

Reto BM008 – Insights for Weather

Noviembre 2015



El servicio Insights for Weather en Bluemix permite consultar la información producida por los cientos de miles de sensores desplegados alrededor del mundo para conocer las condiciones climatológicas actuales y predecir sus cambios en el futuro inmediato.

En este reto utilizarás este servicio para crear una aplicación que despliegue las condiciones actuales del sitio donde te encuentras. Este documento te dará algunas pistas para ayudarte a construir la aplicación.

Reto

Construye una aplicación que presente las condiciones climáticas del sitio donde te encuentras usando el servicio Insights for Weather de Bluemix. En la siguiente URL puedes ver un ejemplo:

<http://clima.mybluemix.net>

Puedes construir la aplicación en cualquier lenguaje, aunque en este documento hay tips para construirla más rápidamente usando Node.js.

Prerrequisitos

1. Cuenta de Bluemix.
2. Conocimientos básicos de Node.js

Creando la aplicación



1. En el catálogo de servicios, selecciona el runtime SDK for Node.js para crear una nueva aplicación y **asígnale un nombre único** en el dominio mybluemix.net.
2. Una vez aprovisionada podrás acceder tu nueva aplicación en la URL generada: <http://<myapp>.mybluemix.net>

Por ahora sólo verás la página por defecto, debes modificar el código para que haga lo que tú quieres.

Ve a la sección **visión general** de tu aplicación.



ADD GIT

3. Tienes diferentes opciones para editar el código. Para que puedas avanzar más rápido te recomendamos usar el editor en el browser. Para esto necesitas habilitar un repositorio dando click en el botón **ADD GIT** de la esquina superior derecha.



4. Una vez creado el repositorio se habilitará el botón **EDIT CODE** por medio del cual podrás acceder al editor en línea de IBM Bluemix DevOps Services. Al dar click el editor se mostrará en una nueva pestaña del navegador.



5. En la nueva pestaña habilita la opción **Live Edit** que encontrarás en la parte superior, de esta manera los cambios que realices en el código se reflejen en la aplicación sin necesidad de tener que desplegarla nuevamente.



6. Selecciona el archivo que quieras modificar (por ejemplo app.js) y realiza los cambios que necesites. Para que tus cambios se reflejen debes reiniciar la aplicación.

Estás listo para construir tú aplicación.



Agregando servicios



1. En la sección **visión general** de la aplicación, dentro del portal de Bluemix, puedes asociar servicios con tu aplicación haciendo click en ADD SERVICE OR API

**ADD A SERVICE OR API**

2. Selecciona el servicio **Insights for Weather**. Cuando se crea el servicio Bluemix te solicitará reiniciar la aplicación para actualizar las variables de ambiente.
3. La instancia del servicio que ha sido creada incluye las credenciales requeridas para accederlo desde la aplicación. Vas a necesitar estos datos para desarrollar las llamadas.

Estos datos están disponibles como variables de ambiente en el contexto de ejecución de tu aplicación.

```
{
  "weatherinsights": [
    {
      "name": "Insights for Weather-mt",
      "label": "weatherinsights",
      "plan": "Free",
      "credentials": {
        "username": "6854d1e9-2cb3-4122...",
        "password": "1L4NHNH...",
        "host": "twcservice.mybluemix.net",
        "port": 443,
        "url": "https://6854d1e9..."
      }
    }
  ]
}
```

4. Puedes consultar la documentación del API de este servicio en la siguiente dirección:

<https://www.ng.bluemix.net/docs/services/Weather/index.html>

Tip1: Cómo crear rutas en Node.js y express

```
var express = require('express');
var app = express();

app.get('/j',function(req,res) {
  res.json(JSON.parse('{"msg":"la la la"}'));
});

app.get('/h',function(req,res) {
  res.send("<html>la la la</html>");
});
```

Tip2: Cómo obtener las variables de ambiente de Bluemix desde Node.js

```
var vcap_services = JSON.parse(process.env.VCAP_SERVICES);
```



Tip3: Cómo hacer llamadas REST en Node.js

```
var https = require('request');

var options = {
  uri : 'http://una.api.com/v2/jeje',
  method : 'GET',
  json : true
};

request(options, function(err, res, body) {
  // tu código aquí, body es el json de la respuesta
});
```

Tip4: Cómo obtener tu localización desde el browser usando JavaScript

```
<script src="http://maps.google.com/maps/api/js?sensor=false"></script>
<script>
function doSomething(position) {
  // position.coords.latitude
  // position.coords.longitude
}
navigator.geolocation.getCurrentPosition(doSomething);
</script>
```

Tip5: Cómo invocar un servicio REST desde una página HTML usando JavaScript

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
<script>
jQuery.ajax({
  type: "GET",
  url: "myservice?param1=5",
  contentType: "application/json; charset=utf-8",
  dataType: "json",
  success: function (data) {
    // manejo de datos
  },
});
</script>
```