: [
          {
            email: 'test@example.com',
          },
        ],
        subject: 'Hello World from the SendGrid Node.js Library!',
      },
    ],
    from: {
      email: 'test@example.com',
    },
    content: [
      {
        type: 'text/plain',
        value: 'Hello, Email!',
      },
    ],
  },
});

//With promise
sg.API(request)
  .then(response => {
    console.log(response.statusCode);
    console.log(response.body);
    console.log(response.headers);
  })
  .catch(error => {
    //error is an instance of SendGridError
    //The full response is attached to error.response
    console.log(error.response.statusCode);
  });

//With callback
sg.API(request, function(error, response) {
  if (error) {
    console.log('Error response received');
  }
  console.log(response.statusCode);
  console.log(response.body);
  console.log(response.headers);
});
```

Full documentation of all the features of SendGrid's Node.js module can be found on GitHub (http://github.com/sendgrid/sendgrid-nodejs).

PHP

SendGrid has a PHP library that facilitates the adoption of SendGrid in PHP applications. The module's source code can be found at GitHub (http://github.com/sendgrid/sendgrid-php).

Install package

Add SendGrid to your `composer.json` file. If you are not using Composer (http://getcomposer.org), you should be. It's an excellent way to manage dependencies in your PHP application.

```
{
  "require": {
    "sendgrid/sendgrid": "~5.0.9"
  }
}
```

Then at the top of your PHP script require the autoloader:

```
require 'vendor/autoload.php';
```

Alternative: Install package from zip

If you are not using Composer, simply download and install the **latest packaged release of the library as a zip (https://sendgrid-open-source.s3.amazonaws.com/sendgrid-php/sendgrid-php.zip)**.

⬇ Download Packaged Library ⬇ (https://sendgrid-open-source.s3.amazonaws.com/sendgrid-php/sendgrid-php.zip)

Then require the library from package:

```
require("path/to/sendgrid-php/sendgrid-php.php");
```

Previous versions of the library can be found in the version index (https://sendgrid-open-source.s3.amazonaws.com/index.html).

Example

See here how to add an API-KEY in your App Environment .

The following is the minimum needed code to send an email with the /mail/send Helper (https://github.com/sendgrid/sendgrid-php/tree/master/lib/helpers/mail) (here (https://github.com/sendgrid/sendgrid-php/blob/master/examples/helpers/mail/example.php#L22) is a full example):

With Mail Helper Class

```
<?php
// If you are using Composer
require 'vendor/autoload.php';

// If you are not using Composer (recommended)
// require("path/to/sendgrid-php/sendgrid-php.php");

$from = new SendGrid\Email(null, "test@example.com");
$subject = "Hello World from the SendGrid PHP Library!";
$to = new SendGrid\Email(null, "test@example.com");
$content = new SendGrid\Content("text/plain", "Hello, Email!");
$mail = new SendGrid\Mail($from, $subject, $to, $content);

$apiKey = getenv('SENDGRID_API_KEY');
$sg = new \SendGrid($apiKey);

$response = $sg->client->mail()->send()->post($mail);
echo $response->statusCode();
echo $response->headers();
echo $response->body();
```

The `SendGrid\Mail` constructor creates a personalization object (https://sendgrid.com/docs/Classroom/Send/v3_Mail_Send/personalizations.html) for you. Here (https://github.com/sendgrid/sendgrid-php/blob/master/examples/helpers/mail/example.php#L16) is an example of how to add to it.

Without Mail Helper Class

The following is the minimum needed code to send an email without the /mail/send Helper (here (https://github.com/sendgrid/sendgrid-php/blob/master/examples/mail/mail.php#L28) is a full example):

```php
<?php
// If you are using Composer
require 'vendor/autoload.php';

// If you are not using Composer (recommended)
// require("path/to/sendgrid-php/sendgrid-php.php");

$request_body = json_decode('{
  "personalizations": [
    {
      "to": [
        {
          "email": "test@example.com"
        }
      ],
      "subject": "Hello World from the SendGrid PHP Library!"
    }
  ],
  "from": {
    "email": "test@example.com"
  },
  "content": [
    {
      "type": "text/plain",
      "value": "Hello, Email!"
    }
  ]
}');

$apiKey = getenv('SENDGRID_API_KEY');
$sg = new \SendGrid($apiKey);

$response = $sg->client->mail()->send()->post($request_body);
echo $response->statusCode();
echo $response->body();
echo $response->headers();
```

Full documentation of all the features of SendGrid's PHP library can be found on GitHub (http://github.com/sendgrid/sendgrid-php).

Python

SendGrid has a Python module that facilitates the adoption of SendGrid in Python applications. The module's source code can be found at GitHub (http://github.com/sendgrid/sendgrid-python).

Install Package

