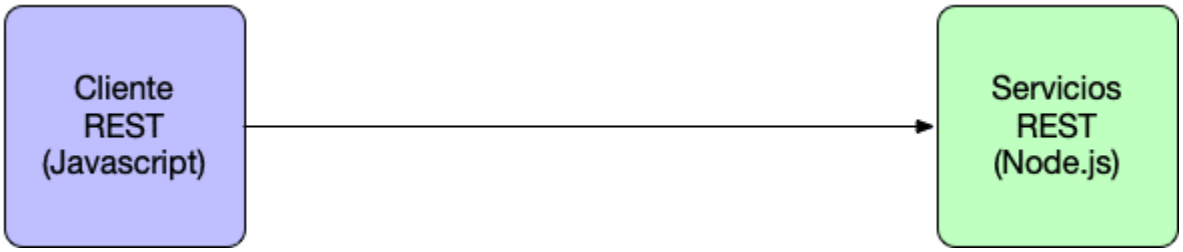
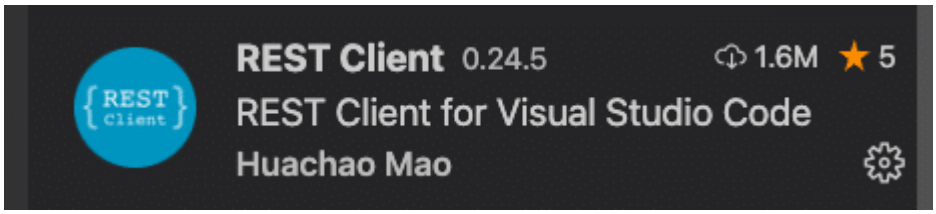
El uso de Visual Studio code REST Client es uno de los más habituales cuando trabajamos con este entorno de desarrollo . En muchas ocasiones usamos visual studio para generar aplicaciones que se conecten a servicios REST y es muy habitual usar clientes adicionales como PostMan para construir estas peticiones



Visual Studio Code REST Client

Este es uno de los plugins más utilizados en Visual Studio para realizar peticiones REST ya que es muy sencillo de configurar . Simplemente le buscamos en la lista de Visual Studio Code y le instalamos

