Rental Payments

“Different” offers a web-based app to the tenants to manage their lease. Due to the high demand, the product owner decided to release a new feature that allows tenants to see a list of their past and upcoming rent payments. Imagine you’re part of the engineering team at Different and you’re tasked to implement this feature. Below is the feature that needs to be implemented:

Given a lease id (via the query string):

The UI should be available via a localhost URL. The id of the lease will be passed to the UI as query string field called `leaseId`. For example “http://127.0.0.1:3000/leases.html?leaseId=12” is a valid URL.

- Fetch the lease information via the `https://hiring-task-api.herokuapp.com/v1/leases/:id` URL.
  Note: “`:id`” needs to be replaced with the actual lease id.
  Example: `https://hiring-task-api.herokuapp.com/v1/leases/12`

- Generate a list of past and upcoming rent payments that are based on start date, end date, payment interval, payment weekday and show it on a web page

Alex, as an example, signed the lease “123”. His lease starts on `August 9th, 2018` and ends on `December 28th, 2018`. He pays `$510` a week, on a `fortnightly` basis that happens on `Tuesday` of every second week. Below is the data that the product owner expects to see for the lease that Alex is signed:

From	To	Days	Amount
August, 9th 2018	August, 13th 2018	5	$364.3
August, 14th 2018	August, 27th 2018	14	$1020
August, 28th 2018	September, 10th 2018	14	$1020
September, 11th 2018	September, 24th 2018	14	$1020
September, 25th 2018	October, 8th 2018	14	$1020
October, 9th 2018	October, 22nd 2018	14	$1020
October, 23rd 2018	November, 5th 2018	14	$1020
November, 6th 2018	November, 19th 2018	14	$1020
November, 20th 2018	December 3rd, 2018	14	$1020
December, 4th 2018	December, 17th 2018	14	$1020
December, 18th 2018	December, 28th 2018	11	$801.4

The first payment is on `August, 9th 2018` which is the first day of the lease. Going forward, every second Tuesday, Alex will be charged for 2 weeks till the lease ends on `December, 28th 2018`.

Acceptance criteria for this task:

1. Your solution should be available via a public Github repository (It's free to both create an account and a public repository in Github)
2. There should be instructions about how to run this app
3. The app should be fetching lease data from the mentioned API
4. You can use any of Backbone, Angular, Vue or React frameworks or libraries or not use a framework at all. You will get bonus points by using React though.
5. To implement this task, you should not use any other Javascript or CSS library.

Docs for [GET] https://hiring-task-api.herokuapp.com/v1/leases/:id

As mentioned earlier, you need to fetch the lease data from an online API that is available via the `https://hiring-task-api.herokuapp.com/v1/leases/:id` URL. This URL is available publicly and you don't need credentials to access it. Using this endpoint is completely free and there are no rate limits on this API. The API always responds in JSON format which contains the following data:

Name	Type	Description
id	string	Lease id
start_date	string	Start of the lease
end_date	string	End of the lease
rent	number	Weekly rent amount
frequency	weekly | fortnightly | monthly	Payment frequency
payment_day	monday | tuesday | wednesday | thursday | friday	Payment day in the week

Below is an example output:

```
{
      “id”: 123,
      “start_date”: “2018-07-12”,
            “end_date”: “2018-11-17”,
      “rent”: 545,
      “frequency”: “weekly”,
      “payment_day”: “wednesday”
}
```

Bonus task 1

The https://hiring-task-api.herokuapp.com/v1/leases endpoint returns a list of current tenants and their lease ids. Similar to the previous task, implement a UI that fetches the leases and renders them on a webpage. When a user clicks on one of the rendered leases, previously implemented UI will be used to show a list of past and upcoming rental payments.

Bonus task 2

Implement both APIs that are used within this code challenge in Javascript.

Assessment

When assessing your coding challenge, we include multiple factors. These are standard factors in software. Below are some of them:

1. Code readability
2. Code maintainability
3. Documentation
4. Testability
5. Automated tests
6. Overall architecture
7. Unhappy paths vs happy paths
8. Security
9. Scalability
10. Reusability of the code
11. The simplicity of the solution

Submission

Please send a link to your public repository to mehdi@different.com.au. We expect to receive a solution within 7 days but let us know if you need more time. If you have any questions about this task, send me an e-mail and I'll do my best to help you :)