Diseño API REST: RAML

Sergio Julio Garcia Urdiales
05/03/2018

DESARROLLO DE LA PRÁCTICA

Los pasos para el desarrollo de la práctica han sido:

1. Desarrollo de los documentos RAML.
2. Creación de la base de datos
3. Desarrollo de la api con NODE JS
4. Desarrollo del cliente con HTML + JS
5. Generación de HTML a partir de RAML.

Siguiendo las especificación, los servicios implementados aparecen documentados en el contrato, siendo este el que indique los parámetros a recibir y los parámetros esperados al ejecutar las consultas. En la entrega se incluyen los 3 documentos html generados para consultar los endpoints de la API.

Cada uno dispone de los siguientes parámetros de entrada y salida.

	Entrada	**Salida**
validarNIF	nif : String SoapKey:String	out:Boolean
validarIBAN	iban : String SoapKey:String	existe: Boolean error: String
consultaCodigoPostal	cp : String SoapKey:String	codigoPostal: String poblacion: String provincia: String
generarPresupuesto	fechaPresupuesto : Date SoapKey:String referenciaProducto:String cantidadProducto: Int idCliente: Int	idPresupuesto: int presupuestoGeneradoCorrectamente: Boolean

Para ejecutar los servicios se consulta la RestKey en la base de datos, permitiendo ejecutar el servicio o devolviendo mensajes de error con el código de de estado HTTP correspondiente indicando que no se tienen permisos si esta clave es incorrecta. Para poder validar esta clave, la api requiere que sea enviada a través de la cabecera "*RestKey*".

Todos los servicios devuelven en el cuerpo de la respuesta 3 datos más el solicitado. Estos 3 datos aportan información de cómo ha ido la consulta.

- Correcto: Del tipo boolean, que indica si la consulta ha sido satisfactoria o no.
- Status: Código http con el estado, para facilitar el acceso a esta información.
- Message: Con información de la consulta, tanto si ha ido bien como si ha ocurrido algún problema.

Además de esta información, se devuelven los datos consultados en la estructura que se solicita.

Validar NIF: Este servicio devolverá un verdadero o falso en función del resultado de la comprobación.

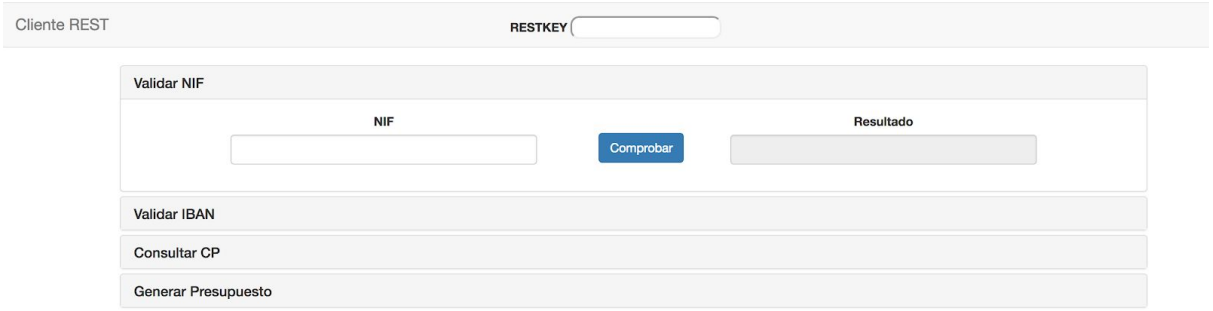
Validar IBAN: Para validar el IBAN se ha utilizado una (iban.js) que permite comprobar su validez.

Consultar CP: La base de datos está poblada con los códigos postales y sus datos correspondientes de todo el territorio. Como el servicio devuelve únicamente una estructura compleja, se devuelve el primer resultado que coincida con el CP introducido. En caso de que no encuentre ningún dato, se devuelve el código de estado 200 que indica que la consulta ha ido bien, pero en este caso se indica que la consulta no ha sido correcta y lleva un mensaje de información acorde.

Generar Presupuesto: Si los datos que se introducen son correctos, guardará una registro de este presupuesto en la base de datos y devolverá la id, en caso de que haya algún problema existen varios códigos de estado que indican que ha ocurrido.

Cliente.

EL cliente ha sido desarrollado en HTML, utilizando javascript, jQuery y AJAX para realizar todas las peticiones a la API.



Documentación

La documentación se ha generado utilizando la herramienta ram2html.

DESPLIEGUE DE LA PRÁCTICA

La aplicación consta de tres partes que necesitan sincronizarse para funcionar:

- API REST
- BBDD
- Cliente

Lanzamiento de la API:

Para lanzar la API hay que situarse con un terminal en el directorio de la api e instalar las dependencias con "*npm install*". Una vez instalado todo basta con lanzar la api con "*nodemon api.js*"

De esta forma la api está corriendo en el puerto 3000 a la que hará peticiones el cliente.

Base de datos.

Se necesita un servicio MYSQL corriendo en el que crear la base de datos que se va a poblar para que los servicios puedan hacer uso de ella.

Para poder hacer uso de sus recursos hay que tener una base de datos llamada "*mtis*", con el usuario "*root*"/"root". Una vez tengamos la base de datos hay que poblar con el archivo sql que se adjunta. Este archivo creará las tablas con los datos correspondientes.

Cliente

Es un archivo html que puede abrirse sin necesidad de un despliegue especial.