



Turgut Özal Üniversitesi Web Tasarımı ve Programlama

Eğitmen: Hamit Mızrak

HTML5



Java SE

- ▶ HTML5 nedir ?
- ▶ Hyper Text Markum Language: (işaretleme dilidir)
- ▶ işaretleme: Root / child

Heading

► h1 ... h6

Paragraf

► `<p>` `</p>`

hr

- horizontal: uzun çizgi

br

► break: </br>

formatter

- ▶ bold => b
- ▶ italic => i
- ▶ big => big
- ▶ small => small
- ▶ sup=> üst
- ▶ sub=> alt
- ▶ fosforlu kalem => mark
- ▶ kodlar => code
- ▶ üstüne çizmek => del
- ▶ altını çizmek => ins
- ▶ vurgulamak => em

div , span

- ▶ div: alt satıra geçen
- ▶ span: aynı satırda

liste

- ▶ unordered list: ul (düzensiz listeler)
- ▶ ordered list: ol (düzenli listeler)

image

► img : resim

link

► click me

table

- ▶ `<table>`
- ▶ `<thead> </thead>`
- ▶ `<tbody> </tbody>`
- ▶ `<tfoot> </tfoot>`
- ▶ `</table>`

form

- Form: bir veriyi bir yerden başka bir yere taşımaktır.

Araştırma:

- ▶ Semantic ile nonsemantic yapı arasındaki fark nedir ?
 - ▶ Ascii ile Unicode arasındaki farklar ?
 - ▶ CDN nedir ?
 - ▶ emmet.io => cheat
-
- ▶ Table: colspan ile rowspan arasındaki fark nedir ?
 - ▶ Form: get / post nedir

Kodlama Ödevimiz

Register Form

username

email

password

phone number



```
▶ h2{
▶   color:brown;
▶ }
▶ #h2_id{
▶   background-color: black;
▶ }
▶
▶ .h2_class{
▶   padding:2rem;
▶ }
```

```
▶ *{
▶     margin: 0;
▶     padding:0;
▶     box-sizing: border-box;
▶ }
▶ html{
▶     font-size: 20px;
▶ }
▶ body{
▶     font-family: 'Roboto Condensed', sans-serif;
▶ }
▶
```

```
▶ img{
▶     max-width: 100%;
▶     height: auto;
▶ }
▶
▶ img:hover{
▶     opacity: 0.8;
▶     transition: 2s all;
▶ }
▶
```

```
▶ #div_id{  
▶     border:1px solid blue;  
▶     margin-top:10px;  
▶     width: 400px;  
▶     height: 25rem;  
▶ }  
▶
```

```
▶ /* #div_id:hover{  
▶     cursor: pointer;  
▶     width: 800px;  
▶ } */
```

```
▶ #param_id2{
▶     color:red;
▶     color:#000;
▶     color:#4d4a4a;
▶     color:rgb(0,0,0);
▶     color:rgba(255, 255, 255, 0.8);
▶     background-color: black;
▶     padding:30px; /*padding:iç boşluk*/
▶     padding-top:30px;
▶     padding-right:30px;
▶     padding-bottom:30px;
▶     padding-left:30px;
▶     padding:10px 40px;
▶     margin:4px auto;
▶ }
```


HTML

```
▶ <div>
▶   <p>Lorem ips, harum eaque culpa in aliquam libero ducimus. Modi amet voluptates obcaecati molestias
nesciunt nisi ad eos provident quam consequuntur, animi architecto debitis dolor adipisci, dolore odio.</p>
▶   </div>
▶
▶   <div>
▶     <p>Lorem ips, harum eaque culpa in aliquam libero ducimus. Modi amet voluptates obcaecati molestias
nesciunt nisi ad eos provident quam consequuntur, animi architecto debitis dolor adipisci, dolore odio.</p>
▶     </div>
▶
▶     <div>
▶       <p>Lorem ips, harum eaque culpa in aliquam libero ducimus. Modi amet voluptates obcaecati molestias
nesciunt nisi ad eos provident quam consequuntur, animi architecto debitis dolor adipisci, dolore odio.</p>
▶       </div>
▶
▶       <div>
▶         <p>Lorem ips, harum eaque culpa in aliquam libero ducimus. Modi amet voluptates obcaecati molestias
nesciunt nisi ad eos provident quam consequuntur, animi architecto debitis dolor adipisci, dolore odio.</p>
▶         </div>
```

RD

```
▶ /*RD:Responsive Design*/
▶ @media screen and (min-width:1px) {
▶     div {
▶         width: 100%;
▶         float:left;
▶         background-color: aqua;
▶     }
▶ }
▶
▶ @media screen and (min-width:600px) {
▶     div {
▶         width: 50%;
▶         float:left;
▶         background-color: rgb(24, 27, 175);
▶     }
▶ }
```

```
▶ @media screen and (min-width:900px) {  
▶     div {  
▶         width: 33%;  
▶         float:left;  
▶         background-color: rgb(104, 175, 24);  
▶     }  
▶ }  
▶  
▶ @media screen and (min-width:1200px) {  
▶     div {  
▶         width: 25%;  
▶         float:left;  
▶         background-color: rgb(183, 224, 180);  
▶     }  
▶ }
```

BOOTSTRAP



Bootstrap

- ▶ Bootstrap açık kaynak kodlu, ücretsiz bir CSS frameworktür.
- ▶ Bootstrap tasarım aracıdır.
- ▶ Bootstrap yazılımcının resim yapma sanatıdır.

navbar

```
<nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top">
  <div class="container">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler d-lg-none" type="button" data-bs-toggle="collapse" data-bs-target="#collapsibleNavId" aria-controls="collapsibleNavId"
      aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="collapsibleNavId">
      <ul class="navbar-nav me-auto mt-2 mt-lg-0">
        <li class="nav-item">
          <a class="nav-link active" href="#" aria-current="page">Home <span class="visually-hidden">(current)</span></a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#" data-bs-toggle="modal" data-bs-target="#modalId">Register</a>

        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="dropdownId" data-bs-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Dropdown</a>
          <div class="dropdown-menu" aria-labelledby="dropdownId">
            <a class="dropdown-item" href="#">Action 1</a>
            <a class="dropdown-item" href="#">Action 2</a>
          </div>
        </li>
      </ul>
      <form class="d-flex my-2 my-lg-0">
        <input class="form-control me-sm-2" type="text" placeholder="Search">
        <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
      </form>
    </div>
  </div>
</nav>
```

carousel

```
<div id="carouselId" class="carousel slide" data-bs-ride="carousel">
  <ol class="carousel-indicators">
    <li data-bs-target="#carouselId" data-bs-slide-to="0" class="active" aria-current="true" aria-label="First slide"></li>
    <li data-bs-target="#carouselId" data-bs-slide-to="1" aria-label="Second slide"></li>
    <li data-bs-target="#carouselId" data-bs-slide-to="2" aria-label="Third slide"></li>
  </ol>
  <div class="carousel-inner" role="listbox">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-target="#carouselId" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-target="#carouselId" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
```

breadcrumb

```
<nav class="breadcrumb">
  <a class="breadcrumb-item" href="#">Main</a>
  <a class="breadcrumb-item" href="#">Sub</a>
  <span class="breadcrumb-item active" aria-current="page">Active</span>
</nav>
```


alert

```
<div class="alert alert-secondary alert-dismissible fade show" role="alert">  
  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>  
  
  <strong>Holy guacamole!</strong> You should check in on some of those fields below.  
</div>
```

spinner

```
<div class="d-flex justify-content-center align-items-center">  
  <div class="spinner-grow text-primary spinner-border-lg"  
    role="status">  
    <span class="visually-hidden">Loading...</span>  
  </div>  
</div>
```

accordion

```
<div class="accordion" id="accordionExample">
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingOne">
      <button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne">
        Accordion Item #1
      </button>
    </h2>
    <div id="collapseOne" class="accordion-collapse collapse show" aria-labelledby="headingOne" data-bs-parent="#accordionExample">
      <div class="accordion-body">
        This is the first item's accordion body.
      </div>
    </div>
  </div>
</div>
```

modal

```
<!-- Modal Body -->
<!-- if you want to close by clicking outside the modal, delete the last endpoint:data-bs-backdrop and data-bs-keyboard -->
<div class="modal fade" id="modalId" tabindex="-1" data-bs-backdrop="static" data-bs-keyboard="false" role="dialog" aria-labelledby="modalTitleId" aria-hidden="true">
  <div class="modal-dialog modal-dialog-scrollable modal-dialog-centered modal-lg" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="modalTitleId">Modal title</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        Body
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save</button>
      </div>
    </div>
  </div>
</div>
</div>
```

pagination

```
<nav aria-label="Page navigation">
  <ul class="pagination">
    <li class="page-item disabled">
      <a class="page-link" href="#" aria-label="Previous">
        <span aria-hidden="true">&laquo;</span>
      </a>
    </li>
    <li class="page-item active" aria-current="page"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item">
      <a class="page-link" href="#" aria-label="Next">
        <span aria-hidden="true">&raquo;</span>
      </a>
    </li>
  </ul>
</nav>
```



JavaScript nedir ?

- ▶ Html nedir? (Content oluşturmak)
- ▶ CSS nedir? (Makyaj görseiliği daha güzelleştirmek için)
- ▶ JavaScript nedir? (Dynamics web tasarımlarda kullanacağız)
- ▶ OOP

1. JavaScript 1995 yılında
2. Brendan Eich tarafından geliştirilmiş
3. web sayfalarda dinamikleştirmek ,
4. tarayıcı üzerinden çalışan , betik bir dildir.
5. JavaScript bir çok browserda çalışır.
6. Java ile JavaScript aynı değildir.

JavaScript Kullanan Firmalar

- Google. ...
- eBay. ...
- Facebook. ...
- Groupon. ...
- LinkedIn. ...
- Microsoft. ...

variable

```
// comment
/* multiple comment*/

// alert("deneme")
// window.alert("deneme44")
// window.document.write("ana ekran"),
// console.log("console")
// console.warn("console")
// console.error("console")
// console.info("console")

var deneme1 = 44;
var deneme1 = 55;

// var let const farklar ?
//Ecmascript nedir ?
//React: +ES6

// let result=prompt("isminizi giriniz");
// console.log(result)

// class

let $_valueName44 = 44;
console.log($_valueName44)
console.log(typeof $_valueName44)

// whoisting: dikkat var için kullan
deneme = 10 + 20;
var deneme;
console.log(deneme)
```

string

```
//cast
// let data1=Number(prompt("sayı giriniz"));
// console.log(typeof data1)

// NaN undefined infinity
// NaN
console.log(4 / "asd")

// infinity
console.log(4 / 0)

let data;
console.log(data)

//escape character
// \' \'
// \" \"

//String
let vocabulary = "Javascript öğreniyorum Javascript ";
console.log(vocabulary)
console.log(vocabulary.toLowerCase())
console.log(vocabulary.toUpperCase())
console.log(vocabulary.length)
console.log(vocabulary.trim().length)
console.log(vocabulary.charAt(0))
console.log(vocabulary.indexOf("Javascript"))
console.log(vocabulary.lastIndexOf("Javascript"))
console.log(vocabulary.startsWith("J"))
console.log(vocabulary.endsWith(" "))
console.log(vocabulary.substring(1))
console.log(vocabulary.substring(1, 4)) //1<=X<=4-1
```

Math

```
//Math
console.log(Math.min(1, 100, 200))
console.log(Math.max(1, 100, 200))
console.log(Math.sqrt(100))
console.log(Math.abs(-100))
console.log(Math.pow(2, 5))
console.log(Math.sin(45))
console.log(Math.floor(5.9))
console.log(Math.ceil(5.1))
console.log(Math.round(7.4))
console.log(Math.round(7.5))
```

Fucntion

```
//function
function returnsuzParametresiz() {
    console.log("returnsuz parametresiz")
}
returnsuzParametresiz()

function returnsuzParametrelili(data) {
    console.log("returnsuz parametrelili " + data)
}
returnsuzParametrelili("kelime")

function returnluParametresiz() {
    return "returnluParametresiz"
}
let data5 = returnluParametresiz()
console.log(data5)

function returnluParametrelili(number) {
    return "returnluParametrelili " + number;
}
let data6 = returnluParametrelili()
console.log(data6)
////////////////////////
```

```
// normal function
function normal() {
  console.log("normal")
}
normal()

// anonymous function
let anonymous = function () {
  console.log("anonymous")
}
anonymous()

// arrow function
let arrow = () => {
  console.log("arrow")
}
arrow()
```

Conditional

```
//contional
let contional = () => {
  let data = 5;
  if (data > 0) {
    console.log("pozitif sayı")
  } else
    console.log("negatif sayı")
  //ternary ÖDEV
}
contional()

// = atama
// == eşitmi ancak türüne bakmadan
// === eşitmi ancak türüne bakarak
let contional2 = () => {
  let data = 5;
  if (data == 1) {
    console.log("1")
  } else if (data === 2) {
    console.log("2")
  } else if (data === 3) {
    console.log("3")
  } else if (data === 4) {
    console.log("4")
  } else if (data === 5) {
    console.log("5")
  } else
    console.log("negatif sayı")
  //ternary ÖDEV
}
contional2()
```


