

Prácticas con REST

Servicios Telemáticos 2015-2016

Grado en Ingeniería Telemática

Universidad Rey Juan Carlos

<http://gsyc.urjc.es>

Observaciones

- Crea el directorio `~/st/practica05` donde escribirás el resto de ficheros de esta práctica. Recuerda que la recogida de las prácticas se hará automáticamente el día del examen, así que es muy importante que respetes al pie de la letra todos los nombres de ficheros y directorios que indiquen todos los enunciados.

Práctica 5.1 jsonplaceholder

jsonplaceholder¹ es un sitio web para aprender a programar con REST y facilitar la depuración de programas. Ofrece una API para acceder a recursos típicos como usuarios, mensajes de un foro y fotos. Consulta el API descrito en su página principal y:

1. Crea el script `~/st/practica05/foro.py` para consultar este servicio web
2. Si se invoca sin parámetros, mostrará por salida estándar todos los mensajes, con un formato similar a este

```
De: Bret
nesciunt iure omnis dolorem tempora et accusantium
-----
consectetur animi nesciunt iure dolore
enim quia ad
veniam autem ut quam aut nobis
et est aut quod aut provident voluptas autem voluptas
```

```
De: Bret
optio molestias id quia eum
-----
quo et expedita modi cum officia vel magni
doloribus qui repudiandae
vero nisi sit
quos veniam quod sed accusamus veritatis error
```

```
De: Antonette
et ea vero quia laudantium autem
-----
delectus reiciendis molestiae occaecati non minima eveniet qui voluptatibus
accusamus in eum beatae sit
vel qui neque voluptates ut commodi qui incidunt
ut animi commodi
```

Observa que aparece el *username*, el título del mensaje, una línea en texto plano, el cuerpo y una línea en blanco

3. Si se invoca con la opción `-u <NOMBRE_USUARIO> --user <NOMBRE_USUARIOS>`, por ejemplo `-u bret`, mostrará solo los mensajes correspondientes a este *username*

Esta opción será insensible a mayúsculas, esto es, el resultado será el mismo para por ejemplo `bret`, `Bret` o `BrEt`

4. Si el usuario no existe, el script debe mostrar un mensaje de error adecuado por `stderr`

¹<http://jsonplaceholder.typicode.com>

Práctica 5.2 holamundo REST

Prueba un holamundo en flask, como el de las transparencias

Práctica 5.3 servidor REST

Escribe un script `~/st/practica05/server01.py` que al recibir una petición GET al recurso `posts` se comporte igual que `jsonplaceholder`²

Añade también el recurso `/posts/n` donde `n` es el número de mensaje, por ejemplo `/posts/3`

Siempre imitando el comportamiento de `jsonplaceholder`. Los valores a devolver por tu servidor pueden ser cualquiera, con tal de que respeten el formato de `jsonplaceholder`. No hagas que tu servidor consulte a su vez a `jsonplaceholder`, simplemente fija los valores a devolver como una constante en el fuente.

Práctica 5.4 servidor REST

Escribe un script `~/st/practica05/server02.py` incrementando el script anterior con el recurso `/users`

Práctica 5.5 servidor REST

Escribe un script `~/st/practica05/server03.py` incrementando el script anterior con el recurso `/posts?userId=n` por ejemplo `/posts?userId=1`

Comprueba que tu script `~/st/practica05/foro.py` puede consultar tu servidor, con solo cambiar la URI

Práctica 5.6 Cambio de moneda

Escribe un script `~/st/practica05/exchange.py` que devolverá el tipo de cambio del bitcoin para diversas divisas, según la siguiente especificación.

- Cada divisa tiene un código ISO 4217, que es un identificador de 3 letras mayúsculas. Por ejemplo USD (United States Dollar) para el dólar, EUR para el euro, GBP para la libra esterlina, etc. El código ISO 4217 del bitcoin es XBT, aunque es muy habitual usar las siglas BTC.
- El precio de una divisa frente a otra se indica mediante la concatenación de sus dos códigos ISO 4217. La primera se denomina *divisa base* y la segunda, *divisa cotizada*. Por ejemplo EURUSD = 1.13. Aquí la divisa base es el euro, y la cotizada, el dólar. Este valor (1.13) indica cuántas unidades de la divisa cotizada hacen falta para comprar una unidad de la divisa base. Dicho de otro modo, la primera moneda es la que queremos y la segunda, la que tenemos, la que usamos para pagar.
- Nos interesa tomar como referencia el precio del bitcoin en yuanes chinos (CNY) en la casa de cambio `btchina`, por ser una de las de mayor volumen mundial. Este precio podemos obtenerlo mediante una petición a `https://data.btchina.com/data/ticker` pasando como parámetro la *query string* `market=btccny`. La respuesta devuelve varios precios, tomaremos el valor *buy*.
Pero queremos conocer otros *pares*, como XBTEUR, XBTUSD y algunos otros, calculados siempre a partir del precio BTCCNY.
- Queremos saber el precio del bitcoin expresado en: EUR, USD, GPB, CAD, JPY, AUD, CHF y RUB. Para ello, primero averiguamos el precio en yuanes de esta moneda, por ejemplo CNYEUR, y luego calcularemos el precio XBTCNY.
- Para conocer el precio CNYEUR, haremos una petición a `http://api.fixer.io/latest` pasando como argumento `base=CNY`. Usa tu navegador web para ver el formato que devuelve `fixer.io`.

²<http://jsonplaceholder.typicode.com/posts>

- BTC China ofrece algunos pares directamente, pero no nos interesan (la baja liquidez provoca alteraciones), queremos calcularlo siempre a partir del precio en yuanes.
- La URI de tu servidor será
 - /exchange/XBTEUR
 - /exchange/XBTUSD
 - /exchange/XBTGBP
 - etc
- La respuesta será un valor json (una cadena) con el tipo de cambio solicitado. El cliente podrá pasar el parámetro `amount=5.23`. Por ejemplo `/exchange/XBTEUR?amount=5.23`. Esto significará que el cliente está preguntando cuántos euros hacen falta para comprar 5.23 bitcoins.

Práctica 5.7 Cambio de moneda, con caché

Modifica la práctica del apartado anterior para que la respuesta sea un objeto JSON, y el servidor mantenga una caché con las respuestas

- La respuesta será un objeto JSON. Devolverá un par cuya clave serán el cruce de divisas solicitado, y el valor, el precio. Por ejemplo XBTGBP : 306.03
- Si el cliente hizo una petición con el parámetro *amount*, el objeto incluirá un par con la cantidad de bitcoins solicitada y otro par con el la cantidad de divisa cotizada necesaria.
Ejemplo: para la petición XBTGPB?amount=5.23, con un precio del bitcoin en libras de 306.03, la respuesta sería XBTGPB : 306.03, XBT : 5.23, GBP : 1600.5369
- La caché estará en memoria. Cuando el servidor consulte a BTC China o a fixer.io, guardará la respuesta y la fecha y hora de la respuesta.
- Cuando el servidor reciba una petición, si ya tiene una respuesta en caché, y la respuesta no es obsoleta, servirá el valor almacenado en caché.
- Un valor se considerará obsoleto si tiene una antigüedad superior a cierto número de minutos. Esta antigüedad estará indicada mediante constantes definidas en las primeras líneas del código fuente. Para el precio del bitcoin, el valor por omisión será de 5 minutos. Para el precio de las divisas, el valor por omisión será de 1440 minutos (24 horas).

Obviamente, para poder probar que funcione correctamente, deberás usar valores mucho menores. El número de minutos podrá tener decimales. Por ejemplo, 0.5 minutos será un valor admisible.