

Java Coding Exercise

Implement 3 RESTful operations as a webservice in Java that accumulates a total while preventing the total from exceeding a defined quota.

API

1. **POST /consume** - Accumulate the total

Using the input `value`, adjust and return the current total and remaining amount of quota. Error if the quota is exceeded.

```
POST /consume
{
  "value": 123
}
```

Example: `200 - OK`

Returns the total so far and remaining total (ex quota of 1000)

```
{
  "total": 246,
  "remaining": 754
}
```

Example: `503 - Service Unavailable`

If the quota is exceeded Error HTTP 503 with the amount over the quota

```
{
  "excess": 123
}
```

2. `GET /consume` - Return accumulated total and remaining quota

```
GET /consume
```

Returns the total so far and remaining quota

Example `200 - OK`

```
{
  "total": 246,
  "remaining": 754
}
```

3. `POST /reset` Reset the accumulator back to zero with a new quota

```
POST /reset
{
  "quota": 2000
}
```

Example `200 - OK`

The same response format as above

```
{
  "total": 0,
  "remaining": 2000
}
```

Other Requirements

- The default quota is `1000` at startup

Code Delivery

- The final code should be zipped and emailed back to the sender or uploaded to a code repository like GitHub
- Do not bundle dependencies in the emailed/uploaded deliverable: use Maven.
- It should contain a README.md file with instructions for how to build/run the project on our computers.