Loop

```
//loop
let loop = () => {
  for (let i = 0; i < 10; i++) {
    document.write(i + " ")
  }
}
loop()

document.writeln("<br/>")

let loop2 = () => {
  let i = 0;
  while (i < 10) {
    document.write(i + " ")
    i++
  }
}
loop2()

document.writeln("<br/>")
let loop3 = () => {
  let i = 0;
  do {
    document.write(i + " ")
    i++
  } while (i < 10);
}
loop3()
```

```
//ÖDEV
//break
//return
//continue

//ÖDEV (if-else for break continue)
//Fonksiyonlarla
//Örnek: 1 ile kullanıcının vereceği (prompt) bitiş sayısına göre
//kullanıcı: 5 verdi diyelim 1<=X<=5
//1.adım: kaç tane sayı var ?
//2.adım: sayı toplamları var ?
//3.adım: kaç tane tek sayı var ?
//4.adım: tek sayılar toplamı ?
//5.adım: tek sayılar gösterelim ?
//6.adım: kaç tane çift sayı var ?
//7.adım: çift sayılar toplamı ?
//8.adım: çift sayılar gösterelim ?
//Eğer verilen sayılarda 7 sayısı varsa bunu eklemesin (continue)
//Eğer bitiş sayısı 100 fazla ise 100'e kadar olanlar toplansın (break)
//Eğer kullanıcı başlangıç sayıdan küçük girerse uyaralım başlangıçtan büyük girmesini isteyelim
//Eğer kullanıcı secret-key girerse yani 44 sayısını girerse program çalışmayı direk durdursun
```

```
//ÖDEV
//NOT: örneklerimizi function,anonymous ve arrow function ile yapalım.
//NOT: değişken olarak var,let,const kullanalım.
//NOT: örneklerimizi function,anonymous ve arrow function ile yapalım.
//NOT: değişken olarak var,let,const kullanalım.
//Örnek-1
//y=3x+4k ==>1.dereceden2bilinmeyenlidenklem algoritması
//Kullanıcı tarafından alınan x ve k değerlerini hesaplayan algoritma yazınız ?
//örnek-2
//Kullanıcı tarafından alınan dereceyi Fahrenheit'a çeviren algoritma yapalım.
//Formül: (derece*9/5)+32
//örnek-3 operatör işlemleri: aşağıdaki örneği javascript ile yapalım
//4+3*2(3:3-1*6+9:1+(3:3))
```

```
//NOT: örneklerimizi function,anonymous ve arrow function ile yapalım.  
//NOT: değişken olarak var,let,const kullanalım.  
//NOT: örneklerimizi function,anonymous ve arrow function ile yapalım.  
//NOT: değişken olarak var,let,const kullanalım.  
//Örnek-1  
//y=3x+4k ==>1.dereceden2bilinmeyenlidenklem algoritması  
//Kullanıcı tarafından alınan x ve k değerlerini hesaplayan algoritma yazınız ?  
//örnek-2  
//Kullanıcı tarafından alınan dereceyi Fahrenheit'a çeviren algoritma yapalım.  
//Formül: (derece*9/5)+32  
//örnek-3 operatör işlemleri: aşağıdaki örneği javascript ile yapalım  
//4+3*2(3:3-1*6+9:1+(3:3))
```

Ödev

```
//ÖDEV
//Örnek-4: Aşağıdaki örnekleri math ile çözelim ?
// -5.9 sayısının aşağıdaki işlemleri yaptıralım ?
// 1-mutlak değeri alsın 5.9
// 2-yuvarlama yapsın 6.0
// 3-karesini alsın 36.00
// 4-karekök alsın 6.0
// 5-yuvarlama yapsın 6.0
// 6-)çıkan sonucu 5 bölün 6/5=1
// 7-) iki sayı arasından karşılaştırma yapsın en küçüğünü alsın ve 1 ve 5
// 8-) küçük sayı eğer tekse 3 eklesin çiftse 5 eklesin
// Örnek-5 : kullanıcı tarafından girilen bir sayıyı negatif mi pozitif mi olduğu ekran yazdıran algoritma ?
//NOT: cast kullalım ?
// Örnek-6 : kullanıcı tarafından password ve repassword alalım sonrasında bu iki değeri karşılaştırma yapalım
//eğer aynı girilirse aynı veri yoksa birbirine uymadı yazan algoritma yapalım
```

Exception handling

```
//Exception handling
document.writeln("<br/>")
let value = () => {
  try {
    document.write("1.alan" )
    alertx("data")

  } catch (error) {
    document.write("Err: " + error)
  }finally{
    document.write("<br/>db.close" )
  }
}
value()
```

Dizi

```
//Dizi
let arr=[44,"str",true];
console.log(typeof arr)
console.log( arr)

document.writeln("<br/>")
let dizi = () => {
  let temp;
  let sum=0;
  let dizi=[];
  for (let index = 0; index < 5; index++) {
    temp=Math.round(Math.random()*9+1);
    dizi[index]=temp;
    sum+=dizi[index];
  }
  //console.log(dizi)
  dizi.forEach(function(value){
    console.log(value)
  });
  document.write("<br/>")
  dizi.forEach((value)=>{
    console.log(value)
  });
  console.log(sum)
}
dizi()
```

filter forEach

```
//filter
document.writeln("<br/>")
let filter = () => {
  let dizi=[1,2,3,4,5,6,7,8];
  let result=dizi.filter((temp)=>{
    return temp%2===1
  });
  result.forEach((temp)=>{
    document.writeln(temp)
  });
}
filter()
```


Object

```
//OBJECT
document.writeln("<br/>")
let object = () => {
  let obj={
    "name":"Hamit",
    "surname":"Mizrak",
    "language":12,
    "isLogin":true,
    "dizi":[1,2,3,4],
    "innerObj":{
      "name":"computer",
    }
  };
  console.log(obj)
  console.log(obj.name.toUpperCase())
  console.log(obj.dizi[3])
}
object()
```

Date

```
▶ document.writeln("<br/>")  
▶ let date = () => {  
    ▶ let value=new Date().getFullYear();  
    ▶ console.log(value)  
    ▶ }  
▶ date()
```




Common Rules

- ▶ Class isimleri Büyük harfle başlar.
- ▶ Class isim veya sıfat kullanılır.
- ▶ paket isimleri hepsi küçük harfle yazılır.

```
public class _01_Javase {  
    public static void main(String[] args) {  
        String $_variableData44="Merhabalar";  
        System.out.println($_variableData44);  
  
        //S1  
        // 1.soru  
        // 1.dereceden 2 bilinmeyenli denklem  
        //  $y=3x+4k$ ;  
        //  $x=5$ ;  
        //  $k=3$   
  
        //S2  
        // 2.soru  
        // Dereceyi Fahrenheit  
        //  $(derece * 9/5) + 32$   
        // 0 derece 32 fahrenheittir
```

```
/*Scanner klavye=new Scanner(System.in);  
System.out.println("Lütfen bir kelime giriniz");  
String data=klavye.nextLine();  
System.err.println(data);*/
```

```
//sayıyı String'e çevirmek  
int number=44;  
String str1=String.valueOf(number); //object  
String str2=Integer.toString(number);
```

```
// String'i sayıya çevirmek  
String str3="44";  
int number2=Integer.valueOf(str3);  
int number3=Integer.parseInt(str3);  
//Escape character \  
System.out.println("1.satır");  
System.out.println("2.satır");  
System.out.println("3.satır\n4.satır");
```

Math

```
///MATH
System.out.println(Math.min(3,-100));
System.out.println(Math.max(3,-100));
System.out.println(Math.pow(2,5));
System.out.println(Math.abs(-100));
System.out.println(Math.sqrt(Math.abs(-100)));
System.out.println(Math.floor(4.9));
System.out.println(Math.ceil(4.1));
System.out.println(Math.round(6.4));
System.out.println(Math.round(6.5));
System.out.println(Math.sin(45));
```


Random

```
//Rastgele  
System.out.println(Math.floor(Math.random()*10+1));  
Random random=new Random();  
int number8=random.nextInt(10)+1;  
System.out.println(number8);
```

Araştırma Ödevleri

- ▶ Compiler nedir ?
- ▶ Interpreter nedir ?
- ▶ Aralarındaki farklar ?
- ▶ primitive type ile wrapper class arasındaki fark ?
- ▶ JVM, JRE, JDK nedir ?
- ▶ Syntax error , Logical Error , runtime error ?

Metotlar-1

```
public class _02_Javase {  
    //Method  
    public static void voidliParametresiz() {  
        System.out.println("voidli Parametresiz");  
    }  
  
    public static void voidliParametrelili(String name) {  
        System.out.println("voidli Parametrelili " + name);  
    }  
  
    public static String voidsizParametresiz() {  
        return "voidsiz Parametresiz";  
    }  
  
    public static String voidsizParametrelili(int number) {  
        return "voidsiz Parametrelili " + number;  
    }  
}
```

Metotlar-2

```
//PSVM
public static void main(String[] args) {
    voidliParametresiz();
    voidliParametreli("data");
    String changeData1 = voidsizParametresiz();
    System.out.println(changeData1);
    String changeData2 = voidsizParametreli(44);
    System.out.println(changeData2);

}

}
```


String

```
public static void main(String[] args) {  
    String value="Java Öğreniyorum Java ";  
    System.out.println(value);  
    System.out.println(value.length());  
    System.out.println(value.trim().length());  
    System.out.println(value.startsWith("J"));  
    System.out.println(value.endsWith(" "));  
    System.out.println(value.toUpperCase());  
    System.out.println(value.toLowerCase());  
  
    System.out.println(value.charAt(0));  
    System.out.println(value.indexOf("Java"));  
    System.out.println(value.lastIndexOf("Java"));  
  
    System.out.println(value.substring(2));  
    System.out.println(value.substring(0,3)); //0<=X<=3-1  
}
```

StringBuilder StringTokenizer

```
public static void stringBuilderTutorials(){
    String data1="javase", data2="javaee", data3="javame";
    System.out.println(data1+data2+data3);

    System.out.println(data1.concat(data2).concat(data3));

    StringBuilder builder=new StringBuilder();
    builder.append(data1).append(data2).append(data3);
    String changeData=builder.toString();
    System.out.println(changeData);
}

public static void stringTokenizerTutorials(){
    String data="javase+~javaee &javame";
    StringTokenizer tokenizer=new StringTokenizer(data,"+~ &");
    while(tokenizer.hasMoreTokens()){
        System.out.println(tokenizer.nextToken());
    }
}

public static void main(String[] args) {
    //stringBuilderTutorials();
    stringTokenizerTutorials();
}
```


if elseif elseif else

```
public static void conditional3() {  
    int number = 5;  
    if (number == 1) {  
        System.out.println("1");  
    } else if (number == 2)  
        System.out.println("2");  
    else if (number == 3)  
        System.out.println("3");  
    else if (number == 4)  
        System.out.println("4");  
    else if (number == 5)  
        System.out.println("5");  
    else  
        System.out.println("farklı bir sayıdır");  
}
```

Swich-case

```
public static void conditional4() {  
    int number = 5;  
    switch (number) {  
        case 1:  
            System.out.println("1");  
            break;  
        case 2:  
            System.out.println("2");  
            break;  
        case 3:  
            System.out.println("3");  
            break;  
        case 4:  
            System.out.println("4");  
            break;  
        case 5:  
            System.out.println("5");  
            break;  
        default:  
            System.out.println("farklı bir sayıdır");  
            break;  
    }  
}
```


Loop

```
public class _06_javase {  
    //Loop  
    public static void loopFor() {  
        for (int i = 1; i <= 10; i++) {  
            System.out.print(i + " ");  
        }  
    }  
  
    public static void loopWhile() {  
        int i = 1;  
        while (i <= 10) {  
            System.out.print(i + " ");  
            i++;  
        }  
    }  
  
    public static void loopDoWhile() {  
        int i = 1;  
        do {  
            System.out.print(i + " ");  
            i++;  
        } while (i <= 10);  
    }  
}
```

Array

```
//Array:eleman sayısı belli olanlarda
public static void arrayTutorials() {
    int[] arr = new int[4];
    arr[0] = 5;
    arr[1] = 3;
    arr[2] = 4;
    System.out.println(arr[0]);

    for (int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
    }
    System.out.println("");
    Arrays.sort(arr);

    for (int temp : arr) {
        System.out.print(temp + " ");
    }
}
```

Collection

```
//Collection:eleman sayısı belli olmayanlar
public static void collectionTutorials1() {
    //Liste => LAV (LinkedList, ArrayList, Vector)
    List<Integer> list = new ArrayList<>();
    list.add(10);
    list.add(10);
    list.add(30);
    list.add(20);
    list.add(40);

    for (Integer temp : list) {
        System.out.println(temp + " ");
    }
    System.out.println("");

    /* for (int i = 0; i < list.size() ; i++) {
        System.out.println(list.get(i));
    }
    System.out.println("");

    for( Integer temp: list){
        System.out.println(temp+" ");
    }
    System.out.println("");

    list.stream().forEach(System.out::println);
    System.out.println("");

    Collections.sort(list);
    list.stream().forEach((temp)->{
        System.out.println(temp);
    });*/
}
```

```
public static void collectionTutorials2() {  
    //Küme => HalaT  
    //Liste => LAV (linkedList, ArrayList, Vector)  
    Set<Integer> integerSet = new LinkedHashSet<>();  
    integerSet.add(10);  
    integerSet.add(10);  
    integerSet.add(30);  
    integerSet.add(20);  
    integerSet.add(40);  
  
    for (Integer temp : integerSet) {  
        System.out.println(temp + " ");  
    }  
}
```



```
public static void collectionTutorials3() {  
    //Map  
    Map<String, Object> mapList = new TreeMap<>();  
    mapList.put(UUID.randomUUID().toString(), "Malatya");  
    mapList.put(UUID.randomUUID().toString(), "Elazığ");  
  
    for (String temp : mapList.keySet()) {  
        System.out.println(temp + " => " + mapList.get(temp));  
    }  
}
```

```
public static void collectionTutorials4() {  
    //1-9 arasında rastgele 5 tane sayı oluşturulsun bu sayıların toplamını veren algoritma  
    int[] arr = new int[5];  
    Random random = new Random();  
    int sum=0;  
    //Çift olanlardan kaçane ?  
    //çift olanların toplamı ?  
    //tek olanlardan kaçane ?  
    //tek olanların toplamı ?  
  
    for (int i = 0; i < arr.length; i++) {  
        int temp = random.nextInt(9)+1;  
        arr[i]=temp;  
        sum+=arr[i];  
    }  
  
    for (int temp: arr) {  
        System.out.print(temp+" ");  
    }  
    System.out.println("TOPLAM => "+sum);  
}
```

```
public static void main(String[] args) {  
    //loopFor();  
    //System.out.println();  
    //loopWhile();  
    //System.out.println();  
    //loopDoWhile();  
    //exceptionHandling();  
    //arrayTutorials();  
    /*collectionTutorials1();  
    System.out.println();  
    collectionTutorials2();*/  
    //collectionTutorials3();  
  
    collectionTutorials4();  
}  
}
```


