

Internet de las Cosas

Máster MUCNAP

Seminario o RESTClient. Instalación y Uso básico

Versión 1.0

Actualización: 16/11/2020

Profesor: Joan Fons i Cors

Control de versiones

Fecha	Autor	Descripción
16/11/2020	Joan Fons	Versión inicial del documento

RESTClient

Objetivos

- Utilizar RESTClient para consultar servicios REST a través del Navegador
- Analizar ejemplos de consulta vía el cliente RESTClient

Tabla de Contenido

OBJETIVOS	2
TABLA DE CONTENIDO	2
INTRODUCCIÓN	3
INSTALACIÓN DEL CLIENTE RESTCLIENT	4
FUNCIONALIDAD DEL CLIENTE	5
EJEMPLOS DE PETICIONES	8
OBTENER INFORMACIÓN DE UN RECURSO	8
EJECUTAR UNA OPERACIÓN SOBRE UN RECURSO	11

Introducción

Este documento contiene información sobre:

- cómo instalar el cliente RESTClient
- que funcionalidad ofrece para realizar invocaciones a una API REST
- ejemplos de uso

Instalación del cliente RESTClient

Para probar servicios REST, se recomienda instalar un cliente REST a un navegador web, y lanzar las invocaciones de prueba desde allí. Una vez se comprueba que funcionan con este cliente, también funcionarán desde el código.

Se recomienda usar el RESTClient (extensión de Firefox) para tal fin.
<https://addons.mozilla.org/ca/firefox/addon/restclient/>

Una vez instalado, tendremos acceso mediante su barra de herramientas

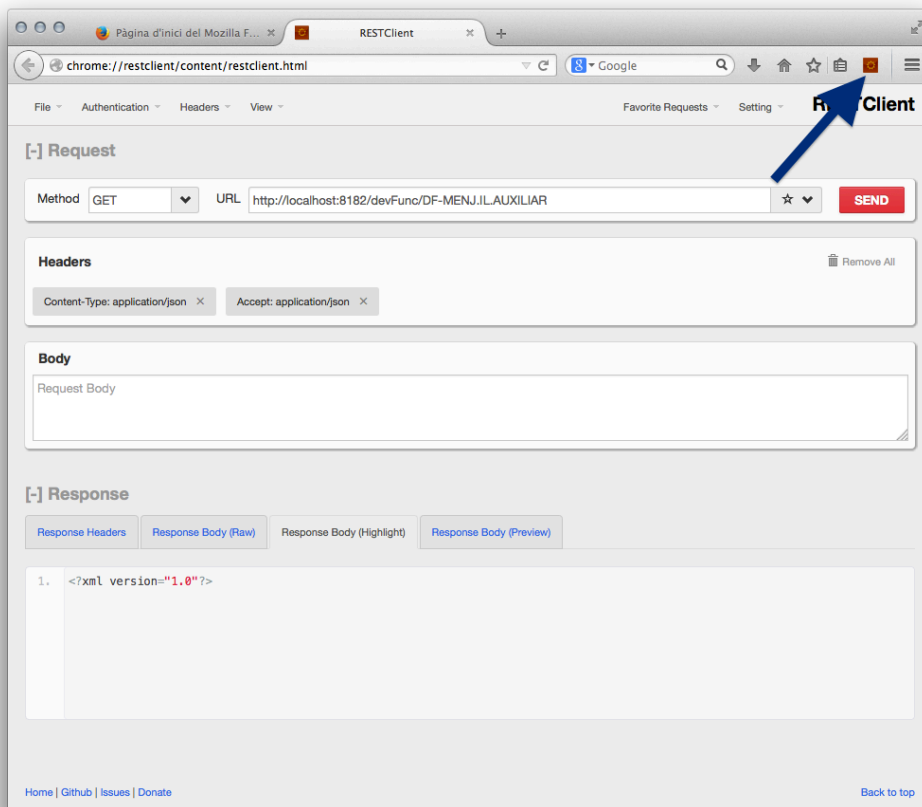


Figura 1. RESTClient instalado en Firefox

Funcionalidad del cliente

Este cliente permite:

- Realizar una petición REST (*Request*)
 - especificar la URL a invocar
 - seleccionar uno de los métodos del API Universal de HTTP (GET, POST, PUT, DELETE, HEAD, OPTIONS, TRACE y CONNECT)
 - configurar las cabeceras (Headers)
 - indicar un Body (Payload) a enviar al servidor
- Analizar la respuesta REST (*Response*):
 - ver las cabeceras
 - ver la respuesta (Body) en 'RAW' (visión 'en bruto')
 - ver la respuesta (Body) en 'Highlight o Preview' (visión 'en bonito')

Para realizar las invocaciones correctamente, será necesario averiguar los formatos en los que hay que enviar el Payload y en qué formato queremos la respuesta, siempre que el servicio la soporte. Para ello, será necesario añadir cabeceras (al Header) para indicar este hecho (Content-Type y Accept, respectivamente).

