|  |
| --- |
|  |
| **NODE.JS İLE REST APİ SERVİSLERİ** |

**GENEL AÇIKLAMA**

**KULANILAN ARAÇLAR**

* **KODLAMA**
  + NODEJS
* **VERİTABANI**
  + MYSQL
* **TEST ARACI**
  + POSTMAN

**VERİ TABANI KODLARI**

CREATE DATABASE IF NOT EXISTS `nodejs` DEFAULT CHARACTER SET utf8 COLLATE utf8\_turkish\_ci;

USE `nodejs`;

CREATE TABLE IF NOT EXISTS `users` (

  `id` int(11) NOT NULL,

  `username` varchar(50) NOT NULL,

  `password` varchar(255) NOT NULL,

  `email` varchar(100) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8;

INSERT INTO `users` (`username`, `password`, `email`) VALUES ('admin', '123456', 'admin@admin.com');

ALTER TABLE `users` ADD PRIMARY KEY (`id`);

ALTER TABLE `users` MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=2;

Gerekli kütüphaneler:

* mysql
* express-session
* express

**PROJEDE 5 ADET ENDPOİN BULUNMAKTA BUNLAR:**

1. **GETALL**: BÜTÜN KULLANICILARI GETİRİR
2. **GETID**: ID Sİ VERİLEN KULLANCILARI GETİRİR
3. **POST**: VERİLEN KULLANICIYI VERİTABANINA KAYDEDER
4. **PUT**: VERİLEB ID DE KULLANICIYI GÜNCELLER
5. **DELETE**: VERİLEN ID KULLANICIYI SILER

**POSTMAN KULLANARAK BUNLARI TEK TEK TEST EDELİM:**

1. **GETALL VE GETID**

**KODLAR**

//GETALL

app.get('/', function(request, response) {

    connection.query('SELECT \* FROM users', function(error, results, fields) {

        const responseData = {

            users: results

        }

        const jsonContent = JSON.stringify(responseData);

        response.end(jsonContent);

    });

});

//GETID

app.get('/:id', function(request, response) {

    var userId = request.params.id;

    connection.query('SELECT \* FROM users WHERE id = ?', [userId], function(error, results, fields) {

        if (results.length > 0) {

            const responseData = {

                user: results

            }

            const jsonContent = JSON.stringify(responseData);

            response.end(jsonContent);

        } else {

            const responseData = {

                message: "KULLANICI BULUNAMADI",

                code: 404,

            }

            const jsonContent = JSON.stringify(responseData);

            response.end(jsonContent);

        }

    });

});

GETALL=><http://localhost:8088/>

POSTMANDE URLDEN GET İLE BÜTÜN KULLANICILARI GETİRİYORUZ

metin, ekran görüntüsü, ekran, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

**JSON ÇIKTISI**

{

"users": [

{

"id": 18,

"username": "MURAT",

"password": "123456",

"email": "admin@admin.com"

},

{

"id": 19,

"username": "GÖKAY",

"password": "123456",

"email": "admin@admin.com"

},

{

"id": 20,

"username": "YUNUS EMRE",

"password": "123456",

"email": "admin@admin.com"

}

]