OOP Inheritance

```
package com.hamitmizrak.oop.inheritancex;
```

```
import lombok.AllArgsConstructor;  
import lombok.Builder;  
import lombok.Data;  
import lombok.NoArgsConstructor;
```

```
//LOMBOK  
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
@Builder  
public class Person {  
    private long id;  
    private String username;  
    private String surname;  
  
    public void deneme(String name){  
        System.out.println(name);  
    }  
}
```

```
package com.hamitmizrak.oop.inheritancex;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
public class Student extends Person{
    //global variable
    private String unicodeCharacter;

    public Student(long id, String username, String surname, String unicodeCharacter) {
        super(id, username, surname);
        this.unicodeCharacter = unicodeCharacter;
    }

    @Override
    public void deneme(String name) {
        super.deneme(name);
    }

    @Override
    public String toString() {
        return "Student{" +
            "unicodeCharacter=" + unicodeCharacter + "\" +
            " " + super.toString();
    }
}
```

```
package com.hamitmizrak.oop.inheritancex;
```

```
public class Teacher extends Person{  
    @Override  
    public void deneme(String name) {  
        super.deneme(name);  
    }  
}
```


OOP abstract

```
package com.hamitmizrak.oop.abstractx;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

//LOMBOK
@Data
@NoArgsConstructor
@AllArgsConstructor
abstract public class Person {
    private long id;
    private String username;
    private String surname;

    public void deneme(String name){
        System.out.println(name);
    }

    abstract public void govdesizMethod(int number);
}
```

```
package com.hamitmizrak.oop.abstractx;

public class Student extends Person {
    @Override
    public void deneme(String name) {
        super.deneme(name);
    }

    @Override
    public void govdesizMethod(int number) {
        System.out.println("Student: "+number);
    }
}
```

```
package com.hamitmizrak.oop.abstractx;

public class Teacher extends Person {
    @Override
    public void deneme(String name) {
        super.deneme(name);
    }

    @Override
    public void govdesizMethod(int number) {
        System.out.println("Teacher: "+number);
    }
}
```


Interface

```
package com.hamitmizrak.oop.interfaces;
```

```
public interface ICommon {  
    public void create();  
    public void delete();  
    public void select();  
    public void update();  
}
```

```
package com.hamitmizrak.oop.interfacex;
```

```
public class Deneme implements ICommon{
```

```
    @Override  
    public void create() {
```

```
    }
```

```
    @Override  
    public void delete() {
```

```
    }
```

```
    @Override  
    public void select() {
```

```
    }
```

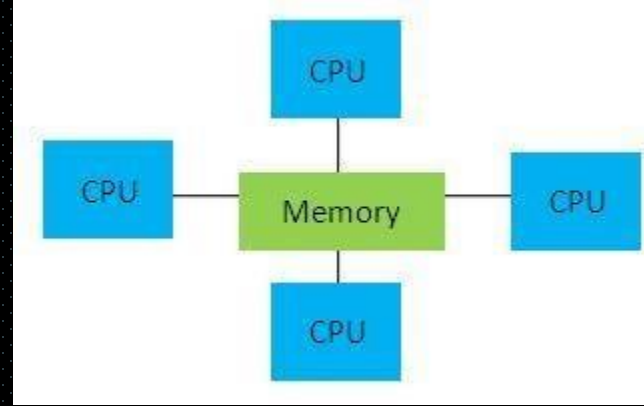
```
    @Override  
    public void update() {
```

```
    }
```

```
}
```


Paralel Programlama Nedir ?

- İşlemcinin aynı anda birden fazla işlem yapabilme yeteneğine denir.
- Örnek: Mikrofondaki ses duymak, ekranı görmek, etc



Paralel Programlama Avantajları ?

- ▶ Hızlılığı
- ▶ Sistem kaynakları ölçeklenebilirliğini sağlar.
- ▶ eş zamanlı çalışma imkanını sağlamak
- ▶ $X100 = 100$ bilgisayar

Paralel Programlama Dezavantajları ?

- ▶ Öğrenmesi zor ,
- ▶ İnce düşünmek ve programlamak.
- ▶ Bilgisayar,server etc ısınma → Soğutucuyla

Seri Hesaplama

- ▶ 100 tane iş merkezim → $1 + 2 + 3 + 4 + 5 + 6 + 7 \dots 100$
- ▶ Tek bir işlemci :

Paralel Hesaplama

- ▶ 100 tane iş merkezim → $1 + 2 + 3 + 4 + 5 + 6 + 7 \dots 100$
- ▶ Çok işlemci

7 (method)+2 (çağırma)

```
// Thread
// senkron: aynı anda sadece 1 iş yapabilme yeteneği (Single Thread)

// asenkron: aynı anda birden fazla iş yapabiliriz. (Multi Thread)

// Java Aynı anda birden fazla processle çalışmamıza olanak sağlar.
// thread kullanıyorsan exception handling kullanmalısın (try-catch)

// Multi Thread:
// Hızlılığı sağlar.
// Güvenilirliği sağlar

// run()    ==> thread çalıştırır.
// start()   ==> thread başlaması
// join()    ==> üstteki thread bitmeden devam etmez.
// wait()    ==> Threadi 2. emre kadar durmasını isteriz.
// notify()  ==> ilgili threadi uyanması istiyoruz.
// notifyAll() ==> Bütün thread'leri ayağa kaldırır.

// priority() ==> Threadlere Öncelik vermek
// destroy()  ==> Thread Ölmesini sağlamak
// getName()  ==> Thread adını bize verir.
// setName()  ==> Thread adını değiştirmek
// isAlive()  ==> ilgili Thread yaşıyor mu yaşamayıp mu bilgisini verir.
// sleep()    ==> belirli milisaniyede uyunun tekrar geri kalksın (sleep-wakeup)
```

join kısaca Threadler bitmeden yeni işleme geçme

- ▶ `thread.start();`
- ▶ `thread2.start();`
- ▶ `thread.join();`
- ▶ `thread2.join();`
- ▶ `sout("buradaki yazı gelmesi için önce yukarıdaki thread ve thread2 bitmeli")`

wait (ikinci emre kadar bekle)

NOT: sleep kendiliğinde uyuyup kalkıyor

► wait =dur

notify

notifyAll

- notify = tek kişi kalsın
- notifyAll = bütün koğuş kalsın

Priority Thread

- ▶ Thread thread1=new Thread(new ThreadClass());
- ▶ Thread thread2=new Thread(new ThreadClass());
- ▶ Thread thread3=new Thread(new ThreadClass());
- ▶
- ▶ thread1.setPriority(Thread.MIN_PRIORITY); //file işlemlerinde
- ▶ thread3.setPriority(Thread.MAX_PRIORITY);
- ▶ thread2.setPriority(Thread.NORM_PRIORITY);
- ▶
- ▶ thread1.start();
- ▶ thread2.start();
- ▶ thread3.start();

Thread ile runnable arası fark

- ▶ İkisinde de Thread oluşturabiliyoruz.
- ▶ extends Thread implements Runnable (farkı)
- ▶ Public void run(){} (ikisinde de var)

thr.start(); //thread thr.run(); //Runnable

EĞER Thread oluşturacaksak önceliğimiz

```
Public class Thread55 extends Thread{Public void run(){} }
```

```
Psvm --> Thread55 thr=new Thread55(); thr.start(); //thread
```

```
Public class Thread66 implements Runnable{Public void run(){} }
```

```
Psvm --> Thread55 thr=new Thread55(); thr.start(); //thread
```

```
Psvm --> Thread66 thr=new Thread66(); thr.run(); //Runnable
```

Ne zaman extends ne zaman runnable

- ▶ //Bu class başka bir Classı extends etmeye çalışırsa ne yapmamız gerekirdi.
- ▶ Tabikide Runnable yapmak zorunda kalıyoruz.
- ▶
- ▶ `public class Ders_A01 extends Thread {`
- ▶ `@Override`
- ▶ `public void run(){`
- ▶
- ▶ `System.out.println("Threading çalışıyor.");`
- ▶ `}`

MultiThread (çağırma yöntemleri) (2)

- ▶ extends (Class ve metot) → kalıtım yoluyla çağırıyoruz.
- ▶ (extends Thread) → kalıtım
- ▶ implements → interface yoluyla çağırıyoruz.
- ▶ (implements Runnable) → interface
- ▶ anonymous → metot yoluyla
- ▶ Thread thread=new Thread(new Runnable(){});

Thread Kullanım Yerleri

```
// 1-) extends (Kalıtım) ==> Class
// public class MultiThreadExam extends Thread{}
// extends ==> javada 1 tane extends kullanabilirsin
// extends zorunlu olarak run metodunu eklemeyiz
public class _34_Thread2 extends Thread {
    @Override
    public void run() {
    }
}

// 2-) implements (Interface) ==> interface
// public class MultiThreadExam implements Runnable{}
// implements ==> Javada istediğimiz kadar implements virgül koyarak
// yazabiliriz.
// implement zorunlu olarak run metodunu zorunlu ekler
class _34_Thread2Implements implements Runnable {
    @Override
    public void run() {
    }
}

// 3-) anonymous (isimsiz) ==> Metot olarak
// Thread thread= new Thread(new Runnable(){});
class _34_Thread2Method{
    public void threadMetodu() {
        Thread thread=new Thread(new Runnable() {
            @Override
            public void run() {

            }
        });
    }
}
```

Kalıtım Thread

```
package com.hamitmizrak.javase;

// Thread
// kalıtım yoluyla

import lombok.Getter;
import lombok.Setter;

@Getter @Setter
public class _19_Threading extends Thread{
    private String threadName;

    public _19_Threading(String threadName) {
        this.threadName = threadName;
    }

    @Override
    public void run() {
        for (int i = 1; i <=10 ; i++) {
            try {
                //1000ms=1sn
                Thread.sleep(1000);
                if(i==1){
                    System.out.println("\n### Başlangıç Thread###");
                    System.out.println(this.threadName+" ==> "+i+" ");
                }else if(i==10){
                    System.out.println("\n### Bitiş Thread###");
                    System.out.println(this.threadName+" ==> "+i+" ");
                }else{
                    System.out.println(this.threadName+" ==> "+i+" ");
                }
            } catch (InterruptedException e) {
                throw new RuntimeException(e);
            }
            System.out.println(threadName+" ==> "+i+" ");
        }
    }
}
```

```

public static void main(String[] args) throws InterruptedException {
    _19_Threading thread1=new _19_Threading("1.thread ==>");
    _19_Threading thread2=new _19_Threading("2.thread ==>");
    _19_Threading thread3=new _19_Threading("3.thread ==>");

    //Thread Genel Bilgileri
    System.out.println("Thread1: isAlive "+thread1.isAlive());
    thread1.start();
    System.out.println("Thread1: ID "+thread1.getId());
    System.out.println("Thread1: hashCode "+thread1.hashCode());
    System.out.println("Thread1: isAlive "+thread1.isAlive());
    System.out.println("Thread1: getName "+thread1.getName());
    thread1.setName("ThreadMalatya");
    System.out.println("Thread1: getName "+thread1.getName());

    //thread1.wait();//wait =>thread: durur
    //thread1.notify();//notify =>thread: çalıştırır.
    //thread1.notifyAll();//join: öncelikle 1 ve2 bitsin

    thread2.start();
    thread1.join();
    thread2.join();
    thread3.start();
}

```


Interface Thread → 1

```
package com.hamitmizrak.javase;

import lombok.Getter;
import lombok.Setter;

// arayüz (interface)
@Getter
@Setter
public class _19_Threading2 implements Runnable {

    private String threadName;

    public _19_Threading2(String threadName) {
        this.threadName = threadName;
    }

    @Override
    public void run() {
        for (int i = 1; i <= 10; i++) {
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                throw new RuntimeException(e);
            }
            System.out.println(threadName+" ==> "+i+" ");
        }
    }
}
```

Interface Thread → 2

```
public static void main(String[] args) throws InterruptedException {  
    Thread thread1 = new Thread(new _19_Threading2("1.thread ==>"));  
    Thread thread2 = new Thread(new _19_Threading2("2.thread ==>"));  
    Thread thread3 = new Thread(new _19_Threading2("3.thread ==>"));  
  
    thread1.start();  
    thread2.start();  
    //Join  
    thread1.join();  
    thread2.join();  
  
    thread3.start();  
}  
}
```


ThreadBase

```
package com.hamitmizrak.javase_tutorials.paralel;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.extern.log4j.Log4j2;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
@Log4j2
public class ThreadBase {

    private String threadName;
}
```

ImplementsRunnable → 1

// Eğer extends hakkını başka class kullanmışsa biz o zaman implements kullacağız
`public class _03_ImplementsRunnable extends ThreadBase implements Runnable{`

```
//constructor
public _03_ImplementsRunnable(String threadName) {
    super(threadName);
}

//RUN
@Override
public void run() {
    for (int i = 1; i <= 9 ; i++) {
        try {
            Thread.sleep(1000);
            if(i==1){
                System.out.println("### Başlangıç ###");
                System.out.println(getThreadName()+" ==> "+i+" ");
            }else if(i==9){
                System.out.println("\n&&& Bitiş &&&");
                System.out.println(getThreadName()+" ==> "+i+" ");
            }else{
                System.out.println(getThreadName()+" ==> "+i+" ");
            }
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        } catch (Exception e){
            e.printStackTrace();
        }
    }
}
```

ImplementsRunnable → 2

```
//PSVM
public static void main(String[] args) throws InterruptedException {
    Thread thread1=new Thread(new _03_ImplementsRunnable("1.Thread"));
    Thread thread2=new Thread(new _03_ImplementsRunnable("2.Thread"));
    Thread thread3=new Thread(new _03_ImplementsRunnable("3.Thread"));

    //1.Thread bitsin
    thread1.start();
    thread1.join();

    //2.Thread devam etsin
    thread2.start();
    thread2.join();

    //3.Thread devam etsin
    thread3.start();
}
}
```