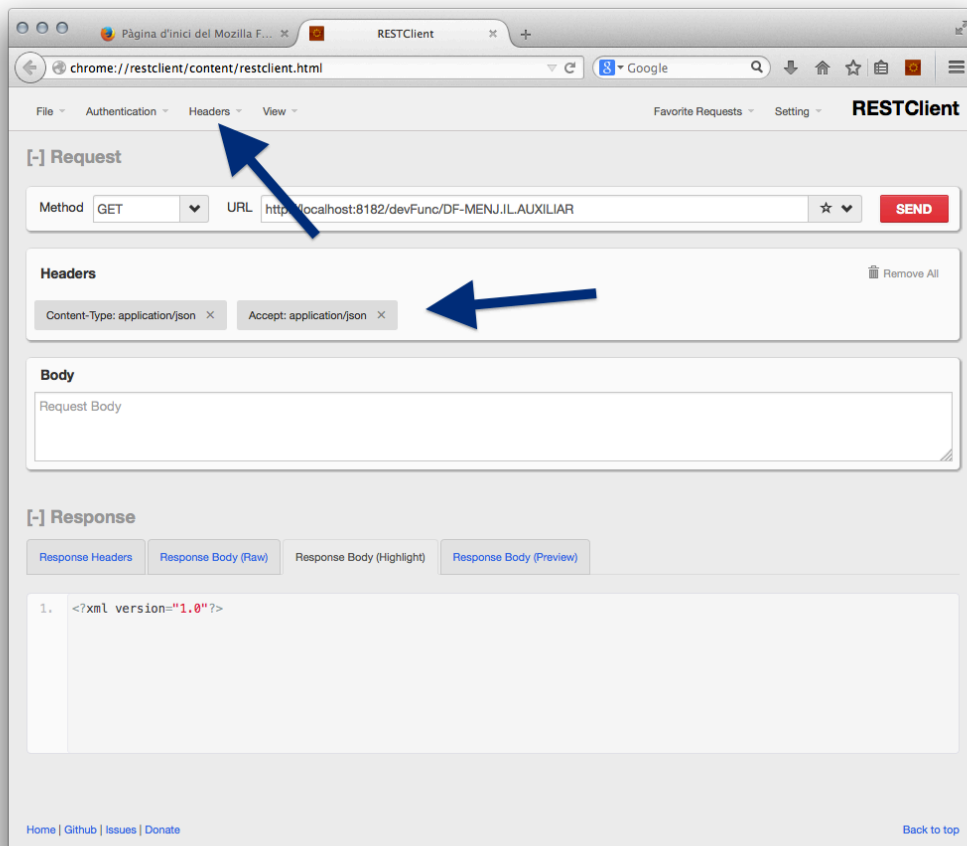


Figura 2. Configurar Cabeceras en RESTClient

Para realizar peticiones GET y PUT, e indicar que se quiere recibir JSON el resultado en formato JSON, se debe configurar la cabecera (Header):

`Accept:application/json`

Para enviar un Payload en JSON (petición PUT), hay que configurar una cabecera:

`Content-Type: application/json`

Para hacer esto en el RESTClient, en el menú superior Headers..., hay que añadir dos Custom Headers:



Request Header

Name
Content-Type

Value
application/json

☒ Save to favorite

Okay Cancel

Figura 3. Cabeceras Personalizadas

Ejemplos de peticiones

A continuación se muestran algunos ejemplos de invocaciones de servicios REST. Se usará como ejemplo el API REST de la ToDoList disponible en el seminario de desarrollo REST (no se requiere conocer el ejemplo a priori).

Obtener información de un Recurso

Para obtener información de un recurso, se va a configurar una llamada mediante la operación GET a una URL en concreto.

En el ejemplo de una ToDoList, podríamos tener un recurso 'listas' que nos ofrezca información sobre las listas disponibles en la ToDoList.

