

# Testing with Microservices

## Whats the scope of our tests?

Test a single piece of code in isolation

*Example*  
Single middleware

Test how different pieces of code work together

Request flowing through multiple middlewares to a request handler

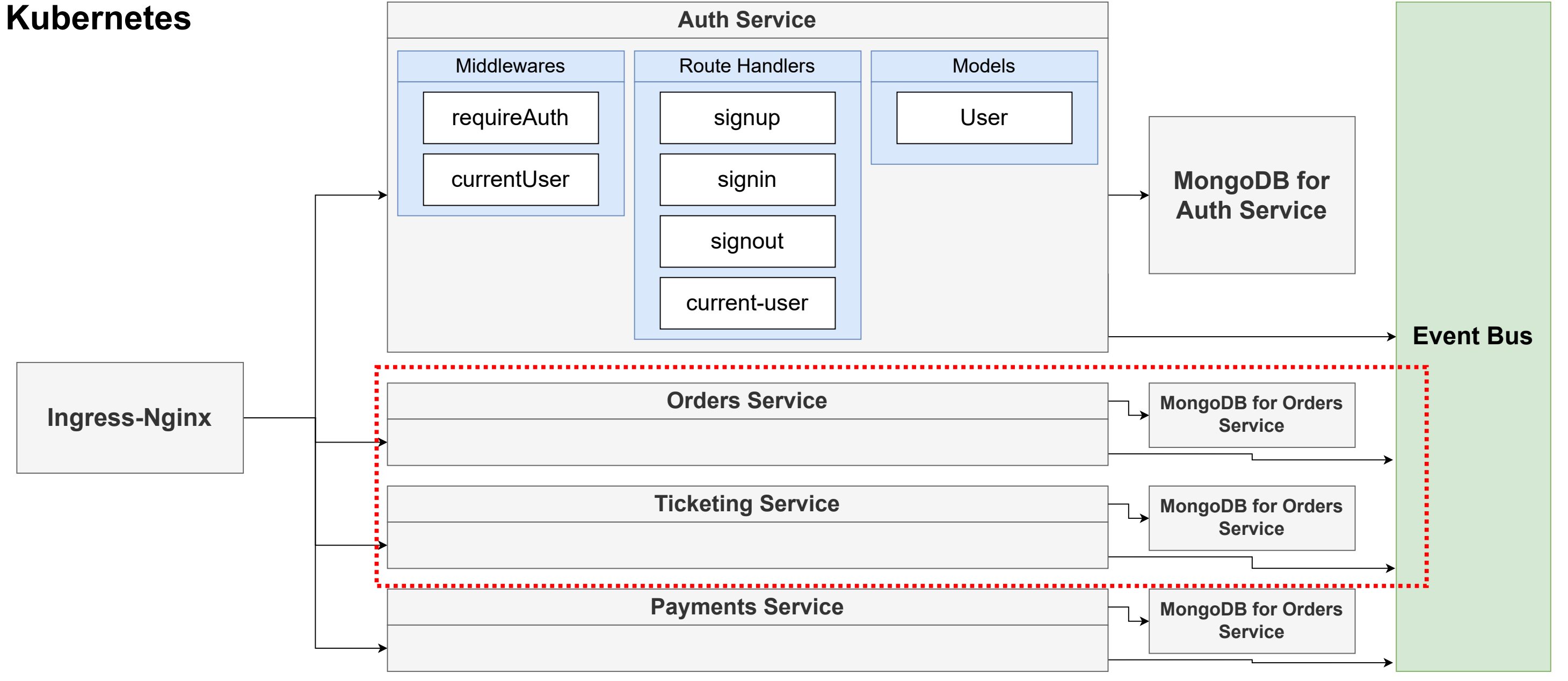
Test how different components work together