Anonymous Thread → 1

```
package com.hamitmizrak.javase;

public class _19_Threading3 {

    public static void main(String[] args) {
        Runnable runnable=new Runnable() {
            @Override
            public void run() {
                for (int i = 1; i <= 10; i++) {
                    try {
                        Thread.sleep(1000);
                    } catch (InterruptedException e) {
                        throw new RuntimeException(e);
                    }
                    System.out.println(i+" ");
                }
            }
        };
        runnable.run();
    }
}
```

synchronized → 1

```
package com.hamitmizrak.javase_tutorials.paralel;

import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.extern.log4j.Log4j2;

// LOMBOK
@Data
@NoArgsConstructor
@Log4j2
public class _06_Synchronized extends Thread{

    // Global Variable
    private String threadName;
    private static int counter=0;

    // parametreli constructor
    public _06_Synchronized(String threadName) {
        this.threadName = threadName;
    }
}
```


synchronized → 2

```
// Counter Method => synchronized
// synchronized: ilk giren Thread işini bitirmeden diğer Thread izin vermiyor.
public synchronized void counterPlus(){
    //counter=counter+1;
    //counter+=1;
    counter++;
    System.out.println(counter);
    log.info(counter);
}
```

synchronized → 3

```
// Thread Run
@Override
public void run() {
    for (int i = 1; i <= 9; i++) {
        System.out.println(threadName+" START");
        try {
            Thread.sleep(1000);
            // synchronized
            counterPlus();
        } catch (InterruptedException e) {
            System.out.println("InterruptedException: "+e);
            throw new RuntimeException(e);
        }
    } // end for
    System.out.println(threadName+" END");
} //end run
```

synchronized → 4

```
// PSVM
// Concurrency: Eş zamanlı çalışma demektir.
// Senkron: Aynı sadece 1 iş yapabilme yeteneğidir.
// Asenkron: Aynı sadece 1'den fazla iş yapabilme yeteneğidir.
// MultiThreading: Aynı anda birden fazla iş yapabilme yeteneği
public static void main(String[] args) throws InterruptedException {
    _06_Synchronized thread1=new _06_Synchronized("Threading-1");
    _06_Synchronized thread2=new _06_Synchronized("Threading-2");

    // START TIME
    long startTime= System.currentTimeMillis();

    thread1.start();
    thread2.start();
    thread1.join();
    thread2.join();

    //9(Döngü adet sayısı) * 2(thread) = 18(counter)
    System.out.println("Main Threading Counter "+counter);
    log.info("Main Threading Counter "+counter);

    // END TIME
    long endTime=System.currentTimeMillis();
    System.out.println("Toplam Geçen Süre: "+ Math.round(endTime-startTime)/1000);
}
}
```

ThreadLock → 1

```
package com.hamitmizrak.javase_tutorials.paralel;

import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.extern.log4j.Log4j2;

// LOMBOK
@Data
@NoArgsConstructor
@Log4j2
public class _07_ThreadLock extends Thread{

    // Global Variable
    private String threadName;
    private static int counter=0;

    // Thread Lock: Thread Kitlemesi
    private Object threadLockObject=new Object();

    // Parametreli Constructor
    public _07_ThreadLock(String threadName) {
        this.threadName = threadName;
    }
}
```

ThreadLock → 2

```
// Method ==> synchronized
// Counter Method => synchronized
// synchronized: ilk giren Thread işini bitirmeden diğer Thread izin vermiyor.
public synchronized void counterPlus(){
    //counter=counter+1;
    //counter+=1;
    counter++;
    System.out.println(counter);
    log.info(counter);
} // end counterPlus
```

ThreadLock → 3

```
// Thread Run
@Override
public void run() {
    // Özel ==> synchronized
    // synchronized (threadLockObject){}
    // sadece 1 Thread'in girmesine izin verdim.
    synchronized (threadLockObject){
        for (int i = 1; i <=9; i++) {
            System.out.println(threadName+" START");
            try {
                Thread.sleep(1000);
                // synchronized
                counterPlus();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException: "+e);
                throw new RuntimeException(e);
            }
        } // end for
    }
    System.out.println(threadName+" END");
} //end run
```

ThreadLock → 4

```
// PSVM
// Concurrency: Eş zamanlı çalışma demektir.
// Senkron: Aynı sadece 1 iş yapabilme yeteneğidir.
// Asenkron: Aynı sadece 1'den fazla iş yapabilme yeteneğidir.
// Multithreading: Aynı anda birden fazla iş yapabilme yeteneği
public static void main(String[] args) throws InterruptedException {
    _07_ThreadLock thread1=new _07_ThreadLock("Threading-1");
    _07_ThreadLock thread2=new _07_ThreadLock("Threading-2");

    // START TIME
    long startTime= System.currentTimeMillis();

    // THREADING
    thread1.start();
    thread2.start();

    thread1.join();
    thread2.join();

    //9(Döngü adet sayısı) * 2(thread) = 18(counter)
    System.out.println("Main Threading Counter "+counter);
    log.info("Main Threading Counter "+counter);

    // END TIME
    long endTime=System.currentTimeMillis();
    System.out.println("Toplam Geçen Süre: "+ Math.round(endTime-startTime)/1000);
}
}
```

Thread Pool Executor Services → 1

```
package com.hamitmizrak.javase_tutorials.paralel;

import lombok.SneakyThrows;
import lombok.extern.log4j.Log4j2;
import java.util.Scanner;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.TimeUnit;

@Log4j2
// @SneakyThrows => Throws
public class _08_ThreadPoolExecutorServices implements Runnable {

    private long id;
    private String name;
    private String surname;

    //parametreli constructor
    public _08_ThreadPoolExecutorServices(long id, String name, String surname) {
        this.id = id;
        this.name = name;
        this.surname = surname;
    }

    // FULL NAME
    public void fullName() {
        System.out.println("Threading: " + this.name + this.surname);
    }
}
```


Thread Pool Executor Services → 2

```
// RUN
@Override
public void run() {
    System.out.println(" START");
    fullName();
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        throw new RuntimeException(e);
    }
    System.out.println(" END");
}
```

Thread Pool Executor Services → 3

```
// Manuel Thread process
@SneakyThrows
public static void manuelThread() {
    // Manuel Thread process
    Thread thread1 = new Thread(new _08_ThreadPoolExecutorServices(1, "Adı1", "Soyadı1"));
    Thread thread2 = new Thread(new _08_ThreadPoolExecutorServices(2, "Adı2", "Soyadı2"));
    Thread thread3 = new Thread(new _08_ThreadPoolExecutorServices(3, "Adı3", "Soyadı3"));
    Thread thread4 = new Thread(new _08_ThreadPoolExecutorServices(4, "Adı4", "Soyadı4"));
    Thread thread5 = new Thread(new _08_ThreadPoolExecutorServices(5, "Adı5", "Soyadı5"));

    // Thread
    thread1.start();
    thread2.start();
    thread1.join();
    thread2.join(); //1 ve 2 Thread bitmeden diğer Threadler başlamasın

    thread3.start();
    thread4.start();
    thread3.join();
    thread4.join(); //3 ve 4 Thread bitmeden diğer Threadler başlamasın

    thread5.start();
    thread5.join();
}
```

Thread Pool Executor Services ➔ 4

```
// Dynamcis Thread Method
@SneakyThrows
public static void dynamicsThread() {
    // Dynamics Thread process
    // Thread Pool: Threadlerimizin bir arada olduğu yerdir.
    // Thread Pool: 2 tane Threadin aynı anda çalışmanı istiyoruz.
    ExecutorService executorService = Executors.newFixedThreadPool(2);
    for (int i = 1; i <= 5; i++) {
        executorService.submit(new _08_ThreadPoolExecutorServices(i, "Adı" + i, "Soyadı" + i));
    }

    // Yukarıda Dinamik olarak yazdığımız Thread bitmeden yeni thread izin vermiyorum.
    // Eğer shutdown() yazmazsam ==> 5 Thread aynı anda çalışmaya başlar.
    // Eğer shutdown() yazarsam ==> Thread en kötü durumda 1 dakika sonra kapanacaktır.
    executorService.shutdown();

    System.out.println("Thread Continues Data");

    // Thread bitmesini zorlamak için yazdım
    executorService.awaitTermination(1, TimeUnit.MINUTES);
    System.out.println("Thread Bitti");
}
```

Thread Pool Executor Services → 5

```
// All Thread Process
public static void allThreadProcess(){
    while(true){
        Scanner klavye = new Scanner(System.in);
        System.out.println("\nLütfen seçim yapınız\n1-)Manuel Thread\n2-)Dinamik Thread\n3-)Çıkış");
        int key = klavye.nextInt();
        switch (key) {
            case 1:
                // START TIME
                long startTime1= System.currentTimeMillis();
                manuelThread();
                // END TIME
                long endTime1=System.currentTimeMillis();
                System.out.println("Toplam Geçen Süre: "+ Math.round(endTime1-startTime1)/1000);
                break;

            case 2:
                // START TIME
                long startTime2= System.currentTimeMillis();
                dynamicsThread();
                // END TIME
                long endTime2=System.currentTimeMillis();
                System.out.println("Toplam Geçen Süre: "+ Math.round(endTime2-startTime2)/1000);
                break;

            case 3:
                System.out.println("Çıkış yapılıyor");
                System.exit(0);
                break;
        }
    }
}
```

Thread Pool Executer Services ➔ 6

```
// PSVM
public static void main(String[] args) {
    allThreadProcess();
} // end PSVM
} // end _08_ThreadPoolExecuterServices
```

await() notify() notifyAll() ➔ 1

```
package com.hamitmizrak.javase_tutorials.paralel;

import lombok.SneakyThrows;
import java.util.Scanner;

// await: ikinci bir emre kadar bekle
// notify: artık emir geldi kalkabilirsin
// notifyAll(): Bütün Threadleri kaldırır.
public class _09_Notify_Await {
    // Global Variable
    private Scanner klavye = new Scanner(System.in);
    private Object lockData = new Object(); // Object
```

await() notify() notifyAll() → 2

```
// await Method
@SneakyThrows // Lombok
public void awaitMethod() throws InterruptedException {
    //synchronized (this){}
    synchronized (lockData) {
        System.out.println("1.Thread çalıştı ancak bundan sonraki 2.Thread çalışması için 1.Thread çalışması gerekiyor.");
        lockData.wait();
    }
    System.out.println("Thread Uykudan Kalktı");
}
```

await() notify() notifyAll() → 3

```
// notify Method
@SneakyThrows // Lombok
public void notifyMethod() {
    Thread.sleep(2000);
    synchronized (lockData) {
        System.out.println("2.Thread Devam etmesi için 1.Thread devam etmelidir bunun için \nherhangi bir tuşa basınız");
        klavye.nextLine();
        lockData.notify();
        System.out.println("1.Thread uyandı ve 2.Thread 5 saniye uyanacak ve sonra devam edecek");
    }
    for (int i = 1; i <= 5; i++) {
        System.out.println(i + " saniye");
        Thread.sleep(1000);
    }
}
```


await() notify() notifyAll() ➔ 4

```
// All Thread Process
@SneakyThrows
public static void allThreadProcess() {
    _09_Notify_Await notify_await = new _09_Notify_Await();

    // Anonymous Thread-1 ( Yatıyor)
    Runnable thread1=() -> {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Thread-X1");
        }
        try {
            notify_await.awaitMethod();
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
    };
    //Thread oluştur ==> interface Anonymous Lambda Expression
    Thread lambdaThread=new Thread(thread1);
```

await() notify() notifyAll() → 5

```
// Anonymous Thread-2 (notify: uyandırıyor)
Thread thread2 = new Thread(new Runnable() {
    @Override
    public void run() {
        notify_await.notifyMethod();
        for (int i = 1; i <= 5; i++) {
            System.out.println("Thread-X2");
        }
    }
}); // end thread2

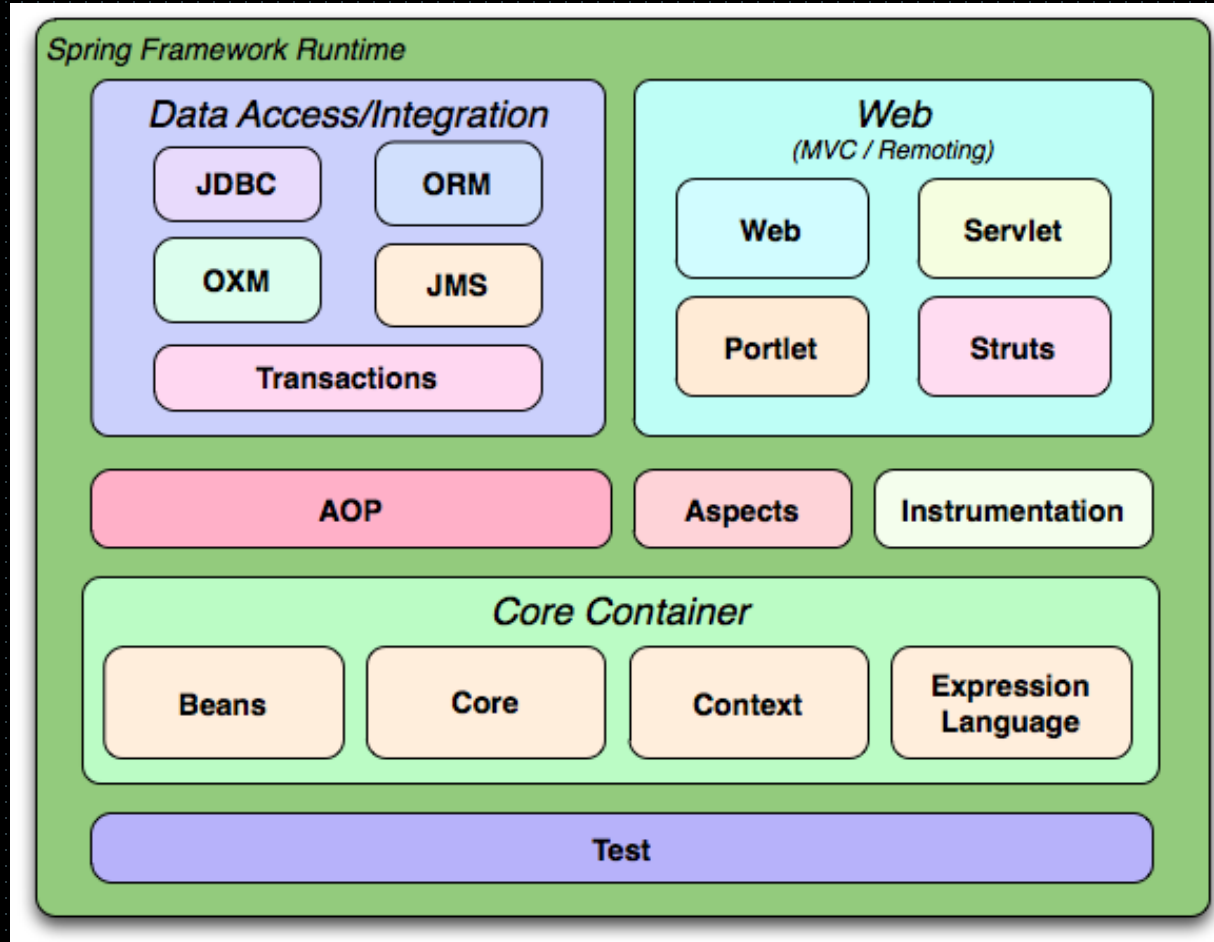
lambdaThread.start();
thread2.start();
lambdaThread.join();
thread2.join();

} // end allThreadProcess
```

