

# Junior Node Test Task

## Background

We want to develop a ticket-selling platform.

The front-end part of the application is not yet ready, so feel free to design the API however you want.

### High-level features

1. Get info about an event
2. Get info about available tickets
3. Purchase ticket

### Feature requirements

*Get info about an event*

1. Event has a name
2. Event has a date and a time

*Get info about available tickets*

1. We should be able to receive information about what quantity
2. of tickets is available.

*Pay for ticket*

1. For the sake of simplicity we operate only on the `EUR` currency. Fee free to use
2. the provided adapter.

### Additional info

1. We really don't want to push you in any certain direction, but if you've
2. made certain assumptions how it should work with FE then please let us know.
3. TL;DR - you can briefly describe how the whole app would interact.
4. Please use a relational (SQL) database such as Postgres, MySQL or SQLite.
5. Tests would be appreciated but are not required.

```
// PAYMENT ADAPTER
```

```
class PaymentGateway {  
  charge(amount, token, currency = 'EUR') {  
    return new Promise((resolve, reject) => {  
      switch (token) {
```

```
        case 'card_error':
            return reject(new Error('Your card has been decline
d.'));
        case 'payment_error':
            return reject(new Error('Something went wrong with y
our transaction.'));
        default:
            resolve({ amount, currency });
    }
})
}
}

module.exports = PaymentGateway;
```