Servis Odaklı Mimari Projesi

Servis Odaklı Mimari dersi için geliştirilmiş bu proje programlama dili olarak Javascript tercih edilmiş olup server tarafı için Node.js kullanılmıştır. Proje içerisinde "client" ve "server" olmak üzere iki ayrı yapı bulunmaktadır. Bu yapıların detayları ve birbirleri ile ilişkileri/haberleşmeleri aşağıda detaylıca anlatılmıştır.

1. Servis

Servis yazımında sunucu tarafında çalışabildiği için Node.js tercih edilmiştir. Projede servis tarafının gerçekleştirmesi beklenen işlemler; client tarafından gelecek isteği almak ve istekle birlikte gelen parametreler ile işlemler gerçekleştirerek sonucu cevap olarak client'a iletmektir.

Kullanılan paketler:

- Gelen istekleri yönetmek/düzenlemek amacıyla Express.js paketi kurulmuştur.
- Node.js temelli geliştirilen bir sunucunun API'larını kullanabilmek için varsayılan 'Access-Control-Allow-Origin' güvenliğini aşabilmemiz gerekmektedir. Bu sebeple "cors" paketi kurulmuştur.
- Gelen isteklerin body kısımlarını json halde alabilmemiz için "body-parser" paketi kurulmuştur.

```
const express = require('express')
const bodyparser = require('body-parser')
const cors = require('cors');
```

Şekil 1.1: Paketlerin modüllerinin eklenmesi

Express'in middleware olarak cors ve body-parser paketlerini kullanması için gerekli kodlar aşağıdaki gibi yazılmıştır.

```
const app = express();

app.use(cors());
app.use(bodyparser.json({
  strict:false
}))
```

Şekil 1.2: Middleware

Port belirlenmiş ve bu portun dinlenmesi için listen metodu kullanılmıştır. Bu sayede sunucu oluşturulmuştur.

```
const port = 3000;

app.listen(port, () =>{
  console.log("listening to port"+port)
});
```

Şekil 1.3: Sunucunun Oluşturulması

Son olarak client'tan gelecek isteğin alınması ve bu isteğe cevap verilmesi için (get isteğine karşılık çalışacak) get metodu kullanılarak bir route belirlenmiş, istek içerisinden gelen parametreler alınmış ve bu parametreler toplanarak cevap olarak client'a iletilmiştir.

```
app.get('/', (req, res) => {
  const {num1, num2} =req.query;

  const result=parseInt(num1)+parseInt(num2);
  res.send(result.toString());
})
```

Şekil 1.4: Gelen isteğin karşılanması

2. Client

Client tarafında HTML sayfası içerisinde Javascript dili kullanılmıştır. Projede client tarafında sunucuya belirli iki sayı ile birlikte istek gönderilir ve dönen cevap karşılanır.

İstek atılabilmesi için 'axios' paketi cdn ile projeye eklenmiştir.

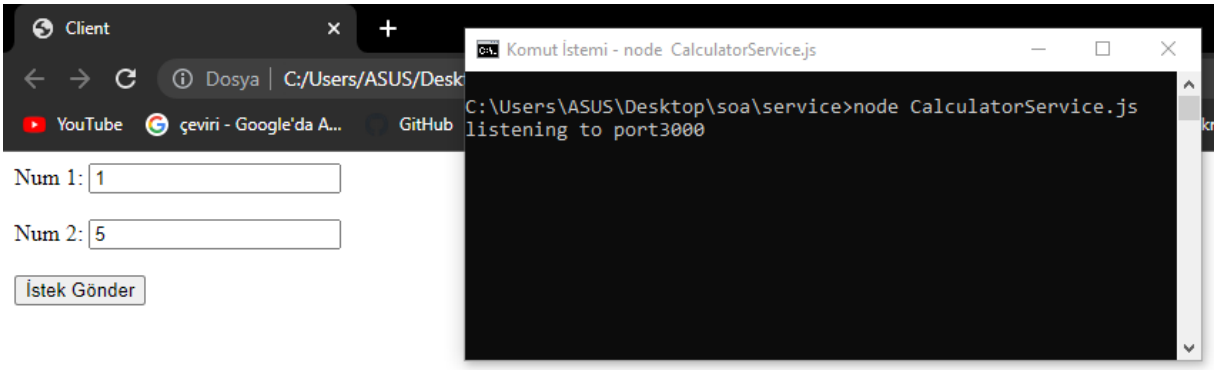
İki sayının kullanıcıdan alınması için iki adet input girişi verilmiştir. İstek yollanabilmesi için button koyulmuş ve bu buton'a tıklandığında sunucuya istek atacak ve cevabı döndürecek getResult() metodunun çalışması sağlanmıştır.

client > index.html > ...

