Welcome to the Blue Co pre interview assignment.

The purpose of this exercise is for us to get a sense of how you would approach designing and implementing a simple service, before we get you in for an interview. We've tried to avoid tricky algorithmic tests in favor of something that shows how you would organise a codebase.

To that end, we'd like you to implement a service with 3 endpoints.

The first endpoint, POST /articles should handle the receipt of some article data in json format, and store it within the service.

The second endpoint GET /articles/{id} should return the json representation of the article

The final endpoint, GET /tags/{tagName}/{date} will return the list of articles that have that tag name on the given date and some summary data about that tag for that day.

An article has the following attributes id, title, date, body, and list of tags. for example

{

"id": "1",

"title": "latest science shows that potato chips are better for you than sugar",

"date" : "2016-09-22",

"body" : "some text, potentially containing simple markup about how potato chips are great",

"tags" : ["health", "fitness", "science"]

}

The GET /tag/{tagName}/{date} endpoint should produce the following json. Note that the actual url would look like /tags/health/20160922

{

"tag" : "health",

"count" : 17,

"articles" :

[

"1",

"7"

],

"related\_tags" :

[

"science",

"fitness"

]

}

the related\_tags field contains a list of tags that are on the articles that the current tag is on for the same day. It should not contain duplicates

the count field shows the number of tags for the tag for that day

the articles field contains a list of ids for the last 10 articles entered for that day.

Deliverables

Please submit the following deliverables

1. Source code for the service described above. Feel free to use any language/toolset you like. The only requirement is that you can describe how to set it up on a mac so we can see it running.
2. Setup/installation instructions
3. A quick (1-2 page) description of your solution, outlining anything of interest about about the code you have produced. This could be anything from why you chose the language and or libraries, why you structured the project the way that you did, why you chose a particular error handling strategy, how you approached testing etc
4. A list of assumptions that you’ve made while putting the system together. We’ve only given you a very loose spec, so you’ll probably need to fill in some blanks while you are working. If you note down the assumptions, for us, then we will be able review the code within the context of those assumptions.
5. We’re always keen to learn and improve so feel free to let us know what you thought of the test and how long it took you to complete.

Bottom of Form