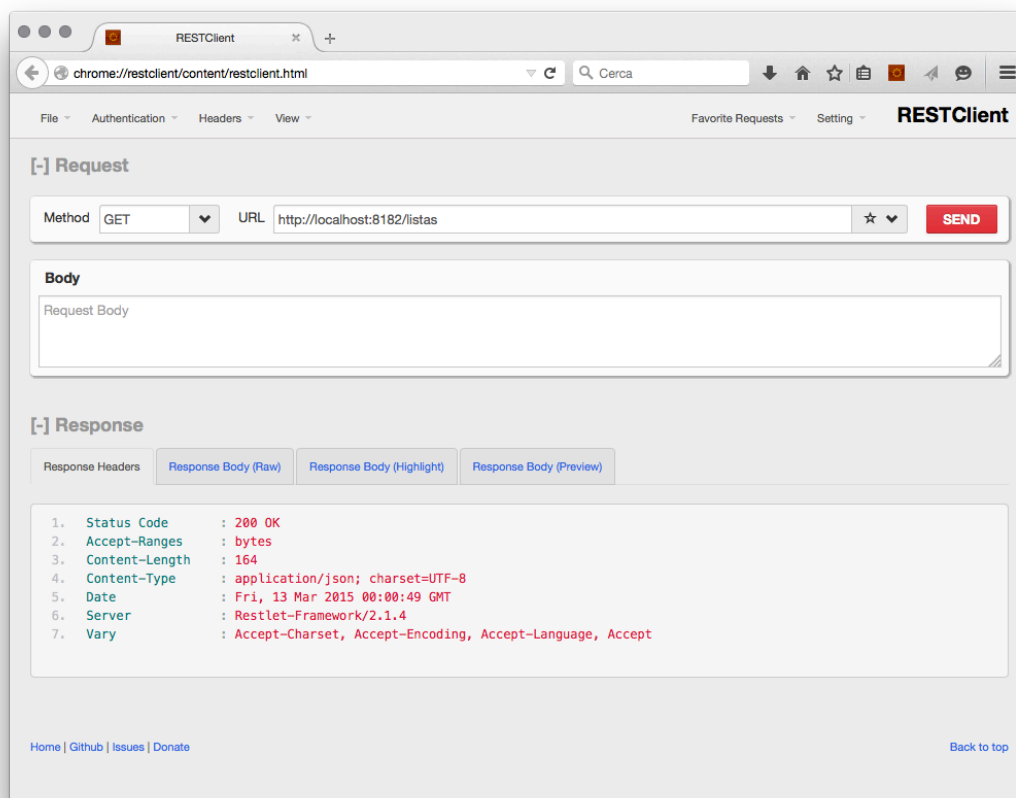


Figura 3. Llamada a GET funcionando correctamente

La Fig. 3 muestra la configuración de la llamada (sección Request) mediante una operación GET a la URL:

`http://localhost:8182/listas`

y la respuesta recibida (sección Response), donde aparecen las cabeceras:

Status-Code: 200 (todo ha ido OK)

Content-Type: application/json (nos informa que hemos recibido datos en formato JSON)

Además de estas cabeceras, existen otras, y el servicio REST con el que se interactúa, podría introducir nuevas.

La Fig. 4 muestra el cuerpo de la petición anterior, donde se puede ver el resultado en JSON (tal y como se ha indicado en las Headers), obteniendo como resultado información sobre el recurso solicitado. En este caso, el recurso es una ToDoList, de la que vemos su link, e información resumida sobre las diferentes listas que la componen y su link.