PSVM

```
// PSVM
public static void main(String[] args) {
    allThreadProcess();
} // end PSVM
} // end _09_Notify_Await
```

<https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/overview.html>



pom.xml

```
<!--Spring Web Mvc Rest-->  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-web</artifactId>  
</dependency>
```

```
package com.hamitmizrak.springboot._1_SpringMVC;

import com.hamitmizrak.springboot.dto.RegisterDto;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

//Spring MVC
@Controller
@RequestMapping("/mvc/v1")
public class ControllerThymeleaf {
```

```
// http://localhost:4444/mvc/v1/thmeleaf0
@GetMapping("/thymeleaf0")
@ResponseBody
public String thymeleaf0(){
    return "Merhabalar Thmeleaf 0";
}

// http://localhost:4444/mvc/v1/thymeleaf1
@GetMapping("/thymeleaf1")
public String thymeleaf1(){
    return "thymeleaf_frontend1";
}

// http://localhost:4444/mvc/v1/thymeleaf2
@GetMapping("/thymeleaf2")
public String thymeleaf2(Model model){
    model.addAttribute("key2","Merhabalar Ben Javadan geldim");
    return "thymeleaf_frontend2";
}
```



```
// http://localhost:4444/mvc/v1/thymeleaf3
@GetMapping("/thymeleaf3")
public String thymeleaf3(Model model){
    RegisterDto registerDto= RegisterDto.builder()
        .id(1L)
        .name("Hamit")
        .surname("Mizrak")
        .email("hamitmizrak@gmail.com")
        .password("Hm123456@.")
        .build();
    model.addAttribute("key3",registerDto);
    return "thymeleaf_frontend3";
}
```

```

// http://localhost:4444/mvc/v1/thymeleaf3
@GetMapping("/thymeleaf4")
public String thymeleaf4(Model model){
    List<RegisterDto> listem=new ArrayList<>();

    for (int i = 1; i <=10 ; i++) {
        RegisterDto registerDto= RegisterDto.builder()
            .id(Long.valueOf(i))
            .name("Hamit "+i)
            .surname("Mizrak "+i)
            .email("hamitmizrak@gmail.com "+i)
            .password("Hm123456@. "+i).
            date(new Date(System.currentTimeMillis()))
            .build();
        listem.add(registerDto);
    }

    model.addAttribute("key4",listem);
    return "thymeleaf_frontend4";
}
}

```

pom.xml

```
<!-- Spring data-jpa-->  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-jpa</artifactId>  
</dependency>
```


application.properties → 1

```
#####  
### Database #####  
## H2Database  
logging.level.org.springframework.jdbc.core = TRACE  
spring.jpa.properties.hibernate.format_sql = true  
spring.jpa.properties.hibernate.show_sql = true  
spring.jpa.show-sql=true  
  
# once create sonra durdur update yap  
#jpa > none update create, create-drop, validate  
spring.jpa.hibernate.ddl-auto = update  
  
## Generic JNDI Data Source data 44  
##in Memory  
##spring.datasource.url=jdbc:h2:mem:denemedb  
  
## Directory on Disk  
## spring.datasource.url=jdbc:h2:file:C:/data/denemedb  
  
##Current User Directory  
##spring.datasource.url=jdbc:h2:file:~/denemedb
```

application.properties → 2

```
## Current Working Directory
## http://localhost:4444/h2-console
## following setting h2-console
spring.h2.console.settings.web-allow-others=true
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.url=jdbc:h2:file:./memory_persist/blog
##spring.datasource.url=jdbc:h2:file:./memory_persist/register;useSSL=falseDB_CLOSE_ON_EXIT=FALSE;AUTO_SERVER=TRUE
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
```

application.properties → 3

```
## Mysql
##CREATE SCHEMA `employee_database` DEFAULT CHARACTER SET utf8 COLLATE utf8_turkish_ci ;root
## use employee_database;
#spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
#spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
#spring.datasource.url=jdbc:mysql://localhost:3306/register?createDatabaseIfNotExist=true&autoReconnect=true&useSSL=false
#spring.datasource.username=root
#spring.datasource.password=root

## Mongo DB
## Robo 3T
## @SpringBootApplication içinde bunu yaz ==> @EnableMongoRepositories
## @Repository ==> public interface IDeneme extends MongoRepository<EntityName,Long>{}
#spring.data.mongodb.database=mongodb
#spring.data.mongodb.uri=mongodb://localhost:27017

## Postgresql
#spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation=true
#spring.datasource.url=jdbc:postgresql://localhost:5432/employee_database
#spring.datasource.username=postgres
#spring.datasource.password=root
#spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.PostgreSQLDialect

# Oracle
#spring.datasource.url=jdbc:oracle:thin:@localhost:1521:orcl
#spring.datasource.username=system
#spring.datasource.password=Password123
```


pom.xml

```
<!--Spring Web Mvc Rest-->  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-web</artifactId>  
</dependency>
```

API

- ▶ API: Application Programming Interface
- ▶ Farklı teknolojilerin, farklı cihazların birbiriyle haberleşmesini sağlar.

```
package com.hamitmizrak.controller.api;

import com.hamitmizrak.springboot.dto.RegisterDto;
import lombok.extern.log4j.Log4j2;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.Date;

@RestController
@RequestMapping("/api/v1")
@CrossOrigin
@Log4j2
public class LoginApiImpl {
```

```
// http://localhost:4444/api/v1/rest1
@GetMapping("/rest1")
public RegisterDto getRest1(){
    return RegisterDto.builder()
        .id(1L)
        .name("Hamit")
        .surname("Mizrak")
        .email("hamitmizrak@gmail.com")
        .password("Hm123456@.")
        .build();
}
```

```
// http://localhost:4444/api/v1/rest2/1
@GetMapping("/rest2/{id}")
public RegisterDto getRest2(@PathVariable(name="id") Long id){
    return RegisterDto.builder()
        .id(id)
        .name("Hamit")
        .surname("Mizrak")
        .email("hamitmizrak@gmail.com")
        .password("Hm123456@.")
        .build();
}
```

```
// http://localhost:4444/api/v1/rest3
// http://localhost:4444/api/v1/rest3/1
@GetMapping("/{rest3}/{rest3/{id}}")
public RegisterDto getRest3(@PathVariable(name="id",required = false) Long id){
    return RegisterDto.builder()
        .id(id)
        .name("Hamit")
        .surname("Mizrak")
        .email("hamitmizrak@gmail.com")
        .password("Hm123456@.")
        .date(new Date(System.currentTimeMillis()))
        .build();
}
```

```

// http://localhost:4444/api/v1/rest4
// http://localhost:4444/api/v1/rest4/0
// http://localhost:4444/api/v1/rest4/1
@GetMapping("/{rest4}/{id}")
public ResponseEntity<RegisterDto> getRest4(@PathVariable(name="id",required = false) Long id){
    if(id==null){
        log.error("404 Not Found hatası");
        return ResponseEntity.notFound().build();
    } else if(id==0){
        log.error("400 Bad Request hatası");
        return ResponseEntity.badRequest().build();
    }
    RegisterDto registerDto=RegisterDto.builder()
        .id(id)
        .name("Hamit")
        .surname("Mizrak")
        .email("hamitmizrak@gmail.com")
        .password("Hm123456@.")
        .date(new Date(System.currentTimeMillis()))
        .build();
    log.info(registerDto);
    return ResponseEntity.ok(registerDto);
}

```



```

// http://localhost:4444/api/v1/rest5
// http://localhost:4444/api/v1/rest5/0
// http://localhost:4444/api/v1/rest5/1
@GetMapping("/{rest5}/{id}")
public ResponseEntity<?> getRest5(@PathVariable(name="id",required = false) Long id){
    if(id==null){
        log.error("404 Not Found hatası");
        return ResponseEntity.notFound().build();
    } else if(id==0){
        log.error("400 Bad Request hatası");
        return ResponseEntity.badRequest().build();
    }
    RegisterDto registerDto=RegisterDto.builder()
        .id(id)
        .name("Hamit")
        .surname("Mizrak")
        .email("hamitmizrak@gmail.com")
        .password("Hm123456@.")
        .date(new Date(System.currentTimeMillis()))
        .build();
    log.info(registerDto);
    return ResponseEntity.ok(registerDto);
}

```

```
// http://localhost:4444/api/v1/rest6?id=1
@GetMapping("/rest6")
public RegisterDto getRest6(@RequestParam(name="id") Long id){
    return RegisterDto.builder()
        .id(id)
        .name("Hamit")
        .surname("Mizrak")
        .email("hamitmizrak@gmail.com")
        .password("Hm123456@.")
        .build();
}
```

```
// http://localhost:4444/api/v1/rest7?id=1&surname=Mizrak
@GetMapping("/rest7")
public RegisterDto getRest7(@RequestParam(name="id") Long id, @RequestParam(name="surname") String surname){
    return RegisterDto.builder()
        .id(id)
        .name("Hamit")
        .surname(surname)
        .email("hamitmizrak@gmail.com")
        .password("Hm123456@.")
        .build();
}
```


CORS

- ▶ CORS: Cross Origin Resource Sharing
- ▶ Farklı uygulamaların farklı portlar arasındaki iletişim kurup kurmamaya denir.

Java

```
package com.hamitmizrak.springboot._3_SpringAPI;

import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

//CORS
@RestController
@RequestMapping("/api/v1")
@CrossOrigin
public class CorsApi {

    // http://localhost:4444/api/v1/cors
    @GetMapping("/cors")
    public String message(){
        return "Merhabalar Ben Java API'dan geldim";
    }
}
```

index.html → 1

```
<!doctype html>
<html lang="en">

<head>
  <title>CORSS jQuery</title>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS v5.2.1 -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

  <!-- jQuery CDN -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
</head>
```

index.html → 2

<body>

<h1>Cors Application</h1>

<p id="cors_id" class="text-center text-primary text-uppercase mb-5 mt-5">Normal Bir yazı</p>

<script>

\$(document).ready(function () {

const URL = "http://localhost:4444/api/v1/cors";

\$.get(URL, function (data, status) {

//log

console.log(data);

console.log(status);

if(status==="success"){

//paragraf

\$("#cors_id").html('\${data}')

}else{

\$("#cors_id").html('Api Gelmiyor')

}

})

});

</script>


```
<!-- Bootstrap JavaScript Libraries -->
```

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
  integrity="sha384-oBqDVmMz9ATKxleP9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlhmh/jp3" crossorigin="anonymous">
</script>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"
  integrity="sha384-7VPbUDkoPSGFnVtYi0QogXtr74QeVeels99Qfg5YCF+TidwNdjvaKZX19NZ/e6oz" crossorigin="anonymous">
</script>
```

