



---

## Java Code Challenge

Nos gustaría tener un servicio web RESTful que almacene transacciones (en memoria está bien) y devuelva información sobre esas transacciones.

Las transacciones a almacenar tienen un tipo y un monto. El servicio debe admitir la devolución de todas las transacciones para su tipo. Además, las transacciones se pueden vincular entre sí (usando un "parent\_id") y, por otro lado, necesitamos saber el monto total involucrado para todas las transacciones vinculadas a una transacción en particular.

1. Por favor, completar en Spring Boot (Java o Kotlin) y en no mas de 3 días consecutivos.
2. Completar el proyecto en Bitbucket o Github, para que podamos revisar el código.
3. No usar SQL.

Requerido:

- Tests de integración.
- Aplicación dockerizada.
- Java 11 o superior.
- Claridad del código.
- Correctitud en diseño de arquitectura.

Se valorará positivamente:

- Uso de TDD.
- Desarrollo incremental de la solución mediante el uso de commits.
- Aplicación de los principios SOLID.
- Documentacion.

## Especificación del servicio

### PUT /transactions/\$transaction\_id

```
1 Body:
2
3 { "amount":double,"type":string,"parent_id":long }
4
5
6
```

Codigo 1: PUT transaction

En dónde:

- **transaction\_id** Es de tipo 'long' identificador de una nueva transacción.
- **amount** Es de tipo 'double' especificando el monto.
- **type** es un 'string' que identifica el tipo de la transacción.
- **parent\_id** Es de tipo 'long', opcional. El cual especifica el id de la transacción padre.

### GET /transactions/types/\$type

```
1 Returns:
2
3 [ long, long, .... ]
4
5
6
```

Codigo 2: GET by type

Una lista json de todos los ids de las transacciones para el tipo especificado.

### GET /transactions/sum/\$transaction\_id

```
1 Returns:
2
3 { "sum", double }
4
5
6
```

Codigo 3: GET SUM

La suma de todas las transacciones que estan transitivamente conectadas por su **parent\_id** a **\$transaction\_id**.

### Algunos ejemplos simples podrian ser:

```
1  PUT /transactions/10 { "amount": 5000, "type": "cars" } => { "status": "ok" }
2
3
4  PUT /transactions/11 { "amount": 10000, "type": "shopping", "parent_id": 10 }
5  PUT /transactions/12 { "amount": 5000, "type": "shopping", "parent_id": 11 }
6
7  GET /transactions/types/cars => [10]
8  GET /transactions/sum/10 => {"sum":20000}
9  GET /transactions/sum/11 => {"sum":15000}
10
11
```

Codigo 4: Ejemplos