```
$ pip install sendgrid
```

Example

See here how to add an API-KEY in your App Environment .

The following is the minimum needed code to send an email with the /mail/send Helper (https://github.com/sendgrid/sendgrid-python/tree/master/sendgrid/helpers/mail) (here (https://github.com/sendgrid/sendgrid-python/blob/master/examples/helpers/mail/mail_example.py#L20) is a full example):

With Mail Helper Class

```python
import sendgrid
import os
from sendgrid.helpers.mail import *

sg = sendgrid.SendGridAPIClient(apikey=os.environ.get('SENDGRID_API_KEY'))
from_email = Email("test@example.com")
subject = "Hello World from the SendGrid Python Library!"
to_email = Email("test@example.com")
content = Content("text/plain", "Hello, Email!")
mail = Mail(from_email, subject, to_email, content)
response = sg.client.mail.send.post(request_body=mail.get())
print(response.status_code)
print(response.body)
print(response.headers)
```

The `Mail` constructor creates a personalization object (https://sendgrid.com/docs/Classroom/Send/v3_Mail_Send/personalizations.html) for you. Here (https://github.com/sendgrid/sendgrid-python/blob/master/examples/helpers/mail/mail_example.py#L16) is an example of how to add to it.

Without Mail Helper Class

The following is the minimum needed code to send an email without the /mail/send Helper (here (https://github.com/sendgrid/sendgrid-python/blob/master/examples/mail/mail.py#L27) is a full example):

```
import sendgrid
import os

sg = sendgrid.SendGridAPIClient(apikey=os.environ.get('SENDGRID_API_KEY'))
data = {
  "personalizations": [
    {
      "to": [
        {
          "email": "test@example.com"
        }
      ],
      "subject": "Hello World from the SendGrid Python Library!"
    }
  ],
  "from": {
    "email": "test@example.com"
  },
  "content": [
    {
      "type": "text/plain",
      "value": "Hello, Email!"
    }
  ]
}
response = sg.client.mail.send.post(request_body=data)
print(response.status_code)
print(response.body)
print(response.headers)
```

Full documentation of all the features of SendGrid's Python module can be found on GitHub (http://github.com/sendgrid/sendgrid-python).

Ruby

SendGrid has a Ruby gem that facilitates the adoption of SendGrid in Ruby applications. The module's source code can be found at GitHub (http://github.com/sendgrid/sendgrid-ruby).

Install Package

Add this line to your application's Gemfile:

```
$ gem 'sendgrid-ruby'
```

And then execute:

```
$ bundle
```

Or install it yourself using:

```
$ gem install sendgrid-ruby
```

Example

See here how to add an API-KEY in your App Environment .

The following is the minimum needed code to send an email with the /mail/send Helper (https://github.com/sendgrid/sendgrid-ruby/tree/master/lib/sendgrid/helpers/mail) (here (https://github.com/sendgrid/sendgrid-ruby/blob/master/examples/helpers/mail/example.rb#L21) is a full example):

With Mail Helper Class

```
require 'sendgrid-ruby'
include SendGrid

from = Email.new(email: 'test@example.com')
subject = 'Hello World from the SendGrid Ruby Library!'
to = Email.new(email: 'test@example.com')
content = Content.new(type: 'text/plain', value: 'Hello, Email!')
mail = Mail.new(from, subject, to, content)

sg = SendGrid::API.new(api_key: ENV['SENDGRID_API_KEY'])
response = sg.client.mail._('send').post(request_body: mail.to_json)
puts response.status_code
puts response.body
puts response.headers
```

For more complex scenarios, please do not use the above constructor and instead build your own personalization object as demonstrated here (https://github.com/sendgrid/sendgrid-ruby/blob/master/examples/helpers/mail/example.rb#L21).

Without Mail Helper Class

The following is the minimum needed code to send an email without the /mail/send Helper (here (https://github.com/sendgrid/sendgrid-ruby/blob/master/examples/mail/mail.rb#L26) is a full example):

```
require 'sendgrid-ruby'
include SendGrid

data = JSON.parse('{
  "personalizations": [
    {
      "to": [
        {
          "email": "test@example.com"
        }
      ],
      "subject": "Hello World from the SendGrid Ruby Library!"
    }
  ],
  "from": {
    "email": "test@example.com"
  },
  "content": [
    {
      "type": "text/plain",
      "value": "Hello, Email!"
    }
  ]
}')
sg = SendGrid::API.new(api_key: ENV['SENDGRID_API_KEY'])
response = sg.client.mail._("send").post(request_body: data)
puts response.status_code
puts response.body
puts response.headers
```

ActionMailer

Configure ActionMailer to Use SendGrid

In config/environment.rb specify your ActionMailer settings to point to SendGrid's servers.