```
</body>
```

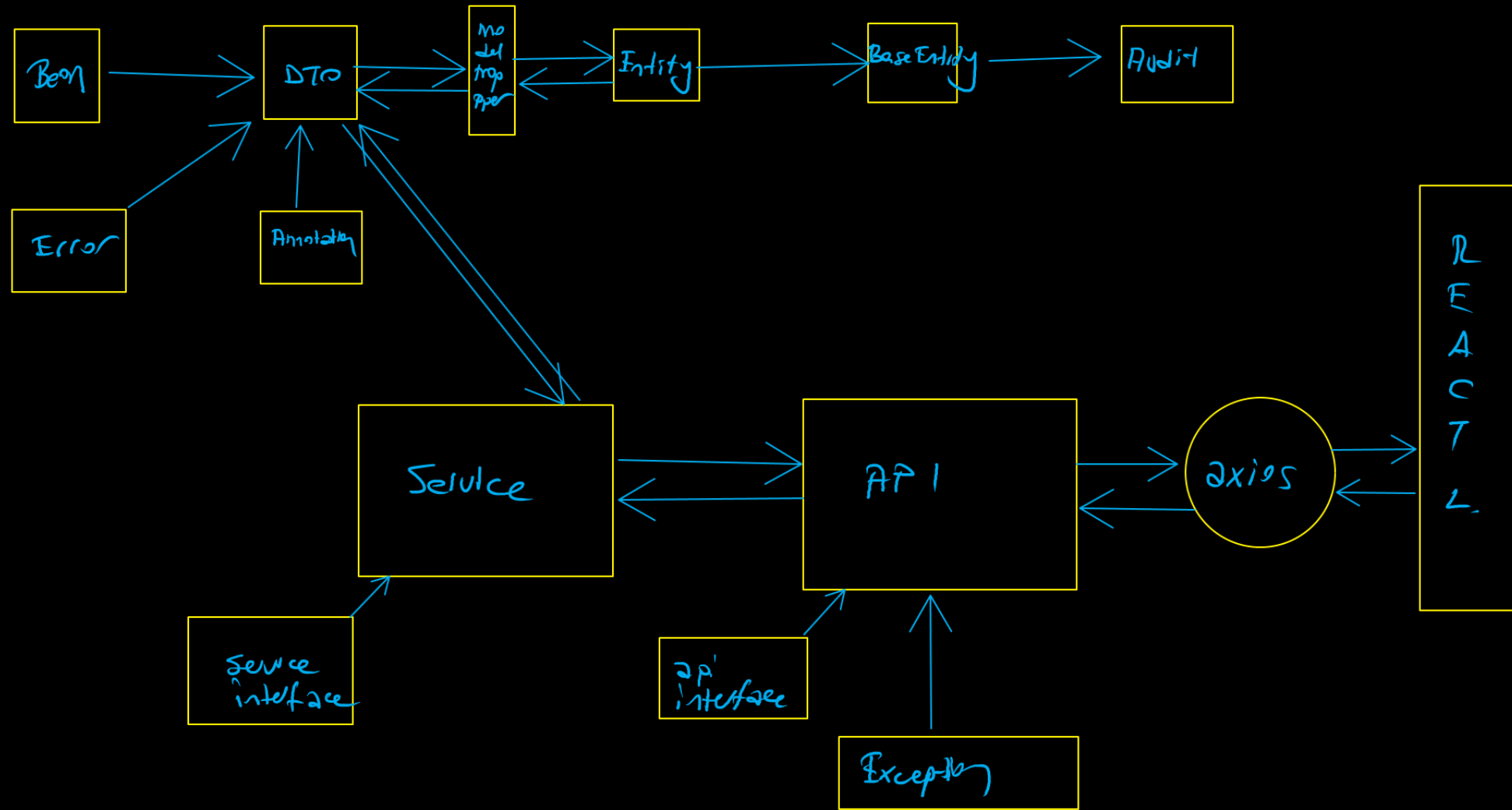
```
</html>
```

Teknolojiler

- ▶ Spring MVC +Thymeleaf
- ▶ Spring DATA
- ▶ Spring API
- ▶ Spring SECURITY

Spring Boot

- ▶ Java SE
- ▶ Java EE= Servlet , JSP ,jstl , JSF, Spring Framework, Spring boot
- ▶ Java ME



Spring boot

- ▶ spring mvc
- ▶ spring data
- ▶ spring security
- ▶ spring api
- ▶ spring Cloud
- ▶ React
- ▶ Lombok
- ▶ modelmapper
- ▶ Google Chrome

Bean

- Spring'e tanıtmak için kullandığımız dosya yapısıdır.

Exception

- istisna yakalamak

ApiResponse

- ▶ Spring'in hata yakalamasını kendim handleli customize.

Proje Görüntüsü

← → ↺ 🏠

localhost:2222/customer/list

🔍 📄 ⭐ 🌐 🛠️ 🗑️ ⚙️

🔍 Google Çeviri 📄 jUniConv: Unicode... 📄 Get started with Bo... 📄 Font Awesome 📄 Material Symbols a... 📄 CDN JS 📄 jUniConv: Unicode... 📄 Turkcell 📄 Code Search 📄 Introducing ChatGPT 📄 Blog Register

LIST

EKLE

Hızlı Ekle

🗑️

Show entries

Search:

ID	ADI	SOYADI	EMAIL	ŞİFRE	TARİH	GUNCELLE	GOSTER	SİL
34	adı 1	soyadı 1	b64e10c6-0be6-4178-8d1b-cfa857f3b37a@gmail.com	\$2a\$10\$rh45Q9RXUIU/JHSYnpho9eoA6.OXUBVxG.obZkLfSC0jhfn8wxjTS	2023-04-28 12:33:35.728	🔧	👁️	🗑️
35	adı 2	soyadı 2	0264c20f-f881-4c9b-b4d2-a3e48157b84e@gmail.com	\$2a\$10\$0.EzwemjvRWYJ8vtuqyNEupWqzK32U/E0z5tAGLXPEFgE.FAy4PBG	2023-04-28 12:33:35.839	🔧	👁️	🗑️
36	adı 3	soyadı 3	7ca7db9a-2e65-41b5-b81a-7f24fe2185b9@gmail.com	\$2a\$10\$L5vzwzDqZw9U6E/kZZ8xOvFhPys5d84LPu8W6joltjwxW3lgXDXi	2023-04-28 12:33:35.913	🔧	👁️	🗑️
37	adı 4	soyadı 4	585d831a-61d0-4910-a57b-12341bef0a47@gmail.com	\$2a\$10\$fKFnW2tAVa3cLbNlyG5X.X3IPI88Q7U3vUvoXLLdxuy/gp2IhSCC	2023-04-28 12:33:35.99	🔧	👁️	🗑️
38	adı 5	soyadı 5	b0799715-2aa2-4bab-b56f-5ba76c31b64a@gmail.com	\$2a\$10\$ym9NIJ3SC1/0b83xzZTX.Or0Q.o8xuH52gsHrdRQK75un1tJ.Nb.	2023-04-28 12:33:36.067	🔧	👁️	🗑️

Showing 1 to 5 of 5 entries

Previous

1

Next

Proje Görüntüsü

← → ↺ 🏠 ⓘ localhost:2222/customer/create 🔍 📄 ☆ 🌐 🛠️ 📱 📺 ⚙️

Google Çeviri jUniConv: Unicode... B Get started with Bo... Font Awesome Material Symbols a... <> CDN JS jUniConv: Unicode... Turkcell Code Search Introducing ChatGPT Blog Register

Customer List

CREATE

ÜYE KAYIT

Adı

kullanıcı adını boş geçemezsiniz.

soyadı

kullanıcı soyadını boş geçemezsiniz.