Make request to service, ensure write to database was completed

Test how different services work together

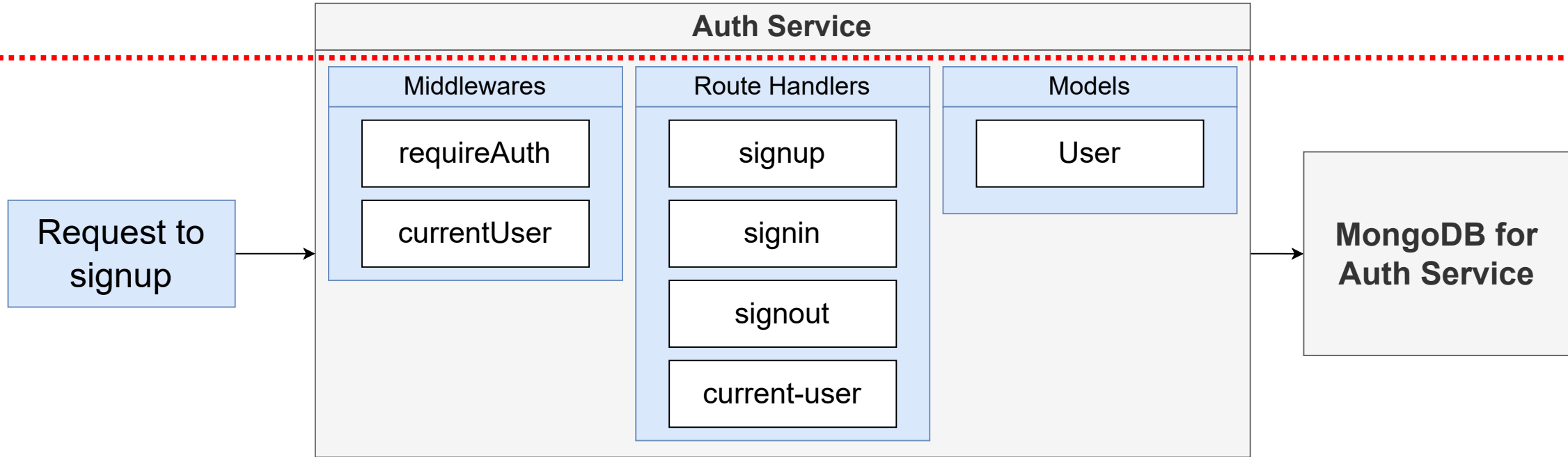
Creating a 'payment' at the 'payments' service should affect the 'orders' service

