

CONFIGURACIÓN DE NUEVOS SERVICIOS

Características generales

Para registrar un nuevo servicio en el sistema, este debe cumplir una serie de especificaciones técnicas para que el bróker pueda consumir correctamente los servicios, y más tarde redirigir la respuesta hacia el cliente.

Arquitectura del servicio

La arquitectura donde estará alojado el servicio debe ser del tipo API REST. Las implicaciones de esto se expondrán a continuación.

Es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP.

REST se compone de una lista de reglas que se deben cumplir en el diseño de la arquitectura de una API.

Llamadas a la API

Las llamadas al API se implementan como peticiones HTTP, en las que:

La URL representa el **recurso**:

- Ej. `http://www.webdeejemplo.com/api/recurso/1`

El método (HTTP Verbs) representa la **operación**:

- Ej. `GET http://www.webdeejemplo.com/api/recurso`

El código de estado HTTP representa el **resultado**:

- Ej. `200 OK HTTP/1.1`
- Ej. `404 NOT FOUND HTTP/1.1`

HTTP VERBS

En este caso únicamente utilizaremos unos de los verbos de CRUD (Crear, Leer, Actualizar y Borrar), ya que el bróker no se encarga de administrar las API's, únicamente solicita un servicio en particular. Por este motivo se utilizará:

- **GET:** Para consultar y leer recursos.

Por lo tanto, lo único que debe tener el recurso para poder ser consumido por el sistema es tener implementado el método GET.

Encabezados HTTP

Dentro de los encabezados HTTP se deberá incluir el encabezado:

- Content-type: application/json

Tipos de datos

El formato a usar será JSON.

JSON admite distintos tipos de datos estándar. De todos los formatos disponibles únicamente utilizaremos el formato String.

- Number: Un número que puede o no contener decimales.
 - Ej: 54, 55.3, 103.5
- String: Una cadena de caracteres.
 - Ej: "buenos dias", "bus", "silla"
- Boolean: Indica verdadero o falso
 - Ej: true, false
- Array: Un conjunto de elementos.
 - Ej: ["buenos dias", "bus", "silla"]
- Object: Se puede definir como un mapa de elementos.
 - Ej: {"id": 2232, "nombre": "Juan", "apellido": "Domínguez"}
- Null: Valor nulo
 - Ej: null

Respuestas

Las respuestas deberán ir en formato JSON. Las respuestas del servicio solicitado deberán venir en formato String no pudiendo rebasar los 200 caracteres (Twitter permite hasta 280 caracteres, pero 80 deben ser reservados para la respuesta estándar del bot). Si el servicio incluye una imagen, deberá venir codificada en base64 en formato String.

Ejemplo:

```
{
  "servicio": [
    {
      "resultado": "resultado del servicio solicitado en texto",
      "imagen": "\/9j\/4AAQSkZJRgABAQAAQABAAD\..\"",
    }
  ]
}
```

Nombres de los recursos

Para que el sistema sea lo más uniforme posible, a la hora de añadir un nuevo servicio se deben seguir un seguido de buenas prácticas para asignarle los nombres a los recursos.

- Plural mejor que singular, para lograr uniformidad:
 - Obtenemos un listado de imagenes: GET /v1/imagenes
 - Obtenemos una imagen en particular: GET /v1/imagen/1
- Url's lo más cortas posibles.
- Una URL debe identificar un recurso o una colección de recursos.
- Evitar guiones y guiones bajos. No utilizar acentos ni caracteres especiales en la URL.
- Deben ser semánticas para el cliente.
- Utiliza nombres y no verbos.
- Estructura jerárquica para indicar la estructura: /v1/imagen/1234/descripción
- La profundidad de una URL que describe un recurso no deberá ser más profunda que recurso/identificador/recurso.

Manejo de errores

Los errores deberán ser devueltos utilizando los errores estándares HTTP en el header de respuesta.

Código HTTP	Descripción
200	Ok
400	Parámetros incorrectos en la entrada
401	Token de acceso expirado o invalido
403	Autenticación OAuth incorrecta
404	Recurso no encontrado
405	Método HTTP no esperado. Por ejemplo: se esperaba un HTTP GET y se recibió un HTTP POST
429	Se están recibiendo muchos requests de parte de tu aplicación. Se está limitando el acceso
500	Error interno del servidor.

Además, en el contenido de la respuesta se deberá incluir un objeto JSON con los siguientes campos:

Llave	Valor
código	Código interno de error
mensaje	Descripción del mensaje de error

Registrar Servicio

Para registrar un nuevo servicio basta con utilizar la aplicación Java ***ConfiguradorDeServicios***. Esta aplicación con interfaz gráfica permite registrar un servicio en el sistema mediante un formulario.

Para agregar un nuevo servicio todos los campos son requeridos.

Nombre del servicio: Especifica el nombre del servicio

URI: Cadena de caracteres que identifica los recursos web de manera inequívoca. A diferencia de las URL's estas no cambian en el tiempo.

Estado: Activo o Inactivo

Descripción: Breve explicación del servicio

Parámetros de entrada: Formato de los parámetros de entrada.

Formato Respuesta: Texto o Texto+Imagen