Email

kullanıcı emailini boş geçemezsiniz.customer.email.validation.regex.constraints.NotNull.message = Email regex

password

kullanıcı şifresinde en az 7 karakter olmalı, En az 1 tane küçük harf,1 tane büyük harf ve 1 tane özel simge yazmalısınız. Examples: Hm123456@.
kullanıcı şifresinde en az 7 karakter olmalı, En az 1 tane küçük harf,1 tane büyük harf ve 1 tane özel simge yazmalısınız. Examples: Hm123456@.
kullanıcı şifresini boş geçemezsiniz.

Temizle Gönder

Proje Görüntüsü

← → ↻ 🏠 ⓘ localhost:2222/customer/update/34 🔍 📄 ☆ 👤 🗒️ 📱 📺 ⚙️

🌐 Google Çeviri 📄 jUniConv: Unicode... 📄 Get started with Bo... 📄 Font Awesome 📄 Material Symbols a... 📄 <> CDN JS 📄 jUniConv: Unicode... 📄 Turkcell 📄 Code Search 📄 Introducing ChatGPT 📄 Blog Register

Customer List

UPDATE

ÜYE GÜNCELLE

adı 1

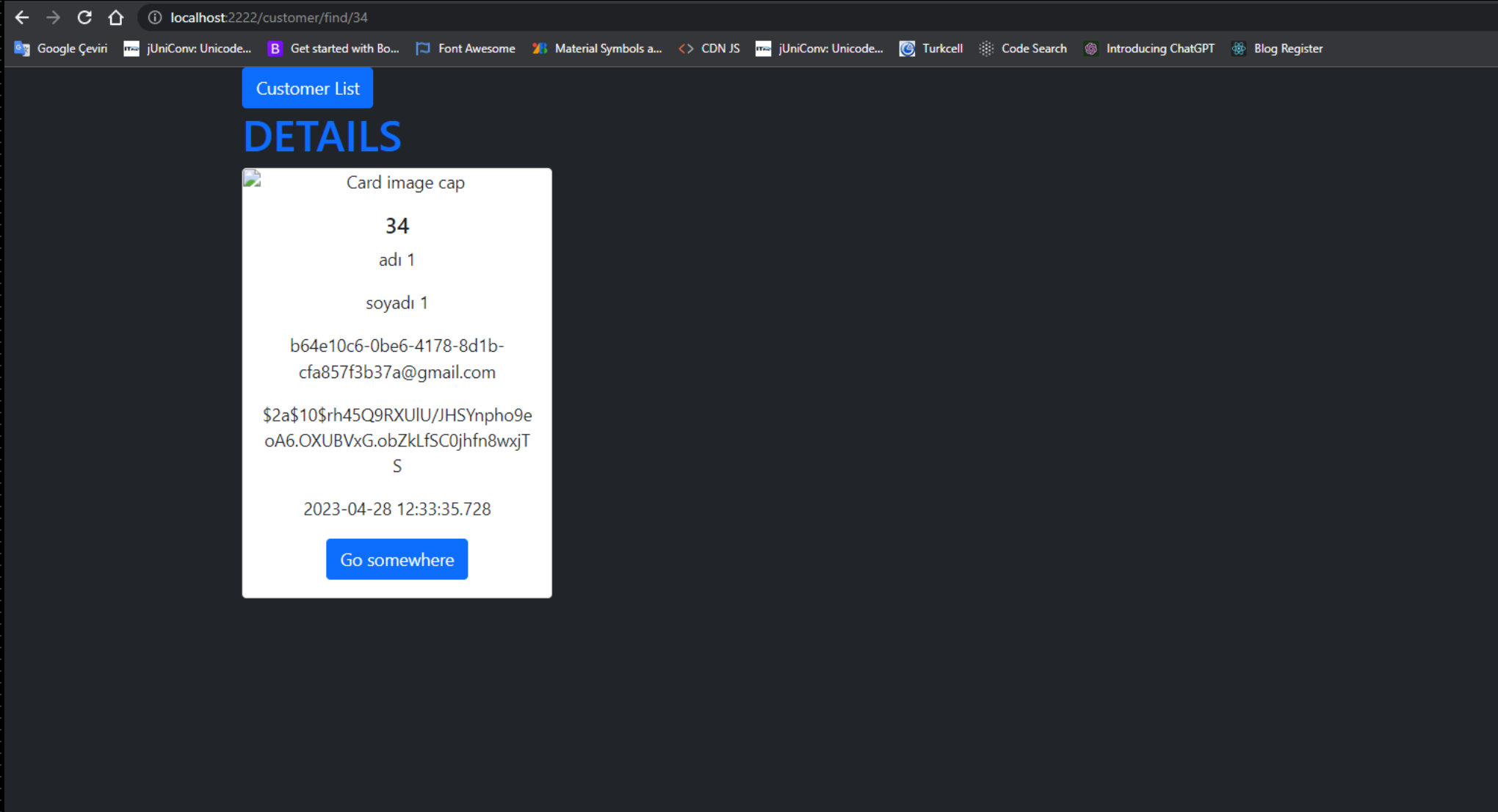
soyadı 1

b64e10c6-0be6-4178-8d1b-cfa857f3b37a@gmail.com

\$2a\$10\$rh45Q9RXUIU/JHSYnpho9eoA6.OXUBVxG.obZkLfSC0jhfn8wxjTS

Temizle Gönder

Proje Görüntüsü



Proje Görüntüsü

localhost:2222/customer/list

Google Çeviri jUniConv: Unicode... B Get started with Bo... Font Awesome Material Symbols a... <> CDN JS jUniConv: Unicode... Türkcell Code Search Introducing ChatGPT Blog Register

LIST

EKLE Hızlı Ekle

Show 10 entries Search:

ID	ADI	SOYADI	EMAIL	ŞİFRE	TARİH	GUNCELLE	GOSTER	SİL
34	adı 1	soyadı 1	b64e10c6-0be6-4178-8d1b-cfa857f3b37a@gmail.co		fn8wxjTS 2023-04-28 12:33:35.728			
35	adı 2	soyadı 2	0264c20f-f881-4c9b-b4c-a3e48157b84e@gmail.co		E.FAy4PBG 2023-04-28 12:33:35.839			
36	adı 3	soyadı 3	7ca7db9a-2e65-41b5-b81a-7f24fe2185b9@gmail.co		3lgXDXi 2023-04-28 12:33:35.913			
37	adı 4	soyadı 4	585d831a-61d0-4910-a57b-12341bef0a47@gmail.com		p2lhSCC 2023-04-28 12:33:35.99			
38	adı 5	soyadı 5	b0799715-2aa2-4bab-b56f-5ba76c31b64a@gmail.com	\$2a\$10\$ym9NIJ3SC1/0b83xzZTX.Or0Q.o8xuH52gsHrdRQK75un1tJ.Nb.	2023-04-28 12:33:36.067			

Showing 1 to 5 of 5 entries

Previous 1 Next

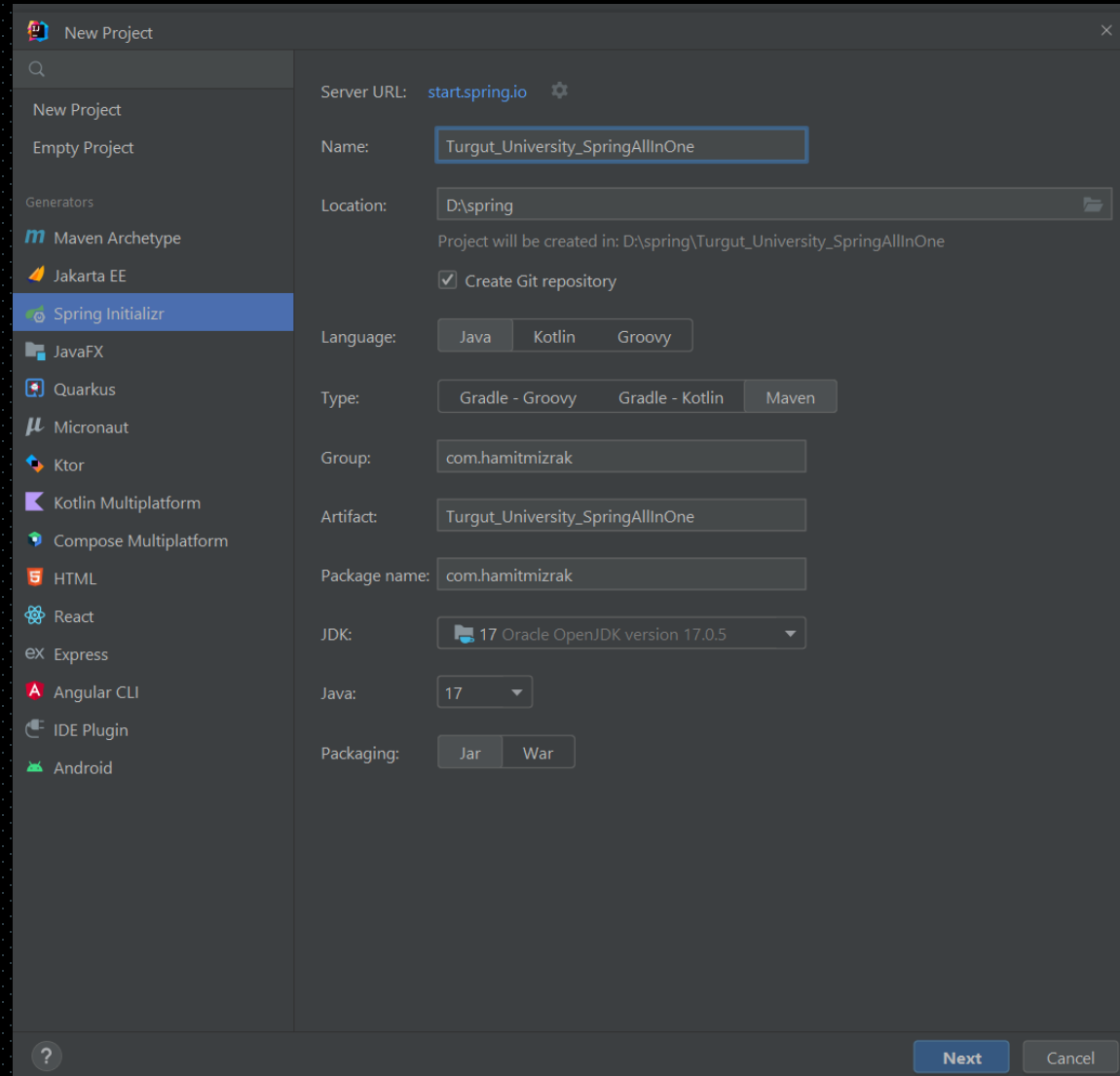
Silmek istediğinizden emin misiniz?

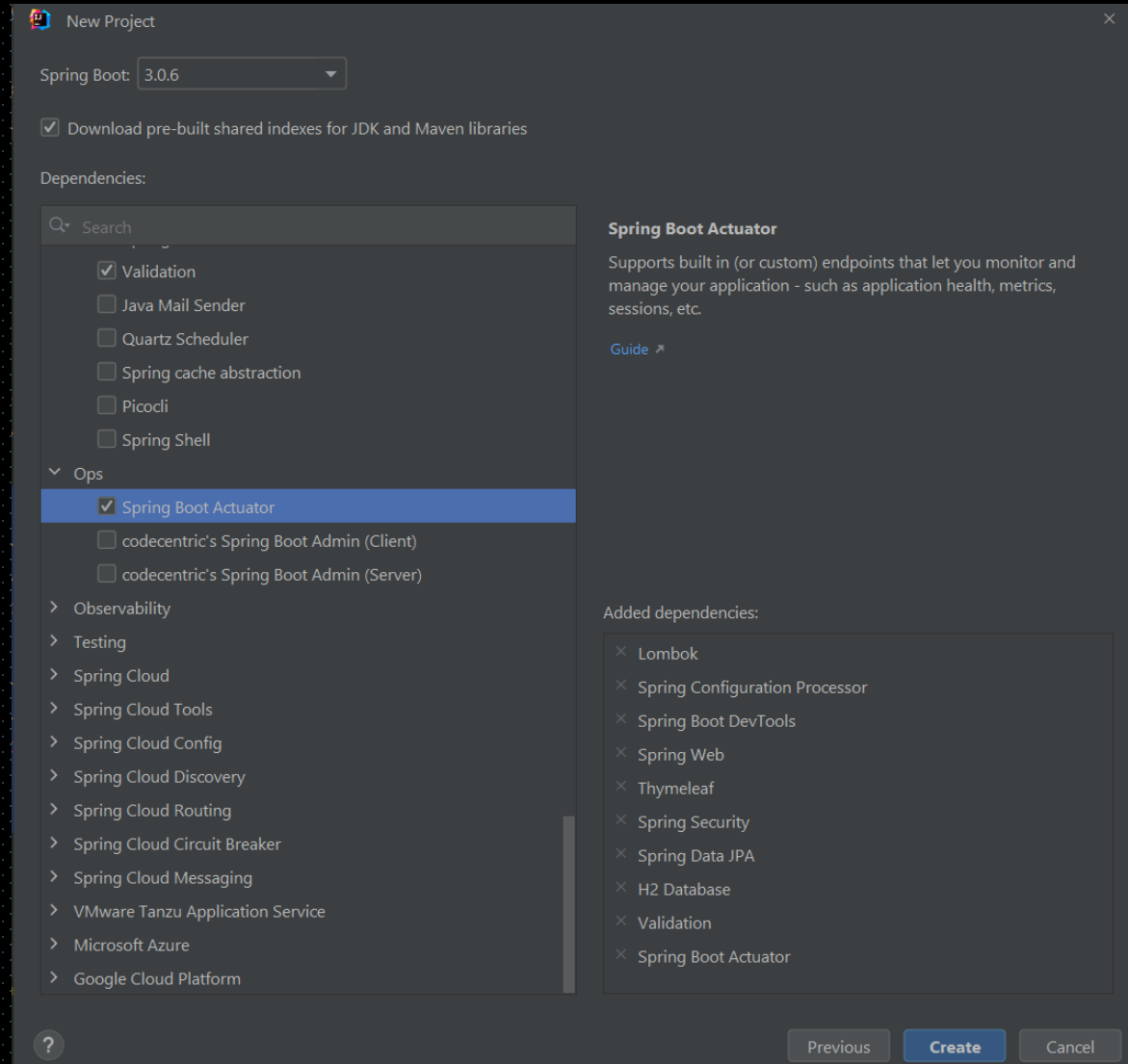
Silindikten sonra tekrar geri alınamaz.

İptal Sil

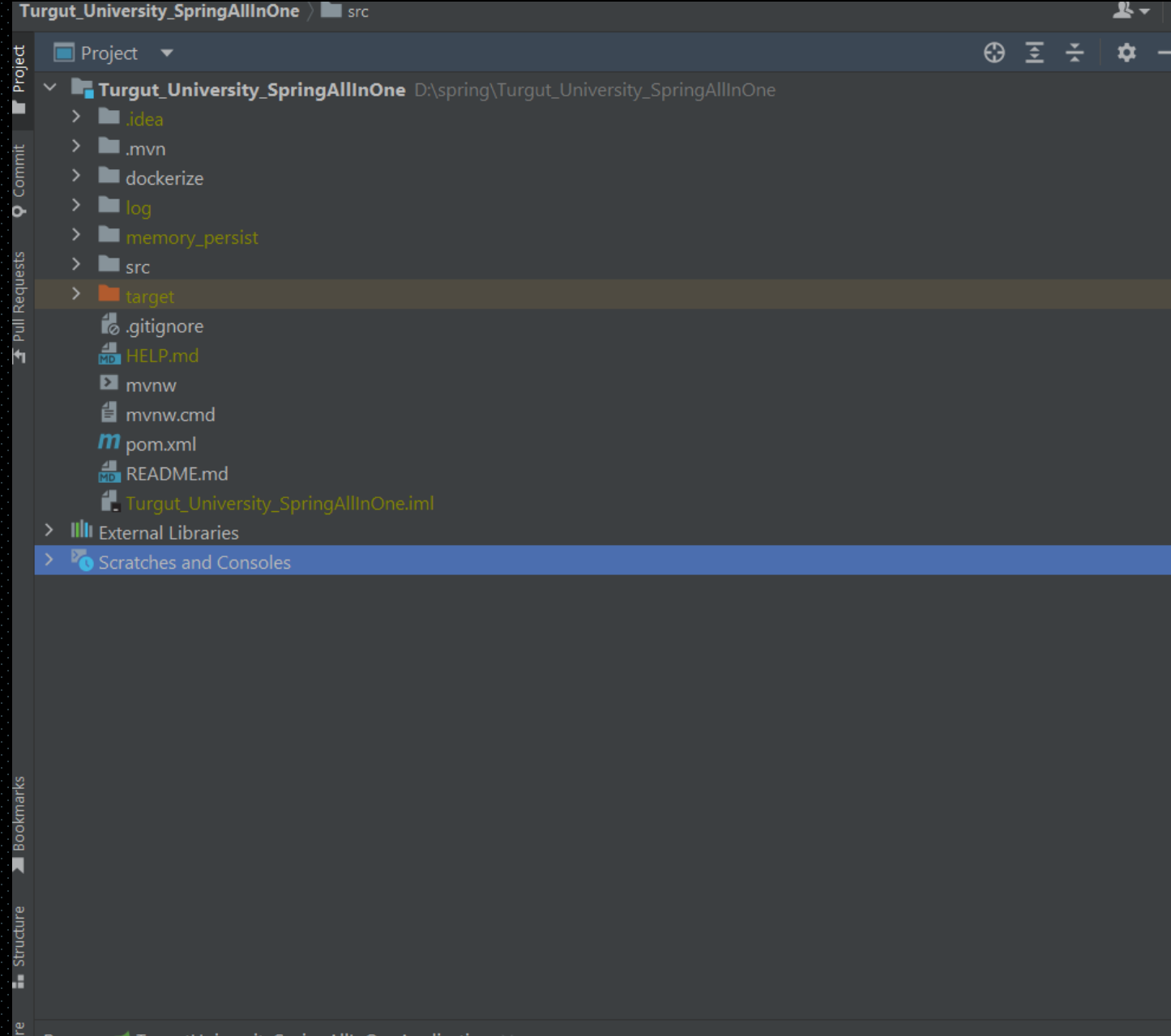
```
package com.hamitmizrak.persist;
```