}

GETID=><http://localhost:8088/>ID

POSTMANDE URLDEN GET İLE 18 İD YE SAHİP KULLANICIYI GETİRELİM

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**JSON ÇIKTISI**

{

"user": [

{

"id": 18,

"username": "MURAT",

"password": "123456",

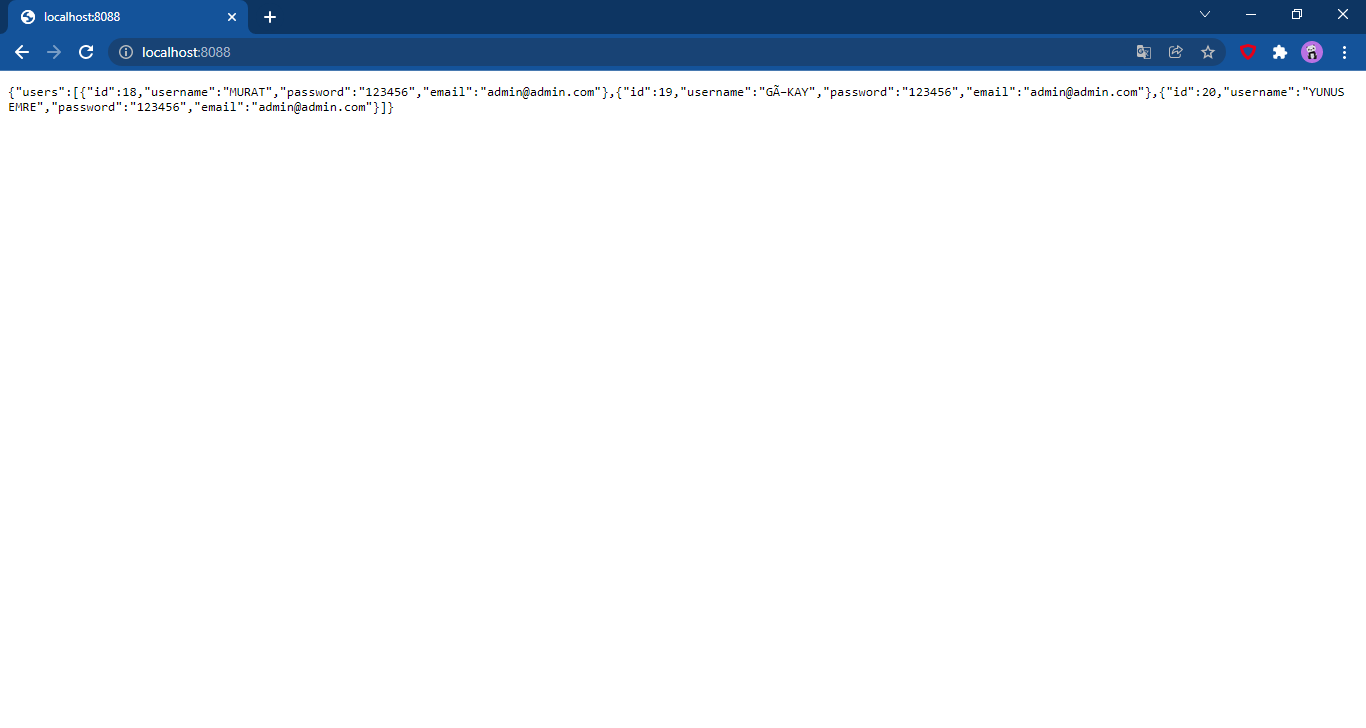
"email": "admin@admin.com"

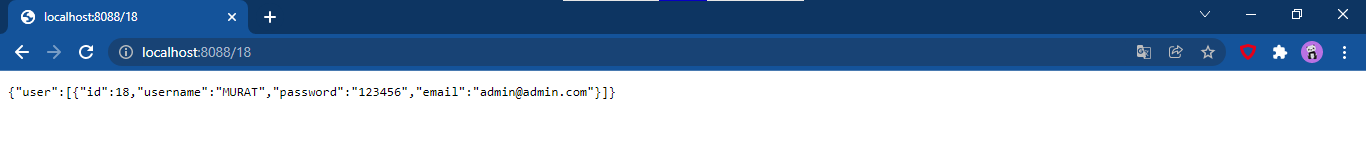
}

]

}

**GETALL VE GETID TARAYICI GÖRÜNTÜLERİ**



****

1. **POST USER (CRUD)**

**KODLAR**

//POST USER

app.post('/', function(request, response) {

    var username = request.body.username;

    var password = request.body.password;

    var email = request.body.email;

    connection.query('INSERT INTO users (username, password,email) VALUES (?,?,?)', [username, password, email], function(error, results, fields) {

        const responseData = {

            message: "KAYIT BAŞARILI",

            code: 200,

            endingMessage: "KAYIT İD:" + results.insertId

        }

        const jsonContent = JSON.stringify(responseData);

        response.end(jsonContent);

    });

});

POST(USER)=><http://localhost:8088/>

POSTMENDE BODYSİNİ(USER) EKLEYEREK POST EDİYORUZ

BODY:

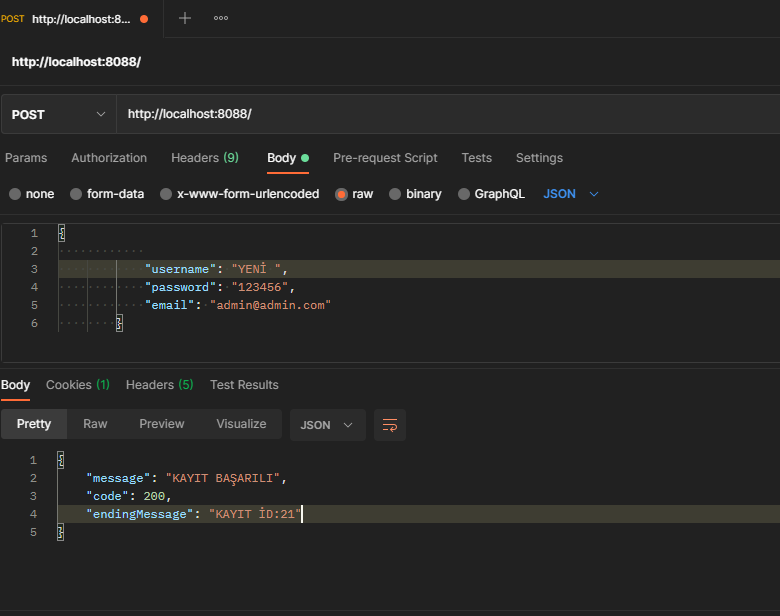
{

            "username": "YENİ ",

            "password": "123456",

            "email": "admin@admin.com"