# Kubernetes



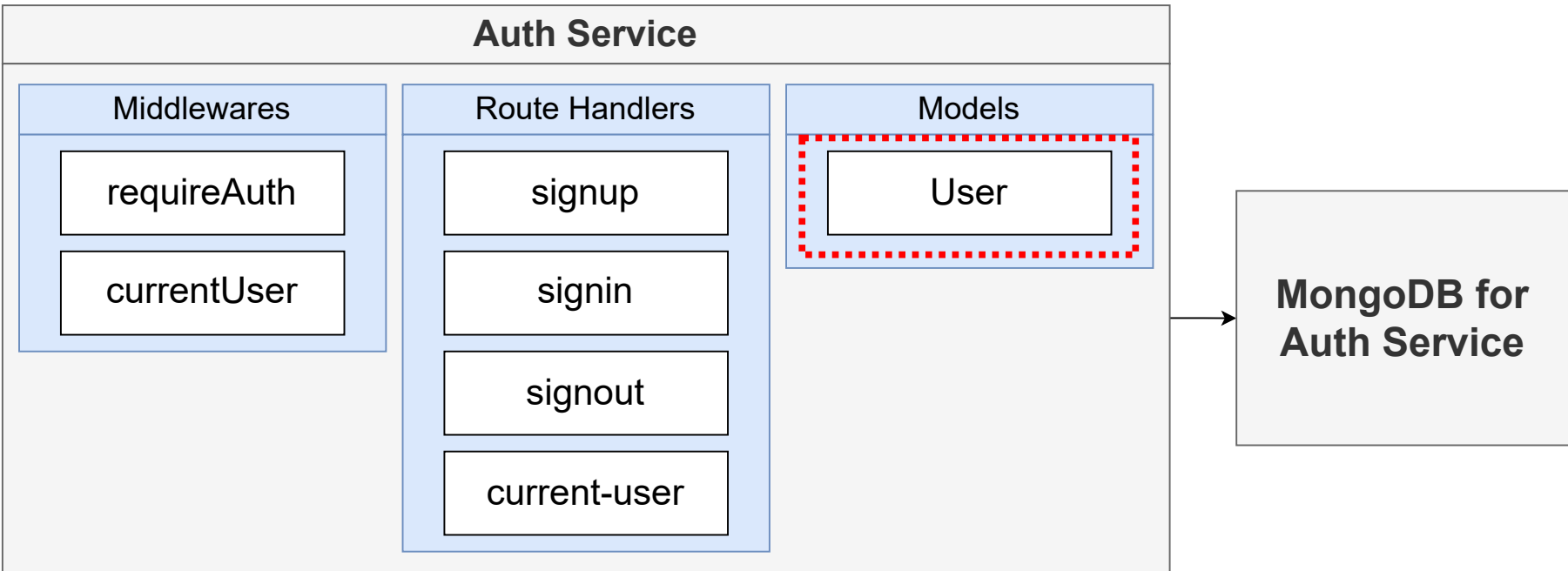
## Testing Goal #1

# Basic Request Handling



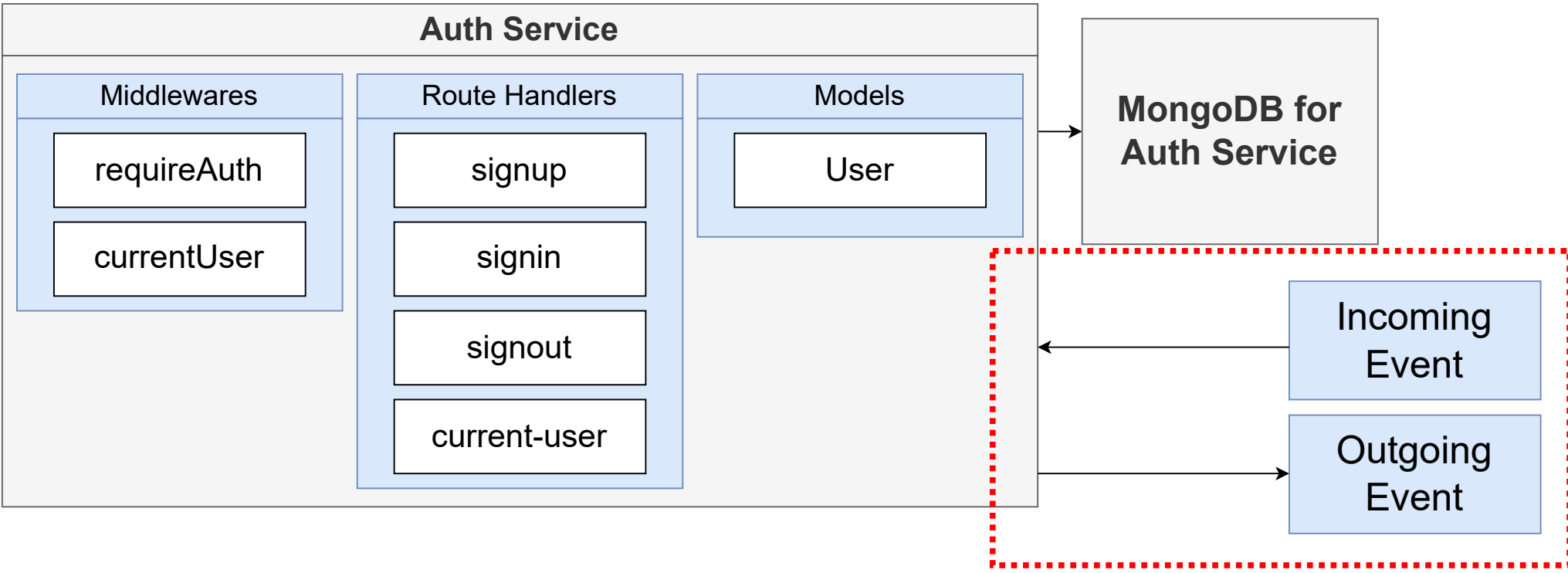
## Testing Goal #2

# Some tests around models



## Testing Goal #3

# Event emitting + receiving



# How will we run these tests?

We are going to run these tests directly from our terminal without using docker



This implies that our *local environment* is capable of running each service!



Simple enough now, but more *complex projects* might make this hard!

**npm run test**



**Jest**

Start in-memory copy of MongoDB

Start up our express app

Use supertest library to make fake requests  
to our express app

Run assertions to make sure the request did  
the right thing



Code to create  
Express server (this  
is in our index.js file)

```
const request = require('supertest');  
const express = require('express');  
  
const app = express();  
  
app.get('/user', function(req, res) {  
  res.status(200).json({ name: 'john' });  
});
```

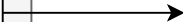
Code to test the  
express server (in  
test file)

```
request(app)  
  .get('/user')  
  .expect('Content-Type', /json/)   
  .expect('Content-Length', '15')  
  .expect(200)  
  .end(function(err, res) {  
    if (err) throw err;  
  });
```

# Our Current Setup

**index.js**

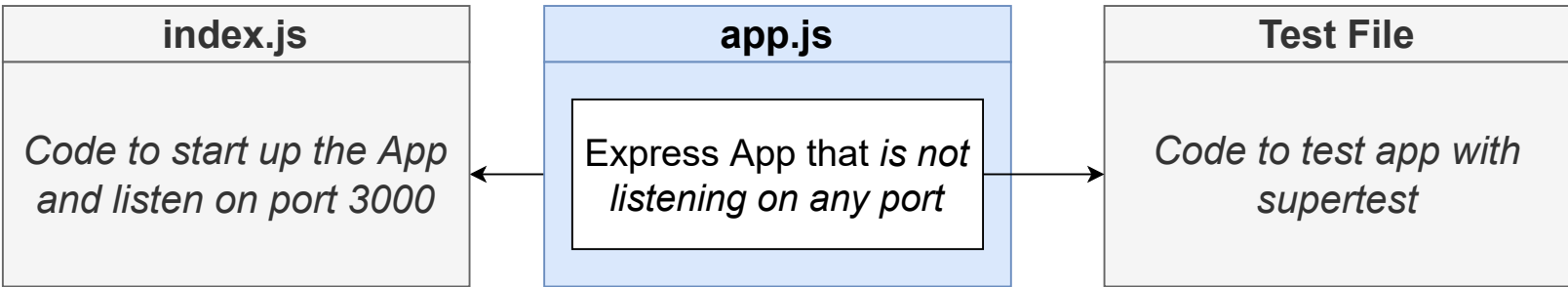
*Express App already  
listening on port 3000*



**Test File**

*Code to test app with  
supertest*

# Ideal Setup



Executed only when we want to run our app in a dev/prod environment