Python Programming

Módulo 05

Arquitectura REST

Arquitectura REST

Dijimos que la comunicación entre un servicio web y un consumidor de ese servicio (en nuestro caso, una aplicación de Python) se realiza vía el protocolo HTTP. Pero para que las cosas sean más simples, muchos servicios web tienen una estructura similar llamada *REST.*

REST, *REpresentational State Transfer,* se trata de una arquitectura estándar para comunicaciones web entre sistemas, logrando que se entiendan mucho mejor entre ellos. A los servicios que cumplen con este diseño se les llama RESTful API.

La arquitectura REST se basa en que el cliente envía peticiones para recuperar o modificar recursos, y el servidor responde con el resultado, que puede ser con los datos que hemos pedido o el estado de la petición.

Una petición está formada por:

- Un verbo HTTP que define la operación a realizar.
- Una cabecera o header que incluye información sobre la petición.
- Una ruta o path hacia un recurso.
- El cuerpo del mensaje o body con los datos de la petición.

Veamos un ejemplo. Consideremos un servicio web ficcional a partir del cual se gestionan los alumnos del instituto, cuya dirección es http://api.educacionit.com/.

Utilizando este servicio podemos obtener la lista de alumnos, la información de alguno en particular e incluso agregar nuevos y modificar o eliminar alumnos existentes. Pero también podremos realizar operaciones similares con instructores y personal administrativo del instituto. Así, para poder distinguir qué tipo de información queremos obtener, crear, modificar o eliminar, el servicio proveerá direcciones de URL más específicas, tales como:

- `http://api.educacionit.com/alumnos`
- `http://api.educacionit.com/instructores`
- `http://api.educacionit.com/administrativos`

Cada una de estas “secciones” de un servicio web, en la terminología de la arquitectura REST, se denomina recurso. Así, quitando la primera parte del dominio para simplificar, tenemos tres recursos: `/alumnos`, `/instructores` y `/administrativos`. No obstante, por lo general, los recursos tienen nombre en singular (más adelante se verá más claramente por qué), de modo que serían, más bien, los siguientes:

- `http://api.educacionit.com/alumno`
- `http://api.educacionit.com/instructor`
- `http://api.educacionit.com/administrativo`

Ahora bien, el protocolo HTTP define un conjunto de verbos (con el nombre de *métodos)* para identificar qué tipo de operación se quiere ejecutar sobre un recurso determinado, a saber:

- `GET`, para leer información;
- `POST`, para agregar nueva;
- `PUT`, para modificar información preexistente;
- `DELETE`, para eliminar.

De modo que toda operación que ejecutemos sobre un servicio web estará constituida, por el momento, por un método (`GET`, `POST`, `PUT` o `DELETE`) y la dirección de URL de un recurso. Por ejemplo, si queremos obtener la lista de alumnos del servicio web en cuestión, la operación se enuncia de la siguiente forma:

```
GET /alumno
```

Si queremos agregar un nuevo alumno, de esta otra:

```
POST /alumno
```

Cuando queremos modificar los datos de un alumno o eliminarlo, debemos indicar, además, cuál es ese alumno. Por lo general, en un recurso, cada registro está identificado por un número. Así, si queremos eliminar el alumno cuyo identificador es el número 3, haremos:

```
DELETE /alumno/3
```

Y asimismo si queremos modificarlo:

```
PUT /alumno/3
```

También es posible indicar un identificar usando el método `GET`, en cuyo caso se retorna la información del alumno especificado en lugar de la lista completa.

```
GET /alumno/3
```

Estos son los conceptos básicos de la arquitectura REST. Con lo visto será suficiente para seguir avanzando.

¡Sigamos trabajando!