        }



**JSON CIKTISI**

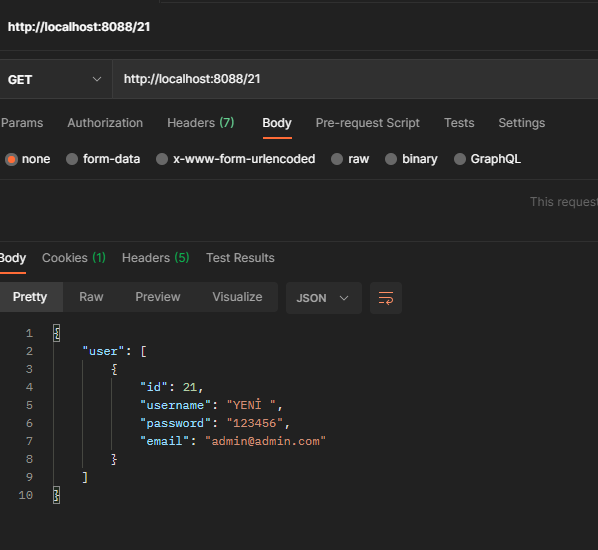
{

"message": "KAYIT BAŞARILI",

"code": 200,

"endingMessage": "KAYIT İD:21"

}

KAYIT OLAN KULLNICININ ID 21 ŞİMDİ SORGU YAPALIM:

1. **PUT USER (UPDATE)**

**KODLAR**

**//PUT USER**

**app.put('/:id', function(request, response) {**

**var userId = request.params.id;**

**var username = request.body.username;**

**var password = request.body.password;**

**var email = request.body.email;**

**connection.query('SELECT \* FROM users WHERE id = ?', [userId], function(error, results, fields) {**

**if (results.length > 0) {**

**for (var i = 0; i < results.length; i++) {**

**if (email) {**

**} else {**

**email = results[i].email;**

**}**

**if (username) {**

**} else {**

**username = results[i].username;**

**}**

**if (password) {**

**} else {**

**password = results[i].password;**

**}**

**}**

**connection.query('UPDATE users SET  username = ?, password = ?, email = ?  WHERE id =? ', [username, password, email, userId], function(error, results, fields) {**