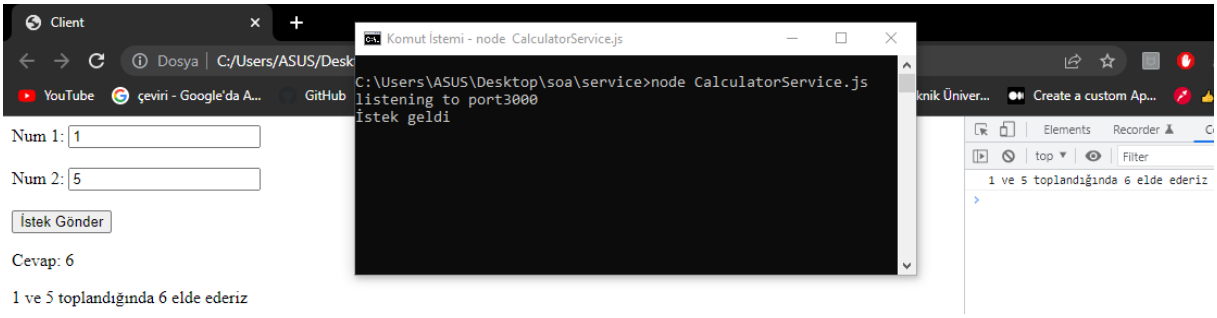
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Client</title>
</head>
<body>
    <!-- <p>İstek gönderildi. Cevap için konsola bakınız</p> -->
    Num 1: <input type="text" name="fname" value="1" id="num1"><br><br>
    Num 2: <input type="text" name="lname" value="5" id="num2"><br><br>

    <button type="submit" onClick="getResult()">İstek Gönder</button>
    <p id="result"></p>
    <p id="conclusion"></p>
</body>
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>

    const getResult = async () => {
        let num1 = document.getElementById("num1").value;
        let num2 = document.getElementById("num2").value;
        const result = await axios.get('http://127.0.0.1:3000/', {params : {num1, num2}})
        document.getElementById("result").innerHTML ="Cevap: "+ result.data;
        document.getElementById("conclusion").innerHTML =`${num1} ve ${num2} toplandığında ${result.data} elde ederiz`
        console.log(`${num1} ve ${num2} toplandığında ${result.data} elde ederiz`);
    }
</script>
</html>
```

Şekil 2.1: index.html

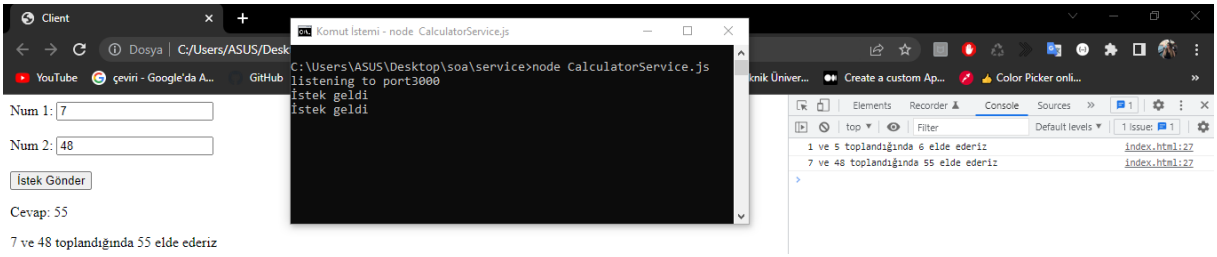
3. Uygulama Görselleri



Şekil 3.1: İstek Gönderilmeden Önce Client / Sunucu Çalışır Durumda



Şekil 3.2: İstek Gönderildikten Sonra



Şekil 3.3: İstek Gönderildikten Sonra

Tüm projeye https://github.com/aslihanturkdonmez/soaProje adresinden erişebilirsiniz.

Aslıhan Türkdönmez

18360859040