Coding Exercise

In a language of your choice, please develop a small REST API that yields results of simple calculations and data lookups.

Steps:

- Please submit your code as a link to a github repository.
- The solution should be as self contained as possible with a README.md that explains:
  - How to execute tests
  - Run in local environment
  - Any prerequisites to setup

Solution Outline

The following is provided

Item	Location	Description
Raw Data	`data/sample.txt`	Raw data file dump of 1 company
Mapping file	`data/lookup.json`	Mapping file of data item to id
Custom Calcs	`README.md`	Custom calcs to be implemented
API Spec	`api-spec.yml`	Swagger doc outlining API
Assertions	`assertions/*`	JSON extracts of calc results

Raw data

Please see the file `data/sample.txt` for a single company data dump. Data format is follows:

company_id	company_name	fiscal_year	data_item_id	data_item_value
124423	Acme	1989	1001	767.28
124423	Acme	1989	1002	157.31
124423	Acme	1990	1001	997.81
124423	Acme	1990	1002	98.50

… and so on …

Where `data_item_id` and `data_item_value` correspond to a lookup json file (`data/lookup.json`):

```
[{
    "data_item_id": "1001",
    "data_item_name": "alpha"
}, {
    "data_item_id": "1002",
    "data_item_name": "beta"
}]
```

Pass through data-items

The full REST api specification is provided below. The general idea is if the API is called with this company id and this data-item type then a simple JSON array will return with all the data for that item. e.g.:

```
/v1/data-items/alpha?company_id=124423
```

Would return

```
[{
  "period": 1989,
  "value": 767.28
}, {
  "period": 1990,
  "value": 997.81
}]

... For all years
```

And likewise beta would return:

```
/v1/data-items/beta?company_id=124423
```

```
[{
  "period": 1989,
  "value": 157.31
}, {
  "period": 1990,
  "value": 98.5
}]

... For all years
```

Custom Calculations

In addition, the solution should support the following custom calculations (rounding the 4 digits behind the decimal) and have an easy to follow pattern to extend them further in the code.

```
omega = alpha * beta * 0.5
gamma = omega * alpha / beta
```

Omega Example:

```
/v1/data-items/omega?company_id=124423
```

Results:

```
[{
  "period": 1989,
  "value": 60350.4084
}, {
  "period": 1990,
  "value": 49142.1425
}]

... For all years
```

REST API specifications

Please follow the specifications in the following Swagger document:

```
swagger: '2.0'
info:
  title: Data Item API
  version: "1.0.0"
host: data.calculator.com
schemes:
  - https
basePath: /v1
produces:
  - application/json
paths:
  /data-items/{type}:
    get:
      summary: Get annual results based on data-item type
      parameters:
        - name: type
          in: path
          description: The calculation type (alpha, beta, gamma)
          required: true
          type: string
        - name: company_id
          in: query
          description: Company numeric identifier
          required: true
          type: number
          format: integer
      responses:
        200:
          description: An array of data items
          schema:
            type: array
            items:
              $ref: '#/definitions/DataItem'

definitions:
  DataItem:
    type: object
    properties:
      year:
        type: number
        format: integer
      value:
        type: number
        format: double
```