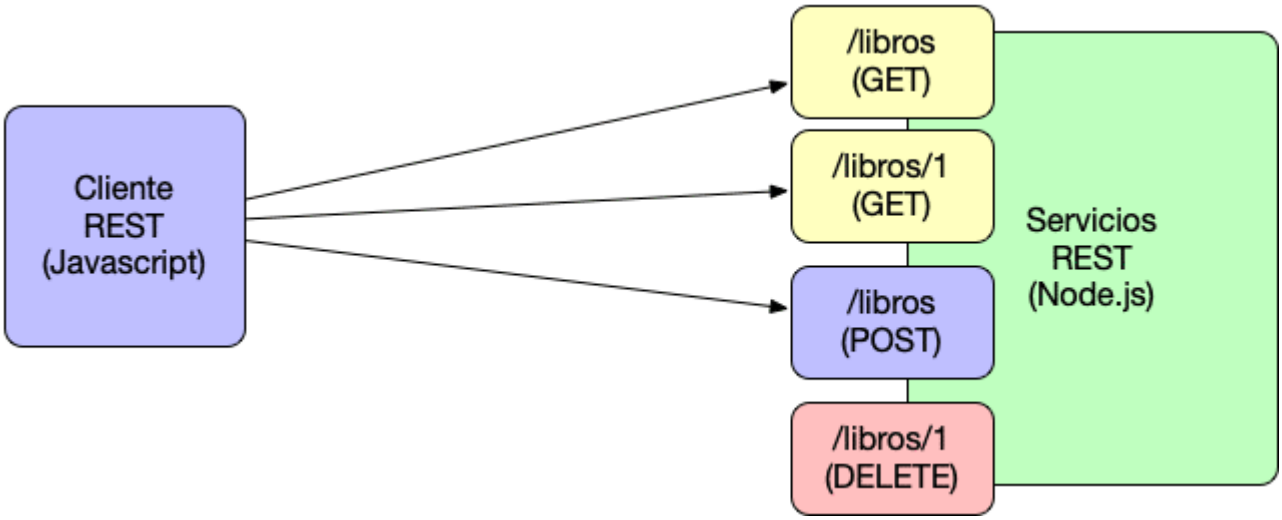


Vamos a construir un ejemplo en el cual usemos Node.js como servidor REST y publiquemos una serie de urls con el recurso de Libros.

```
app.get('/libros', (req, res) => {
  res.send(libros)
})

app.get('/libros/:isbn', (req, res) => {

  let seleccionado=libros.filter(function (elemento) {

      return elemento.isbn==req.params.isbn;
```

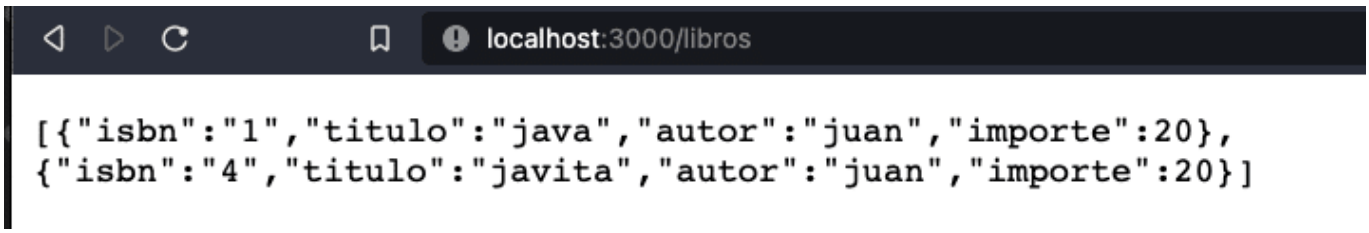
```
  })[0];

  res.send(seleccionado);
})

app.post('/libros',function(req,res) {

    libros.push(req.body);
    res.status(201).send();
})



app.delete('/libros/:isbn', (req, res) => {

  let seleccionado=libros.filter(function (elemento) {
      return elemento.isbn==req.params.isbn;
  })[0];

  let indice= libros.indexOf(seleccionado);
  libros.splice(indice,1);
  res.status(204).send();
})
```

Como podemos observar en este caso hemos publicado las siguientes urls:



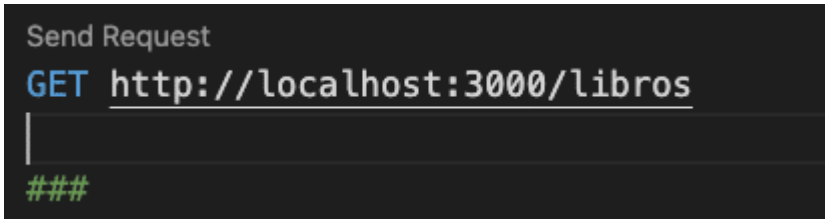
GET y Peticiones REST

Normalmente para realizar peticiones GET no necesitamos mucho más que el propio navegador ya que este es lo que ejecuta como peticiones por defecto.



Veamos lo sencillo que es realizar una petición GET usando Visual Studio Code REST Client en vez de el navegador. Para ello simplemente creamos un fichero con extensión .http .