```
ActionMailer::Base.smtp_settings = {
  :user_name => ENV['SENDGRID_USERNAME'],
  :password => ENV['SENDGRID_PASSWORD'],
  :domain => 'yourdomain.com',
  :address => 'smtp.sendgrid.net',
  :port => 587,
  :authentication => :plain,
  :enable_starttls_auto => true
}
```

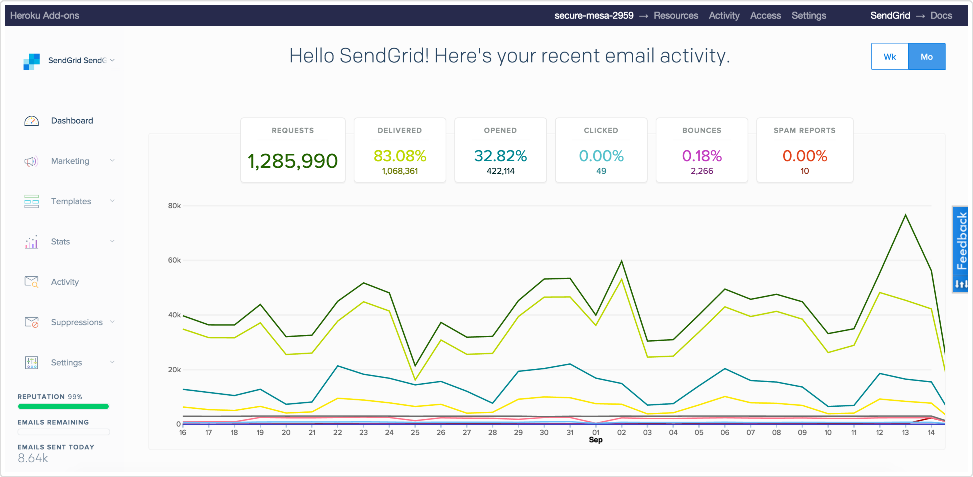
If you want different settings for ActionMailer depending on the environment, update the `config/environments/{environment}.rb` with:

```
# Setup the mailer config
 config.action_mailer.delivery_method = :smtp
 config.action_mailer.perform_deliveries = true
 config.action_mailer.smtp_settings = {
   :user_name => ENV['SENDGRID_USERNAME'],
   :password => ENV['SENDGRID_PASSWORD'],
   :domain => 'yourdomain.com',
   :address => 'smtp.sendgrid.net',
   :port => 587,
   :authentication => :plain,
   :enable_starttls_auto => true
 }
```

Dashboard

For more information on the features available within the SendGrid dashboard please see the docs at **sendgrid.com/docs (http://sendgrid.com/docs/Delivery_Metrics/index.html)**.

SendGrid offers statistics a number of different metrics to report on what is happening with your messages.



The dashboard can be accessed via the CLI:

```
$ heroku addons:open sendgrid
Opening sendgrid for sharp-mountain-4005â€¦
```

or by visiting the Heroku Dashboard (https://dashboard.heroku.com/) and selecting the application in question. Select SendGrid from the Add-ons menu.

Migrating between plans

Plan migrations are easy and instant. Use the `heroku addons:upgrade` command to migrate to a new plan.

```
$ heroku addons:upgrade sendgrid:platinum
-----> Upgrading sendgrid:platinum to sharp-mountain-4005... done, v18 ($399.95/mo)
       Your plan has been updated to: sendgrid:platinum
```

Removing the add-on

SendGrid can be removed via the CLI. If you accidentally remove the SendGrid add-on, re-adding SendGrid will resume your access to the account

```
$ heroku addons:destroy sendgrid
-----> Removing sendgrid from sharp-mountain-4005... done, v20 (free)
```

Support

All SendGrid support and runtime issues should be submitted via on of the Heroku Support channels (https://devcenter.heroku.com/articles/support-channels). Any non-support related issues or product feedback is welcome at http://support.sendgrid.com/home (http://support.sendgrid.com/home).

Additional resources

Additional resources are available at:

- Integrate With SendGrid (http://sendgrid.com/docs/Integrate/index.html)
- Code Examples (http://sendgrid.com/docs/Code_Examples/index.html)