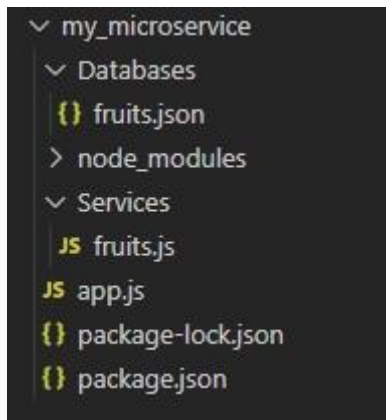


Практическая работа №4

Создание микросервиса:

Структура:



В папке Services создаём файл fruits.js, в нём создаём маршрут, который будет отображать информацию о фруктах из нашей БД.

```
1  const express=require('express')
2  const fruits=require('../Databases/fruits.json')
3  const app=express()
4  app.get('/',(req,res)=>{
5    res.status(200).json("Hello world")
6  })
7
8  app.get('/fruits',(req,res)=>{
9    res.status(200).json(fruits)
10 })
11
12 app.get('/fruits/:fruitname',(req,res)=>{
13   const fruitName=req.params.fruitname
14   const fruitDetails=fruits.filter((fruit)=>{
15     return fruit.name.toLowerCase()===fruitName.toLowerCase()
16   })
17   fruitDetails.length==0 ? res.status(404).send(`${fruitName} not found in database :(`) : res.status(200).
18 })
19
20 module.exports.app=app
```

Создаём app.js – здесь будут импортироваться Services и слушаться порт 3000

```
my_microservice > JS app.js > ...
1  const {app}=require('./services/fruits')
2  app.listen(3000,()=>{
3    console.log("Server is running on PORT 3000")
4  })
```

Этот эндпоинт будет отображать hello world при запуске

```
app.get('/',(req,res)=>{  
  res.status(200).json("Hello world")  
})
```

А этот будет выводить информацию о фруктах

```
app.get('/fruits',(req,res)=>{  
  res.status(200).json(fruits)  
})
```

Результаты:

The top part of the image shows a web browser window with the address bar at `localhost:3000/`. The page content displays `"Hello world"` in a pink font. Below the browser, the Postman interface is shown. A new GET request is configured for `localhost:3000/fruits`. The response body is displayed in JSON format, showing an array of three fruit objects:

```
[  
  {  
    "name": "Apple",  
    "calories": "95cal"  
  },  
  {  
    "name": "Grapes",  
    "calories": "104cal"  
  },  
  {  
    "name": "Mango",  
    "calories": "202cal"  
  }  
]
```

The status of the response is 200 OK, with a time of 49 ms and a size of 463 B.

Сделано с помощью статьи: <https://medium.com/effective-developers/postman-%D0%BA%D0%B0%D0%BA-%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-api-6c0f76358cf2>