Code Challenge

We would like to have a RESTful web service that stores some transactions
(in memory is fine) and returns information about those transactions.
The transactions to be stored have a type and an amount. The service should
support returning all transactions of a type. Also, transactions can be linked
to each other (using a "parent_id") and we need to know the total amount
involved for all transactions linked to a particular transaction.

In general we are looking for a good implementation, code quality and how the
implementation is tested. Some discussion about asymptotic behaviour would also
be appreciated.

In detail the api spec look like the following:

`PUT /transactionservice/transaction/$transaction_id`

Body:

```
{  "amount": double, "type": string, "parent_id": long  }
```

where:

- `transaction_id` is a long specifying a new transaction
- `amount` is a double specifying the amount
- `type` is a string specifying a type of the transaction.
- `parent_id` is an optional long that may specify the parent transaction of this transaction.

`GET /transactionservice/transaction/$transaction_id`

Returns:

```
{  "amount": double, "type": string, "parent_id": long  }
```

`GET /transactionservice/types/$type`

Returns:

```
[ long, long, .... ]
```

A json list of all transaction ids that share the same type `$type`.

`GET /transactionservice/sum/$transaction_id`

Returns

```
{ "sum", double }
```

A sum of all transactions that are transitively linked by their `parent_id` to `$transaction_id`.

Some simple examples would be:

```
PUT /transactionservice/transaction/10
    { "amount": 5000, "type": "cars" }
=> { "status": "ok" }


PUT /transactionservice/transaction/11
    { "amount": 10000, "type": "shopping", "parent_id": 10 }
=> { "status": "ok" }


GET /transactionservice/types/cars
=>  [ 10 ]


GET /transactionservice/sum/10
=>  { "sum": 15000 }


GET /transactionservice/sum/11
=>  { "sum": 10000 }
```