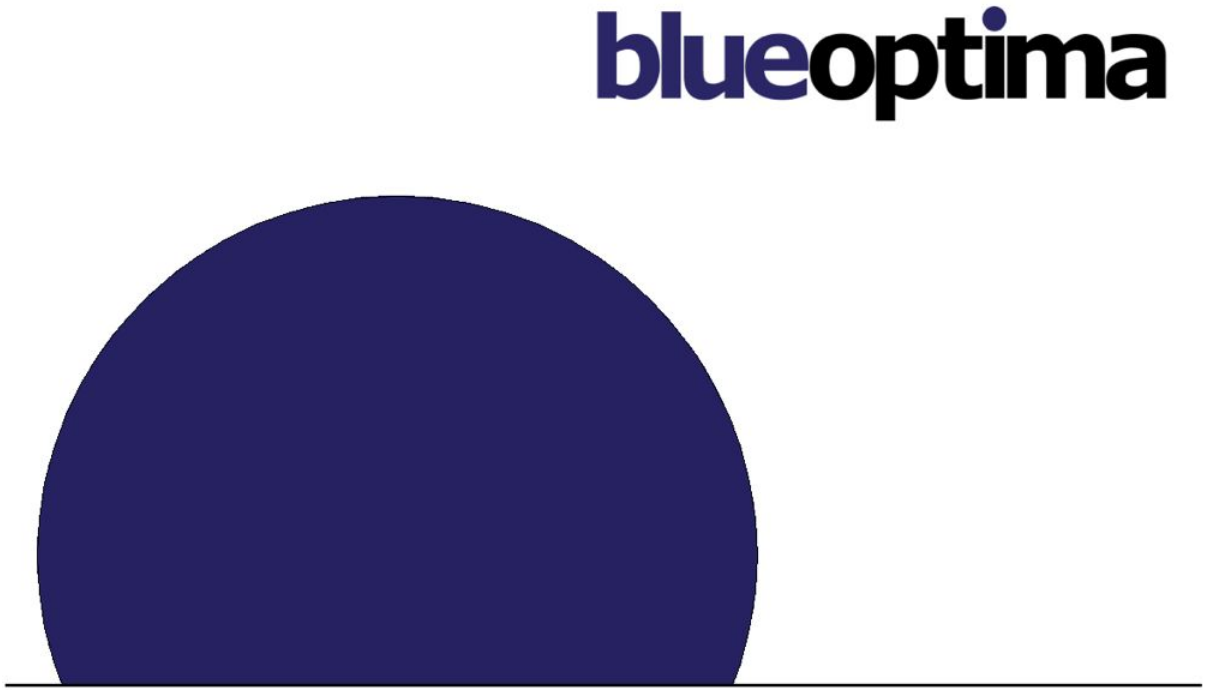

API Rate Limit,

Version: 1.0

Released: Tuesday, Sep 18, 2018

Problem Statement

Deliverables

Scenario

Your application uses RESTful API's which are user by your Clients to build Integration to your application. These integration could be for administration of the application or reporting.

Problem

Off the multiple APIs that are exposed, different APIs would have different resource requirement and while some APIs are light and fast, there are subset that are heavy and increases stress on the Server. Internally it is clear the cost of every API call, however, a client building an integration could be unaware of such cost and could implement an integration which could be taxing on the server and impact the overall performance of the application.

An alternative scenario here is also a programming mistake by a client's developer which might end up calling an API repeatedly in automated way, which could lead to simulate a DOS (Denial of Service) attack and adversely impact the performance of the application.

Work Sample Objective

This work sample requires the team to build a server side rate limiting solution, which tracks the number of requests that are made by a User over a given period of time and block requests when the number of call exceeds the predefined threshold.

The rate limiter solution should cover following cases:

- Different APIs would have different rate limits
- Should be possible to set a default limit. This will be applied when an API specific limit has not be configured
- Solution should consider rate limiting based on the User+API combination. I.e. rate limiting should work based on the combination of User & the API which they are calling. It might happen that to certain users for certain API's we would like to give threshold higher as compared to other users, so system should be able to handle such scenarios. Below is the table which example that with an example.
- Solution should be agnostic to a web application framework and be a plug and play solution.

User	API	Rate Limit
User 1	/api/v1/developers	100
User 2	/api/v1/developers	50
User 1	/api/v1/organizations	250
User 2	/api/v1/organizations	500

Deliverables

1. Source code:
    a. A comilable and working solution with source code
    b. Source code should be zipped and shared either on email or a shareable google drive link. (Do not share the solution or put the code on GitHub or other code sharing system)
    c. Readme file detailing steps to configure an Eclipse or Intellij (or relevant IDE) for the source and steps to compile it
    d. Any dependency on separate application or a server should be listed.
2. **Project Documentation** should include
    a. Explain the approach taken, including (but not limited to) understanding of rate limits, your approach to manage it, information being processed and what it represents.
    b. Data storage format and reasons for the approach. If alternatives were explored, provide them as well.
    c. Any assumptions made.
    d. Suggest any additional parameters that could be additionally taken as input to improve the accuracy of the solution.
    e. State any improvements that you would like to make if more time was available for implementation.

Evaluation

Your submission will be evaluated on completing all the items specified in deliverables list. Once the submission is made, we will arrange a follow up discussion to understand from you the solution itself and ask any question that come out of the submission you make.

Important:

To make evaluation accurate and less time consuming please take note of following.

1. Code formatting: Please indent the code (using tabs or space) and use camel case when defining variables and functions.
2. Documentation: Please document the code where necessary and document it just enough. Excessive documentation is worse than no documentation.
3. Overall summary: Please explain on how to build and execute the code. How and what inputs to be passed and where output can be seen. A general execution flow will also be good for the evaluator to understand the solution that is being provided. As mentioned above, this needs to be just enough to make evaluation effective and avoid excessive documentation.