**if (error) {**

**const responseData = {**

**message: "GÜNCELLEM BAŞARISIZ",**

**code: 500,**

**er: error**

**}**

**const jsonContent = JSON.stringify(responseData);**

**response.end(jsonContent);**

**} else {**

**const responseData = {**

**message: "GÜNCELLEM BAŞARILI",**

**code: 200,**

**}**

**const jsonContent = JSON.stringify(responseData);**

**response.end(jsonContent)**

**}**

**});**

**} else {**

**const responseData = {**

**message: "GÜNCELLEM BAŞARISIZ",**

**code: 404,**

**Error: "KULLANICI BULUNAMADI"**

**}**

**const jsonContent = JSON.stringify(responseData);**

**response.end(jsonContent);**

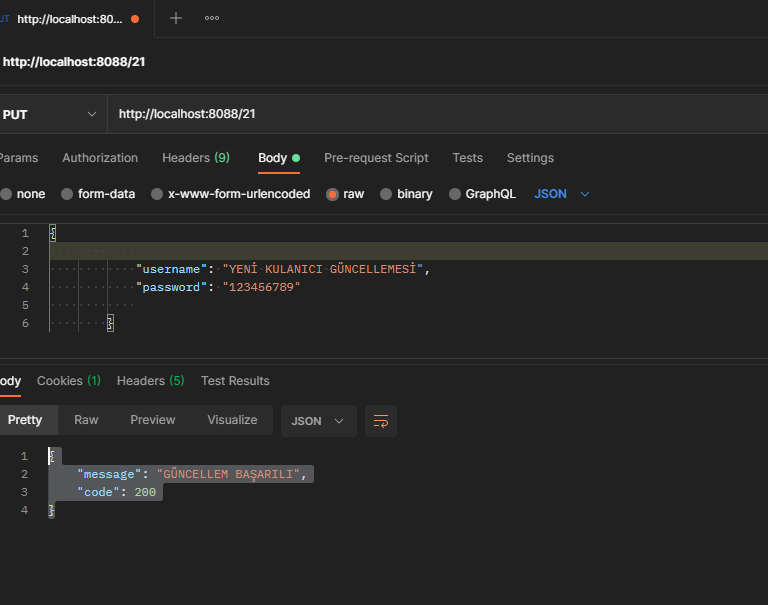
**}**

**});**

**PUT (USER BODY) =>**<http://localhost:8088/>ID

**POSTMANDE BODY KISMI VE ID SİNİ VEREREK İSTEDİĞİMİZ KISIMLARI GÜNCELİYORUZ**

**BİR ÖNCEKİ KISIMDA KAYDETTİĞİMİZ 21 ID Lİ KULANICININ ADINI VE ŞİFRESİNİ DEĞİŞTİRELİM**

****

**JSON CIKTISI**

{

"message": "GÜNCELLEM BAŞARILI",

"code": 200

}

**KONTROL EDELİM**

GETID=><http://localhost:8088/21>

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. **DELETE USERİD**

**KODLAR**

**// DELETE USER**

app.delete('/:id', function(request, response) {

    var userId = request.params.id;

    connection.query('SELECT \* FROM users WHERE id = ?', [userId], function(error, results, fields) {

        if (results.length > 0) {

            connection.query('DELETE FROM users WHERE id = ?', [userId], function(error, results, fields) {

                if (error) {

                    const responseData = {

                        message: "SİLME İŞLEMİ BAŞARISIZ",

                        code: 500,

                        er: error

                    }

                    const jsonContent = JSON.stringify(responseData);

                    response.end(jsonContent);

                } else {

                    const responseData = {

                        message: userId + " idli kulanıcı silindi",

                        code: 200,

                    }

                    const jsonContent = JSON.stringify(responseData);

                    response.end(jsonContent);

                }

            });

        } else {

            const responseData = {

                message: "SİLME BAŞARISIZ",

                code: 404,

                Error: "KULLANICI BULUNAMADI"

            }

            const jsonContent = JSON.stringify(responseData);

            response.end(jsonContent);

        }

    });

});

**POST (USER ID) =>**<http://localhost:8088/>ID

**POSTMANDE ID SİNİ VEREREK İSTEDİĞİMİZ KİŞİYİ SİLİYORUZ**

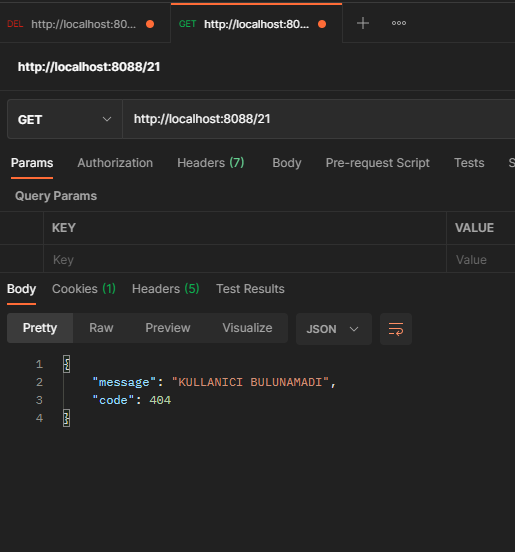
**BİR ÖNCEKİ KISIMDA KAYDETTİĞİMİZ 21 ID Lİ KULANICIYI SİLELİM**

**metin, ekran görüntüsü, ekran, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu**

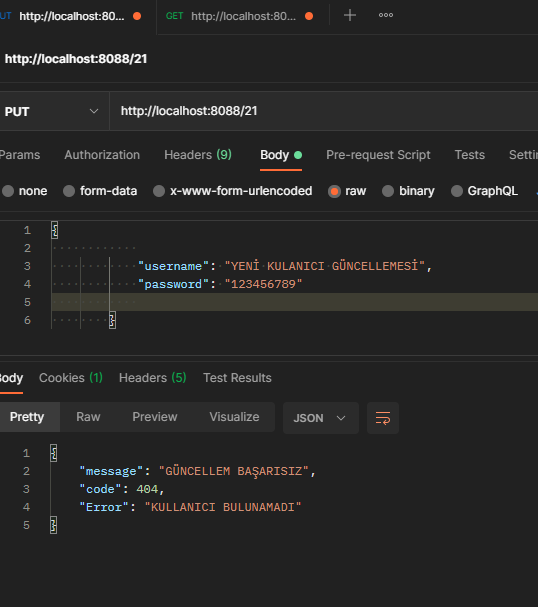
**KONTROL EDELİM**

GETID=><http://localhost:8088/21>

****

**DİĞER HATA ÇIKTILARI:**

**OLMAYAN KULLANICI GÜNCELLEMESİ:**

****

**OLMAYAN KULLANICI SİLİNMESİ:**

**metin, ekran görüntüsü, ekran, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu**