Valentines day is arriving and a new special landing page will be created on our Website <http://cheapawesome.travel/>.

As part of the connectivity team, you have been tasked to create a service that will consume the hotel avail api from our new supplier “Bargains for Couples”.

The request is quite simple: [https://webbedsdevtest.azurewebsites.net/api/findBargain?**destinationId**=1419&**nights**=2&**code**=aWH1EX7ladA8C/oWJX5nVLoEa4XKz2a64yaWVvzioNYcEo8Le8caJw==](https://webbedsdevtest.azurewebsites.net/api/findBargain?destinationId=1419&nights=2&code=aWH1EX7ladA8C/oWJX5nVLoEa4XKz2a64yaWVvzioNYcEo8Le8caJw==)

Where we have the params destinationId, number of nights and our secret authentication code.

The goal is to consume this api and return a list of hotels with 1 final price per boardType with response time under 1 second.

The response  will contain a list of hotels, with rates. The rates have a boardType, value and rate type (PerNight or Stay). If rate type is PerNight, you have to calculate the final price (value \* numberOfNigths). If Stay, value is already the final price.

Prices and availability from the supplier are fairly static and are only likely to change every few hours.

**The “Bargains api” has some surprises for the devs:**

* 30% of chance of being slow (random sleep between 2sec and 5sec)
* 10% of chance of returning hotel with inconsistent data (blank hotel name)
* 10% of chance of returning inconsistent rate data (price value = -1)
* 10% of chance of an internal server error

**What we will evaluate:**

* General architecture: organization, components design and responsibility, testability, unit tests, data structures, including request and response objects.
* Correctness/Functionality: list of hotels where each hotel contains static data, 1 price, per board. Also we want to see how devs handle unexpected/wrong values like negative prices or empty hotel names.
* Resilient coding: request shouldn’t fail when supplier fails  and other general error handling (invalid params for example). Should return a controlled empty response
* Request lifecycle management : as we expect response time under 1 second, we want to see how they will handle. Again, we shouldn’t receive an error but a controlled empty response. Also will be interesting to see resource management.
* Security: secure traffic, secret management

Supplier API response:

[   
   {   
      **"hotel"**:{   
         **"propertyID"**:79732,  
         **"name"**:"JAC Canada (CA$)8314",  
         **"geoId"**:279,  
         **"rating"**:3  
      },  
      **"rates"**:[   
         {   
            **"rateType"**:"PerNight",  
            **"boardType"**:"No Meals",  
            **"value"**:207.6  
         },  
         {   
            **"rateType"**:"PerNight",  
            **"boardType"**:"Half Board",  
            **"value"**:242.2  
         },  
         {   
            **"rateType"**:"PerNight",  
            **"boardType"**:"Full Board",  
            **"value"**:276.8  
         }  
      ]  
   },  
   {   
      **"hotel"**:{   
         **"propertyID"**:79821,  
         **"name"**:"JAC Canada (CA$)8555",  
         **"geoId"**:279,  
         **"rating"**:3  
      },  
      **"rates"**:[   
         {   
            **"rateType"**:"Stay",  
            **"boardType"**:"No Meals",  
            **"value"**:590.4  
         },  
         {   
            **"rateType"**:"Stay",  
            **"boardType"**:"Half Board",  
            **"value"**:688.8  
         },  
         {   
            **"rateType"**:"Stay",  
            **"boardType"**:"Full Board",  
            **"value"**:787.2  
         }  
      ]  
   }  
]