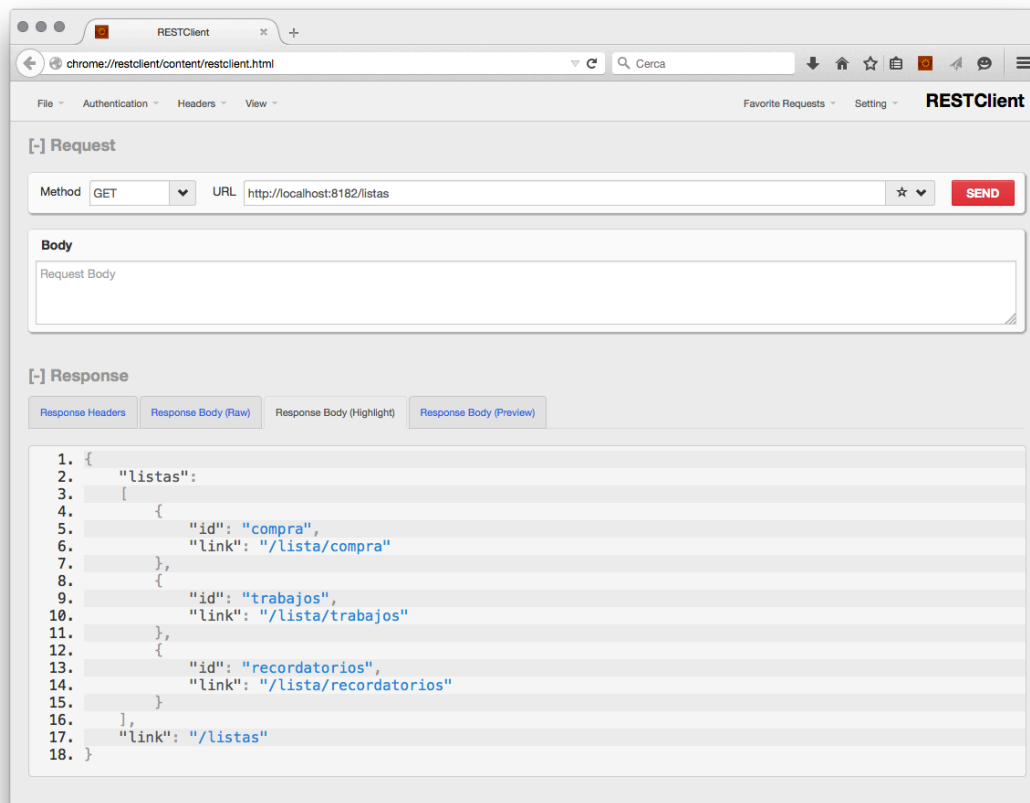


Figura 4. Resultado de la llamada en formato JSON

Ejecutar una operación sobre un recurso

Para ejecutar algún cambio sobre un recurso se suelen usar las operaciones PUT o POST (en función de la situación y de la implementación del servicio REST).

En el caso de la ToDoList, para crear una nueva lista, se debe ejecutar una operación POST pasando como Payload los datos de la nueva lista (en formato JSON).

La Fig. 5 muestra la invocación POST a la URL:

