Airtasker

Recruitment Challenge

Practicalities

Your take-home coding exercise is designed to take close to five hours to complete. You are under no obligation to take any specific amount of time to complete the task. Once you have finished with the challenge, please send us a link to your repo on GitHub.

Please include a README explaining the reasoning behind your implementation and possible limitations. Please help our reviewers by including the commands required to run and test your code. Any relevant documentation for reviewing your challenge should be included in the repo.

The Rules

The code you write should be your own, and should be written without direct assistance. However, feel free to use as many reference resources (Stack Overflow, MSDN, Google, textbooks) as you like.

We are comfortable receiving solutions written in the following languages:

- Ruby / Ruby on Rails
- Java/Kotlin/Scala/Groovy
- Python
- JavaScript
- TypeScript
- GoLang
- C/C++/C#

The Task

The task is to produce a rate-limiting module that stops a particular requestor from making too many http requests within a particular period of time.

The module should expose a method that keeps track of requests and limits it such that a requester can only make 100 requests per hour. After the limit has been reached, return a 429 with the text "Rate limit exceeded. Try again in #{n} seconds".

Although you are only required to implement the strategy described above, it should be easy to extend the rate limiting module to take on different rate-limiting strategies.

How you do this is up to you. Think about how easy your rate limiter will be to maintain and control. Write what you consider to be production-quality code, with comments and tests if and when you consider them necessary.

What we are looking for

There is no one “correct” solution, and there is no trick. The main focus is not a fancy algorithm or a complex codebase. Instead, we are looking for thoughtfully written, high quality software. As you write your solution, consider things such as:

- Readability
- Maintainability
- Efficiency
- Testability
- Extensibility

Be prepared to justify your choices. There are no bonus points for showing off. Good software development skills are far more important to us than extreme language prowess.

Adding to our culture

Our mission is to empower people to realise the full value of their skills.
What is your interpretation of this? Why is this important to you? What do you think this looks like for Airtasker?

These are our values. Please advise on your interpretation of these values. What does this look like for you personally and also as an Engineer?

Stay open	We are open and transparent. We set big goals and then work together to find the solution.
Fit for purpose	We build for purpose, we always ask “why” and don’t embellish.
People matter	We value individuals and believe people are far more than a means to an end.
When it’s on it’s on	We push hard when we need to, play hard as a team and celebrate hard when we win.
Own it	We each own our ideas and our happiness. We always speak up.

Lastly...

- What is important to you in your career?
- What excites you about Airtasker and the opportunity?
- What would you like us to know about you?

Thank you for your efforts and interest in Airtasker!