

Sample Problem

- Publish a repo on Github that i can clone and run on my local machine
- Use Java, Scala or C# to complete this.
 - For JVM - Don't use Spring Boot, Dropwizard or Scalatra to do this, but feel free to use the libraries employed by those frameworks (eg JAX-RS, Jersey, Metrics, Jackson, Akka, Jetty, etc)
 - For C# - Use .NET Core
- Unit tests need to be in place
- API will need documentation and a contract published.
- It should support JSON & XML over HTTP
 - Bonus points for Protocol Buffers
- I should be able to curl against an API that computes a price for a specified datetime range given a JSON file of rates.
- Optional for Junior - Integration tests need to be in place.
- Optional for Mid to Junior - I should also be able to publish that app to AWS/Azure/Google Cloud very simply.
- Optional for Junior - Metrics for endpoint(s) captured and available to be queried via an endpoint (eg average response time)
- Sample file:

```
{
  "rates": [
    {
      "days": "mon,tues,wed,thurs,fri",
      "times": "0600-1800",
      "price": 1500
    },
    {
      "days": "sat,sun",
      "times": "0600-2000",
      "price": 2000
    }
  ]
}
```

- Sample result:

Datetime ranges should be specified in [isoformat](#). A rate must completely encapsulate a datetime range for it to be available.

Rates will never overlap.

2015-07-01T07:00:00Z to 2015-07-01T12:00:00Z should yield 1500

2015-07-04T07:00:00Z to 2015-07-04T12:00:00Z should yield 2000

2015-07-04T07:00:00Z to 2015-07-04T20:00:00Z should yield unavailable

- Sample JSON for testing

```
{
  "rates": [
    {
      "days": "mon,tues,thurs",
      "times": "0900-2100",
      "price": 1500
    },
    {
      "days": "fri,sat,sun",
      "times": "0900-2100",
      "price": 2000
    },
    {
      "days": "wed",
      "times": "0600-1800",
      "price": 1750
    },
    {
      "days": "mon,wed,sat",
      "times": "0100-0500",
      "price": 1000
    },
    {
      "days": "sun,tues",
      "times": "0100-0700",
      "price": 925
    }
  ]
}
```