REPLICON™

Calgary Office | 910 - 7th Ave SW, Suite 800 | Calgary, Alberta, T2P-3N8, Canada
North America: 1-877-662-2519 | International: +800 6622 5192

Technical Assessment – Full Stack Engineer

The Problem

Our customers would like an automated solution to assign employees to shifts. Their current system has a service which provides information about the employees and various schedule rules along with the days employees would like to not be scheduled (Time off requests). Build a web based application which will determine a shift schedule for June 2015 (weeks 23 through 26).

- Users should be able to view the list of employees
- Clicking on an employee name should display a visual representation of the employee's schedule for the month
- Time off requests which violate the shift-rules should be ignored
- Employee specific shift rules override general rules
- In this case, a “shift” is simply 1 day

If you run into anything ambiguous feel free to make a decision on behalf of the customer. Just be sure to make a note of all assumptions you make in your README or code comments.

The Assignment

The problem has been broken down into various features. Implement features 1 and 2 to complete this technical assessment interview step. When finished, submit your solution to the API. Package up your code as described below under “Submitting Your Results”.

Bonus marks: Describe how you’d go about implementing features 3 and 4.
If you really want to impress us: Describe how you’d go about implementing features 5 & 6.

Features:

1. Implement the EMPLOYEES_PER_SHIFT rule (ignore time off requests)
2. Take into account employee time off requests
3. Implement the MAX_SHIFTS rule applying the corporate setting (I.E., ignore employee specific settings)
4. Implement the employee specific MAX_SHIFTS override
5. Implement the MIN_SHIFTS rule applying the corporate setting (I.E., ignore employee specific settings)
6. Implement the employee specific MIN_SHIFTS override

Note: The API will return *all* data for the full set of features. As you implement each feature you can ignore the data that does not apply to your feature set thus far.

Submitting Your Results

To submit your results:

- Post your shift schedule to the /submit route as described in The API.
- If possible, host your application so we can see it
- Email your code to the recruiting agent. Include the url to your hosted application if applicable. Include in this email your description of how to implement the remaining features.

The API

Root url: http://interviewtest.replicon.com/

/employees - GET
Returns the list of employees available for scheduling

/employees/:id - GET
Returns the information of the employee specified by :id

/time-off/requests - GET
Returns the list of time off requests for weeks 23 through 26
Note: Weekdays are indexed 1 through 7 where 1 is Monday and 7 is Sunday

/weeks - GET
Returns the information about all weeks for 2015

/weeks/:week_number - GET
Returns the details of the week specified by :week_number

/rule-definitions - GET
Returns the definitions of the various schedule rules in the system

/shift-rules - GET
Returns the current system settings for the various rule-definitions

/submit - POST
Submit your shift schedule. Will respond with a json object which contains the submission we received.
Body:
JSON representation of your shift schedule. See Shift Schedule Format below.
Required query parameters:
name - Your name so we can match you up with your resume, etc.
email - Your email address
features - An array identifying which features you have implemented. For example, “&features[]=1&features[]=2”
Optional query parameters:
solution - Set to true when you want to submit your solution for consideration. Not setting this parameter or setting it to any other value allows you to test posting your solution without officially submitting it to us for evaluation. Please note, there will be no indication provided regarding the validity of your submission. Should you receive a 500 response you may submit your JSON in the email along with your code but it’s probably wise to review your data to be sure it’s in the correct format.

Shift Schedule Format

The expected shift schedule is an array of weekly schedule objects. An example shift schedule:

```
[
 {
   "week": 23,
   "schedules": [
     {"employee_id": 1, "schedule": [1,2]},  // schedule: array of days scheduled for employee
     {"employee_id": 2, "schedule": [3,4]},
     ...
   ]
 },
 ...
]
```