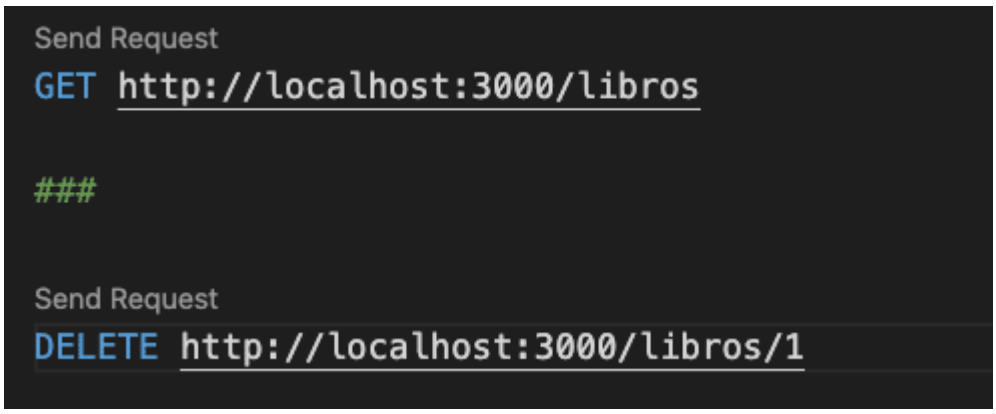
En este fichero lo que tenemos que hacer es simplemente escribir la url a la que deseamos acceder precedida de el verbo HTTP.



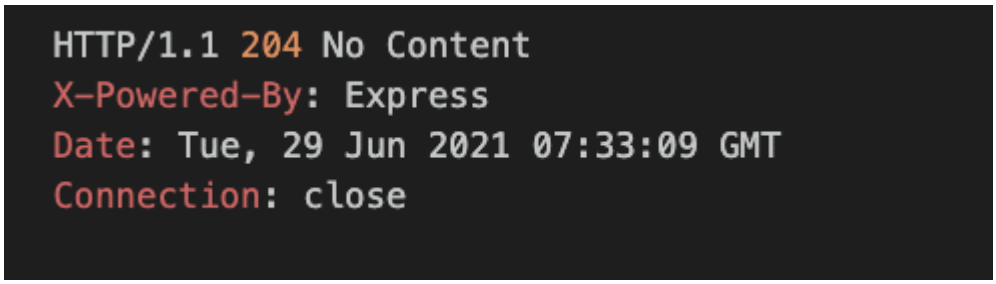
realizada esta operación es suficiente con pulsar en Send Request que es un texto que se activa nada más escribir nuestro código. Realizada esta operación Visual Studio Code realizará la petición al servidor y nos devolverá el resultado

REST y DELETE

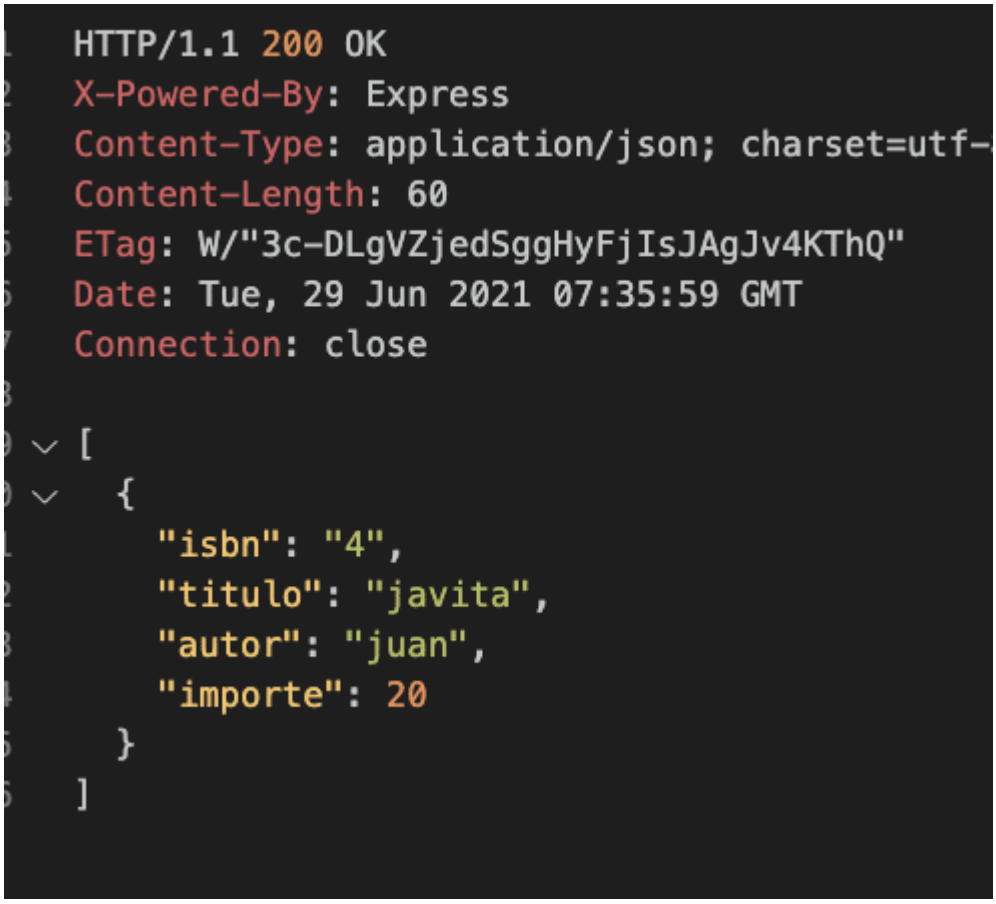
Vamos a lanzar ahora un comando de borrado contra el servicio REST utilizando Visual Studio Code , este comando ya no se puede enviar de forma sencilla desde un navegador:



Como se puede observar simplemente escribimos el verbo y la petición . Separamos una petición de otra con varios caracteres ####. Visual studio lo entenderá sin problemas , una vez pulsado el botón de delete el recurso será borrado del servidor :

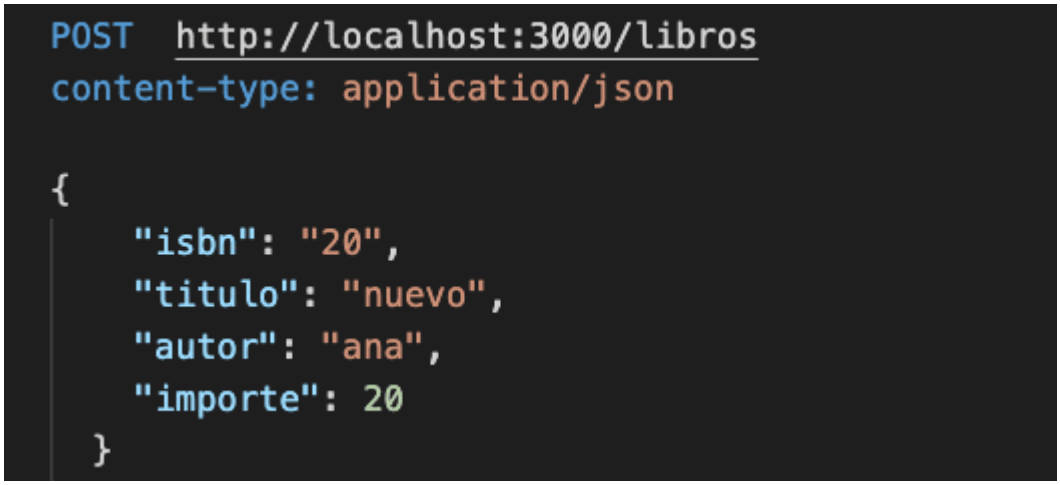


Como podemos ver se trata de algo muy sencillo simplemente se declara la petición de delete y se envía al servidor una vez borrado el registro podemos confirmar el borrado volviendo a solicitar un listado de libros con GET.



Visual Studio Code REST Client JSON y POST

Habitualmente cuando trabajamos con REST es muy común tener que enviar datos en formato JSON al servidor vía POST o PUT . En nuestro caso vamos a ver lo sencillo que se puede trabajar con ello utilizando Visual Studio REST Client. Simplemente definimos la petición de POST y el Objeto JSON que tiene asociado a ella. Enviamos los datos y se insertará un nuevo registro a nivel del servidor



Acabamos de usar Visual Studio Code REST Client para simplificar nuestro trabajo con Arquitecturas REST en el día a día.

Otros artículos relacionados

1. Diseño API REST , rendimiento y OverFetching
2. Spring Boot REST JPA y JSON
3. REST API y su inmutabilidad