**AWS LAMBDA MAIL**

Debemos crear un Lambda que pueda enviar un Mail con nuestra App Mvc

Abrimos Visual Studio y creamos una nueva aplicación AWS Lambda llamada **AWSLambdaEmailLabs**

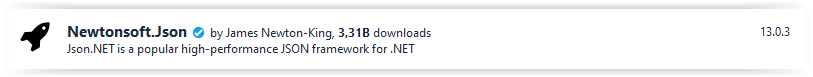
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Aplicación

Descripción generada automáticamente

Instalamos el siguiente Nuget



Creamos una carpeta llamada **Models** y una clase llamada **ModelEmail**

**MODELEMAIL**

**public** **class** ModelEmail

{

**public** **string** Email { **get**; **set**; }

**public** **string** Asunto { **get**; **set**; }

**public** **string** Cuerpo { **get**; **set**; }

}

Creamos una carpeta llamada **Helpers** y una clase llamada **HelperMail**

**HELPERMAIL**

**public** **class** HelperMail

{

**public** **async** Task **SendMailAsync**

        (**string** para, **string** asunto, **string** mensaje)

    {

        MailMessage mail **=** **this.**ConfigureMailMessage(para, asunto, mensaje);

        SmtpClient client **=** **this.**ConfigureSmtpClient();

**await** client**.**SendMailAsync(mail);

    }

**private** MailMessage **ConfigureMailMessage**

        (**string** para, **string** asunto, **string** mensaje)

    {

        MailMessage mailMessage **=** **new** MailMessage();

**string** email **=** "micorreotestingcloudpgs@hotmail.com";

        mailMessage**.**From **=** **new** MailAddress(email);

        mailMessage**.**To**.**Add(**new** MailAddress(para));

        mailMessage**.**Subject **=** asunto;

        mailMessage**.**Body **=** mensaje;

        mailMessage**.**IsBodyHtml **=** **true**;

**return** mailMessage;

    }

**private** SmtpClient **ConfigureSmtpClient**()

    {

**string** user **=** "micorreotestingcloudpgs@hotmail.com";

**string** password **=** "Hotmail12345";

**string** host **=** "smtp.office365.com";

**int** port **=** 587;

**bool** enableSSL **=** **true**;

**bool** defaultCredentials **=** **false**;

        SmtpClient client **=** **new** SmtpClient();

        client**.**Host **=** host;

        client**.**Port **=** port;

        client**.**EnableSsl **=** enableSSL;

        client**.**UseDefaultCredentials **=** defaultCredentials;

        NetworkCredential credentials **=**

**new** NetworkCredential(user, password);

        client**.**Credentials **=** credentials;

**return** client;

    }

}

Escribimos nuestro código en **Function** recibiendo el **ModelEmail** como parámetro

**FUNCTION**

**public** **async** Task<**string**>

**FunctionHandler**(ModelEmail model, ILambdaContext context)

{

    HelperMail helper **=** **new** HelperMail();

**await** helper**.**SendMailAsync(model**.**Email, model**.**Asunto

        , model**.**Cuerpo);

**var** response **=** **new**

    {

        Email **=** model**.**Email,

        Asunto **=** model**.**Asunto,

        Mensaje **=** "Email enviado correctamente"

    };

**string** dataJson **=** JsonConvert**.**SerializeObject(response);

**return** dataJson;

}

El siguiente paso es incluir, dentro de **aws-lambda-tools-defaults.json** el Role para poder subir nuestro Lambda

Texto

Descripción generada automáticamente

Y realizamos un **deploy** de nuestra función Lambda para publicarla.

**dotnet lambda deploy-function lambda-email**

Una vez publicada nuestra función Lambda, nos creamos un Api Gateway para consumirla como Post y comprobamos su funcionalidad.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Implementamos nuestra Api y lo probamos

Texto

Descripción generada automáticamente

Y comprobamos que funciona correctamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente