

## Documentación Backend

### EndPoints:

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/auth/signIn>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/auth/signInAdmin>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/auth/signUp>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/auth/sendCode>

GET - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/auth/userList>

GET - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/auth/getSession>

GET - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/auth/signOut>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/rest/addRest>

GET - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/rest/restList>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/rest/delRest>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/rest/addImg>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/auth/changePass>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/rest/addComm>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/rest/addScore>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/rest/addImgrest>

POST - <https://rt6sa58oih.execute-api.us-east-1.amazonaws.com/dev/rest/updateRest>

Los web services utilizan el Body para los datos de parámetros, en este caso tienen que ir en formato JSON, por ejemplo:

```
{
  "id":"89e13a30-a561-4a95-9082-1930c16807ab"
}
```

## SignIn

**Método para iniciar sesión, los parámetros de entrada son email, password.**

### **SignInAdmin**

**Método para iniciar sesión en el caso de la página web, los parámetros de entrada son email, password.**

### **SignUp**

**Método para registrar un usuario, los parámetros de entrada son email, password.**

### **SendCode**

**Método para mandar el código de verificación para el forgot password, los parámetros de entrada son email.**

### **Userlist**

**Método de obtener la lista de usuarios para ser mostrada en el web app.**

### **GetSession**

**Recupera la sesión por medio de un jwt que está guardado por medio de una cookie.**

### **SignOut**

**Invalida el cookie que está en el header de la solicitud.**

### **AddRest**

**Agrega un restaurante, con los siguientes parámetros: name, x, y, food, contactInfo, schedule, images, cost**

### **RestList**

**Método para obtener el listado de restaurantes guardados en la base de datos.**

### **DelRest**

**Método para eliminar un restaurante determinado por medio de un id, cada restaurante posee un id único.**

### **AddImg**

**Agrega una imagen, en este caso b64, a nuestro bucket de datos, donde devuelve un url público para ser utilizado. Parametros: image.**

### **ChangePass**

**Cambiar la contraseña de un usuario que solicitó forgot password, los parámetros son: code, password, email**

### **AddComm**

**Agrega un comentario a determinado restaurante, los parámetros son: id, comment.**

### **AddScore**

**Agrega un comentario a determinado restaurante, los parámetros son: id, score.**

### **AddImgrest**

**Agrega una imagen a determinado restaurante, parámetros serían la image y el id del restaurante.**

### **UpdateRest**

**Actualiza los datos principales de un restaurante, por medio del id y sus distintos parámetros, para este caso serían: id, name, x, y, food, contactInfo, schedule, cost.**

La tabla de la base de datos de restaurantes cuenta con los siguientes campos:

id	autogenerado (uuid)
name	String
score	List of String
images	List of String
comments	List of String
x	String
y	String
schedule	String
cost	Integer
food	String
contactInfo	String