`http://localhost:8182/listas`

, con PAYLOAD

```
{"id": "invitados", "nombre": "Invitados a la fiesta"}
```

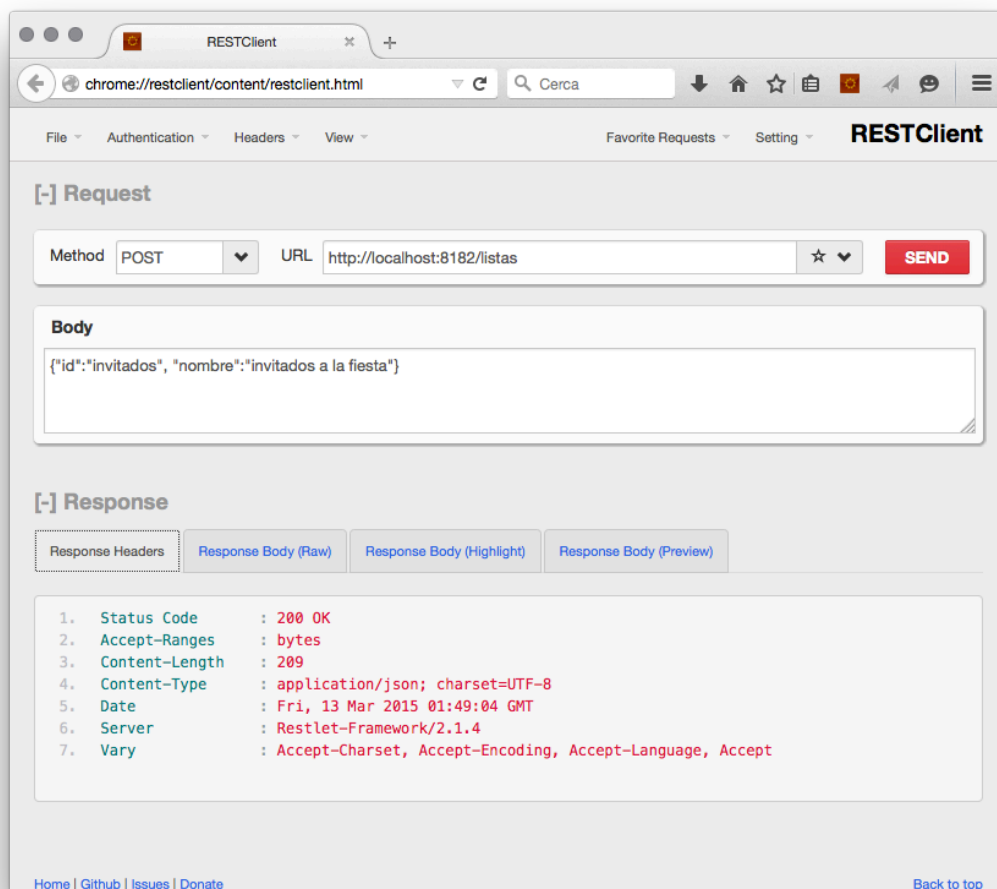


Figura 5. Llamada a POST

Como se puede observar (sección Response) la respuesta recibida indica que:

Status-Code: 200 (todo ha ido OK)

Content-Type: application/json (nos informa que hemos recibido datos en formato JSON)

Si seleccionamos alguna pestaña *Response Body*, podremos ver, en el Fig. 6 (en formato JSON, como se ha indicado), las listas actuales, en las que aparece la nueva lista creada.

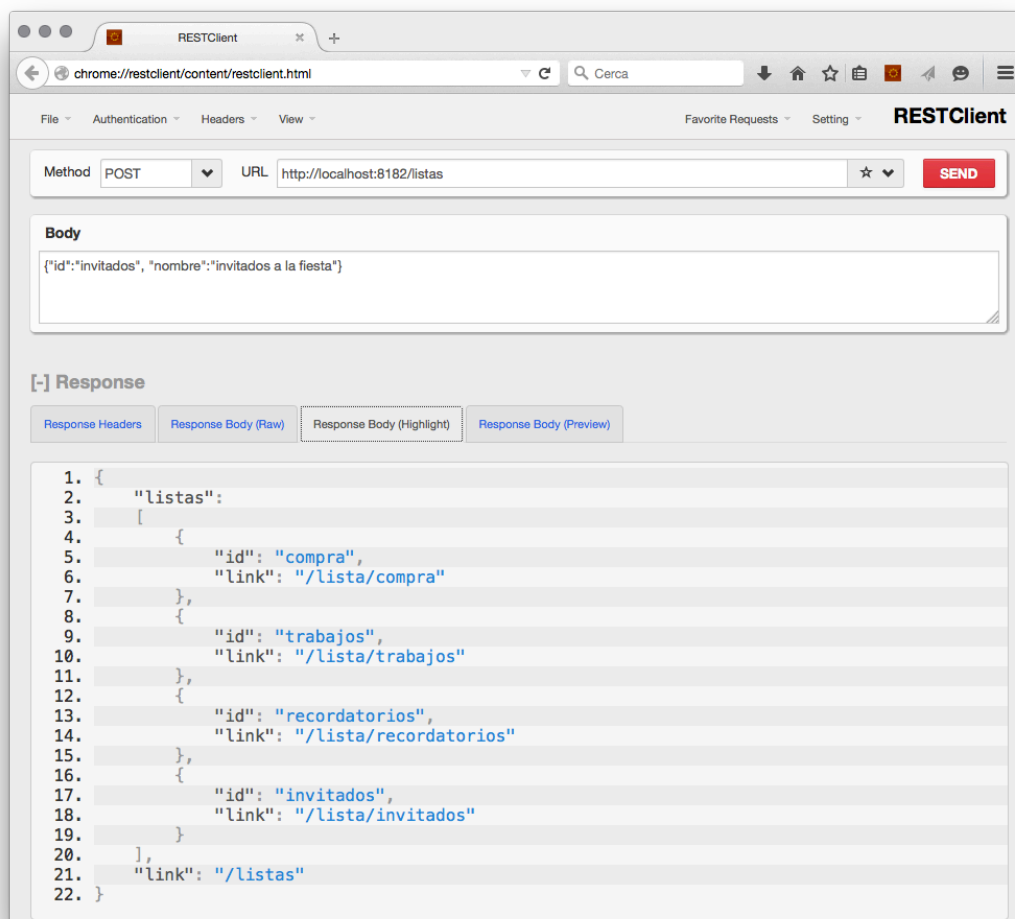


Figura 6. Resultado del POST