```
public class GitHubPersist {  
    //https://github.com/hamitmizrak/Turgut_University_SpringAllInOne  
}
```





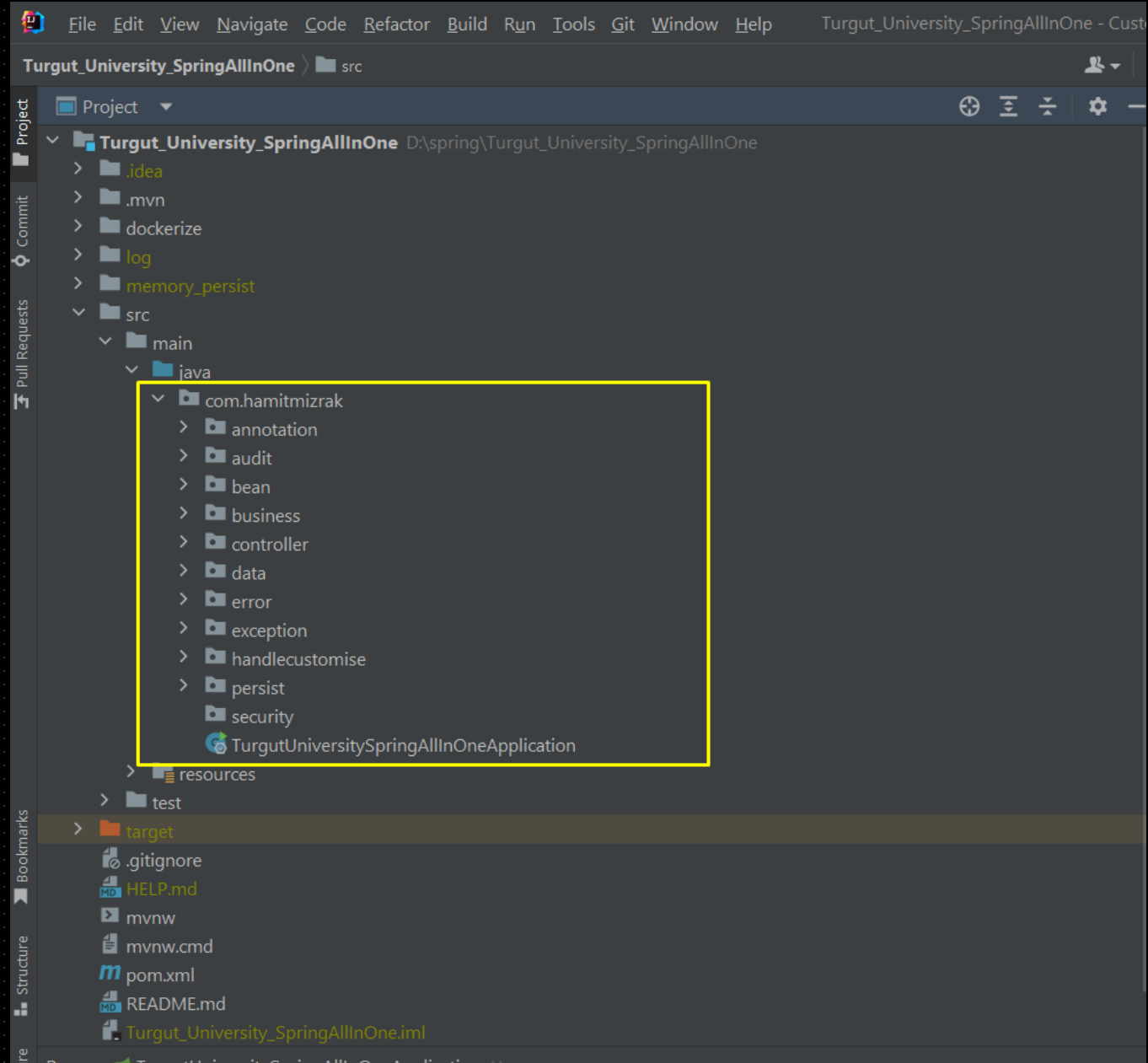
proje genel görüntüsü



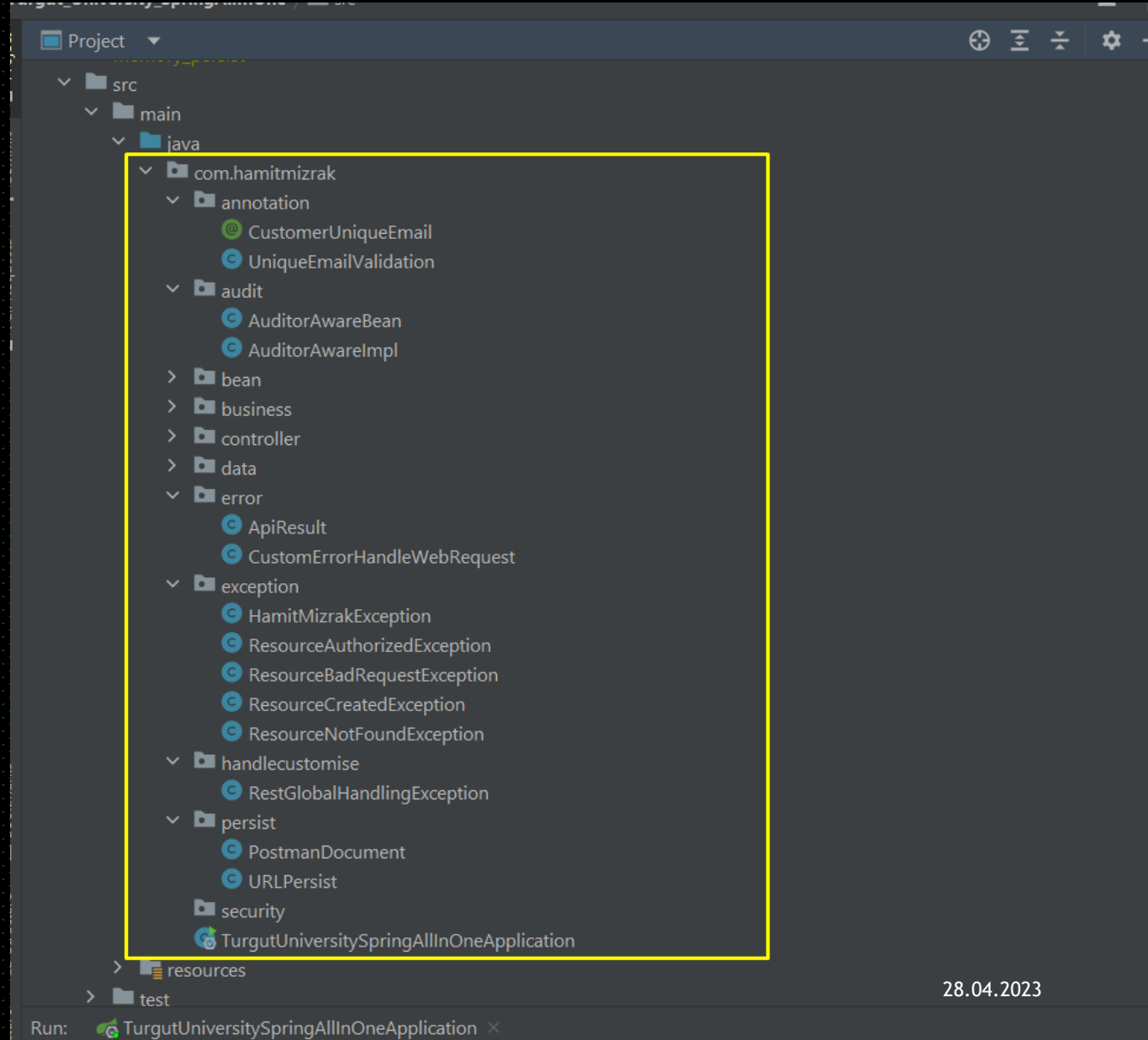
JAVA FULL STACK DEVELOPER HAMİT MIZRAK



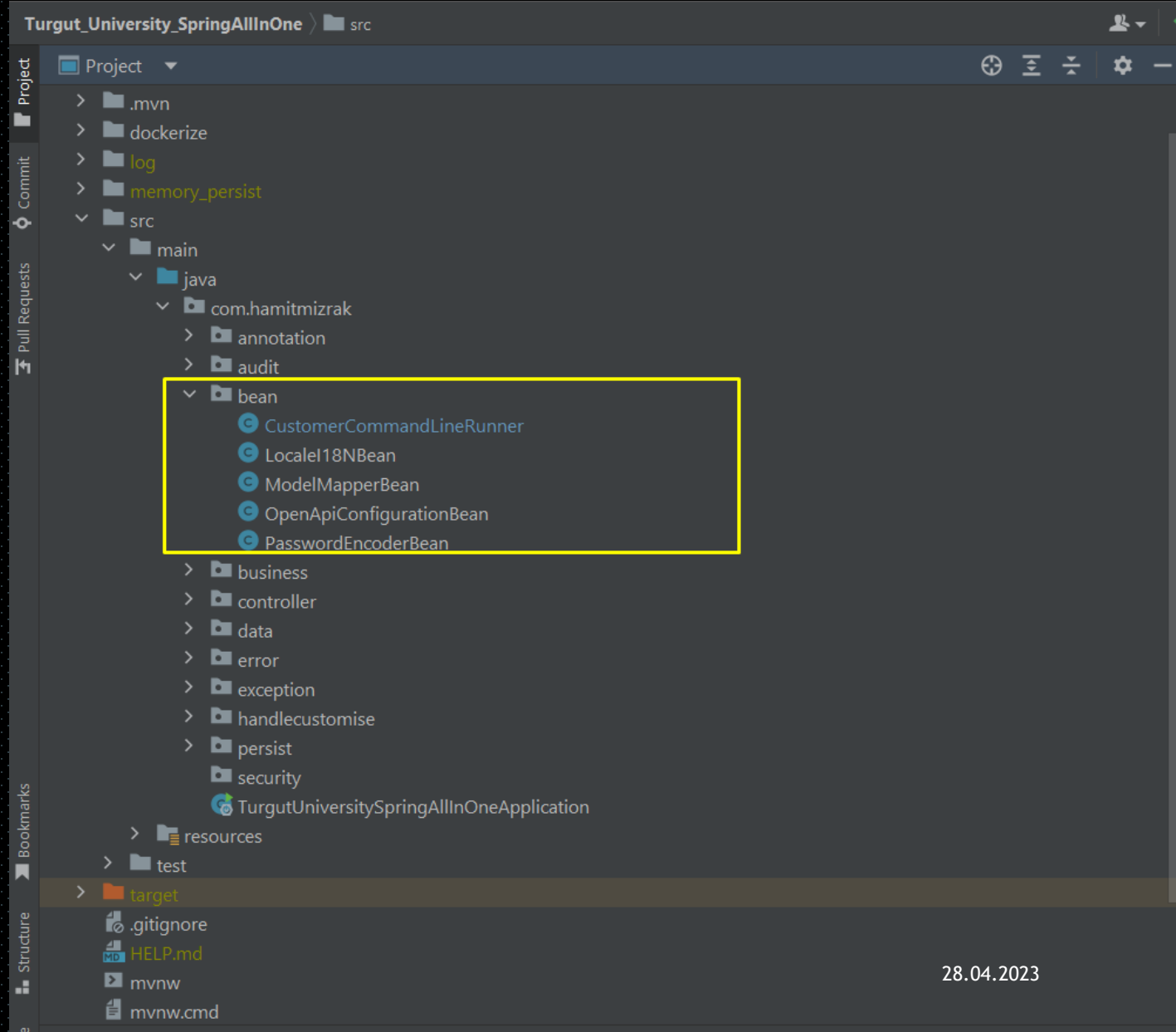
proje genel görüntüsü



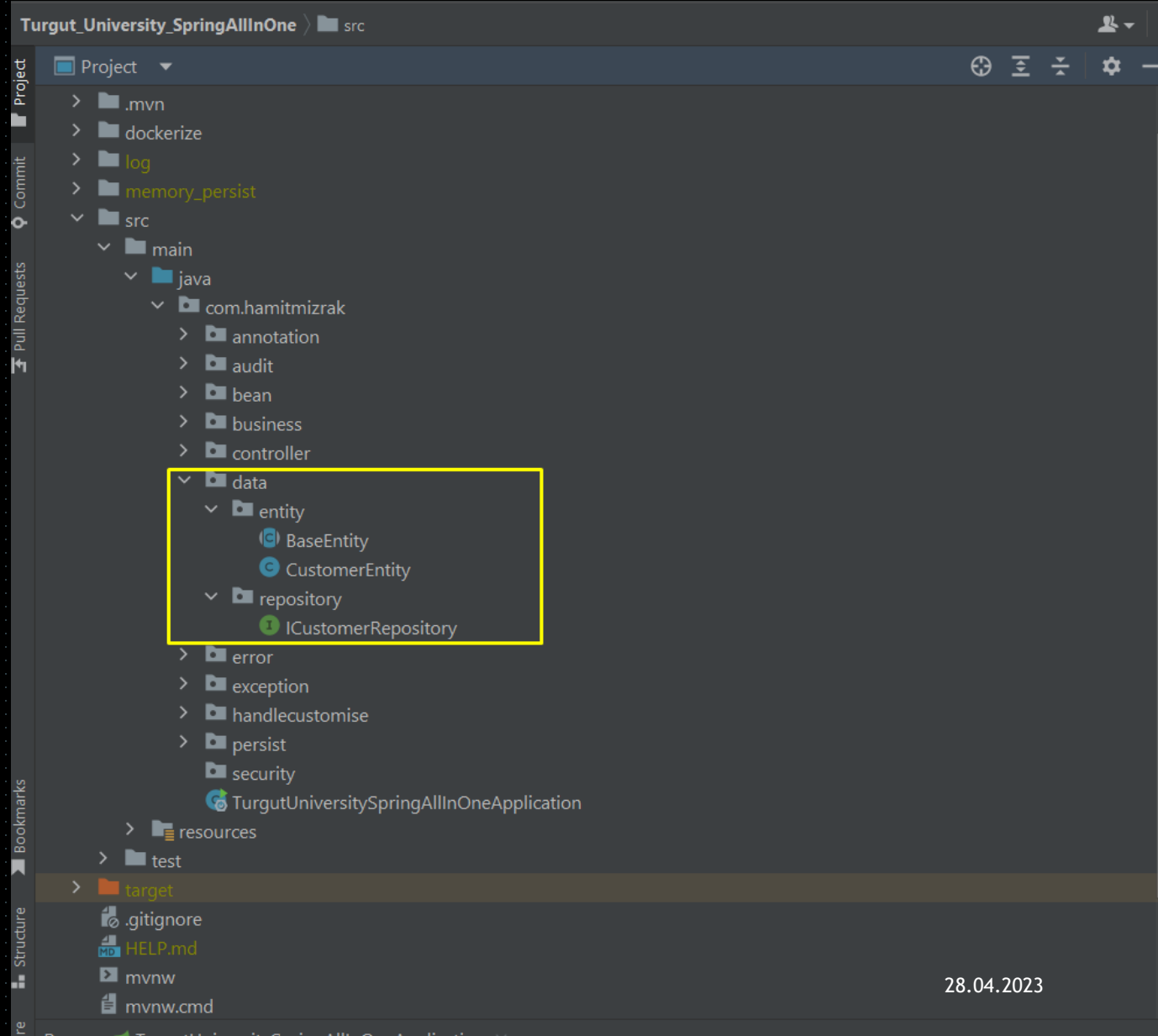
proje genel görüntüsü



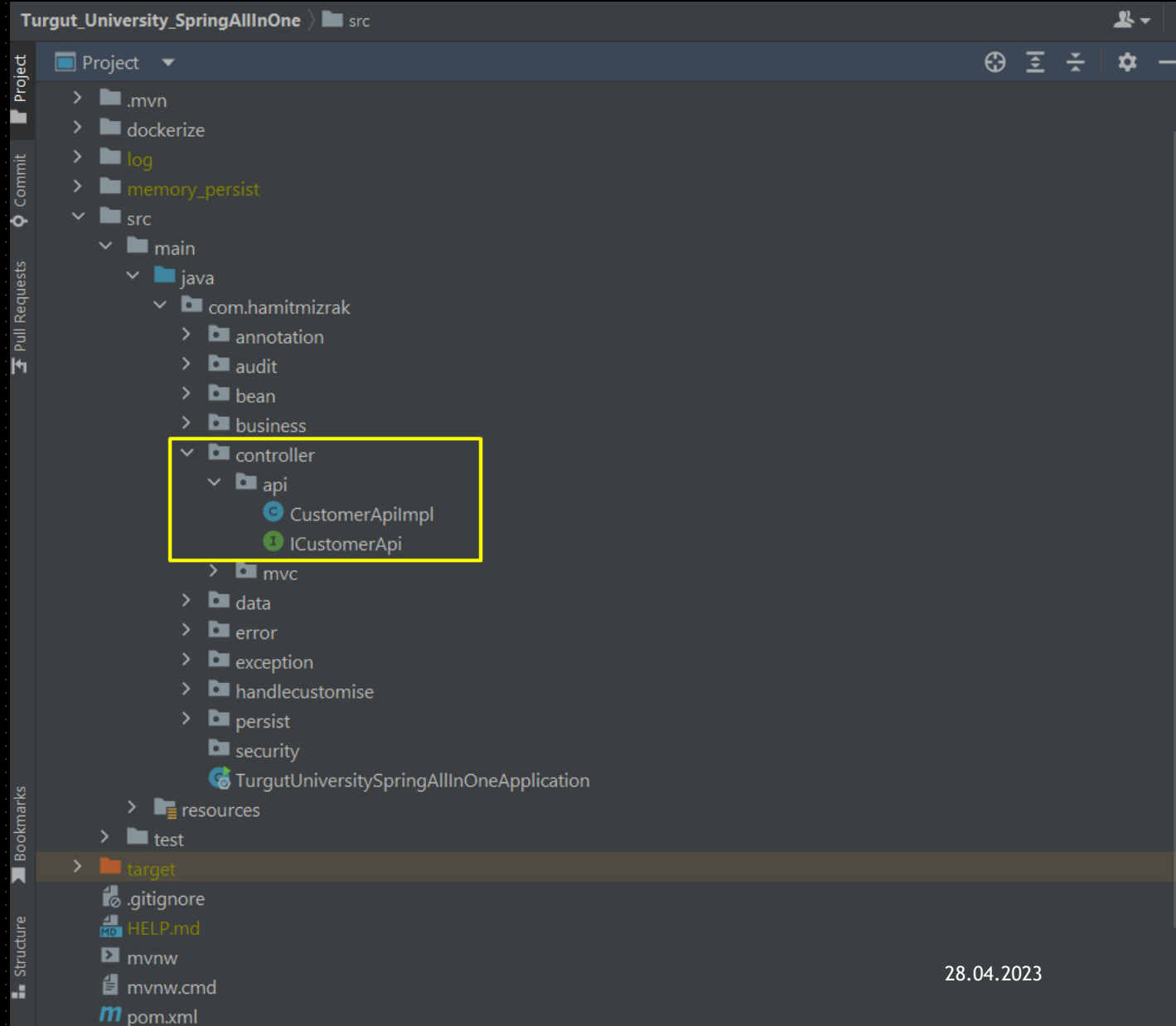
proje genel görüntüsü



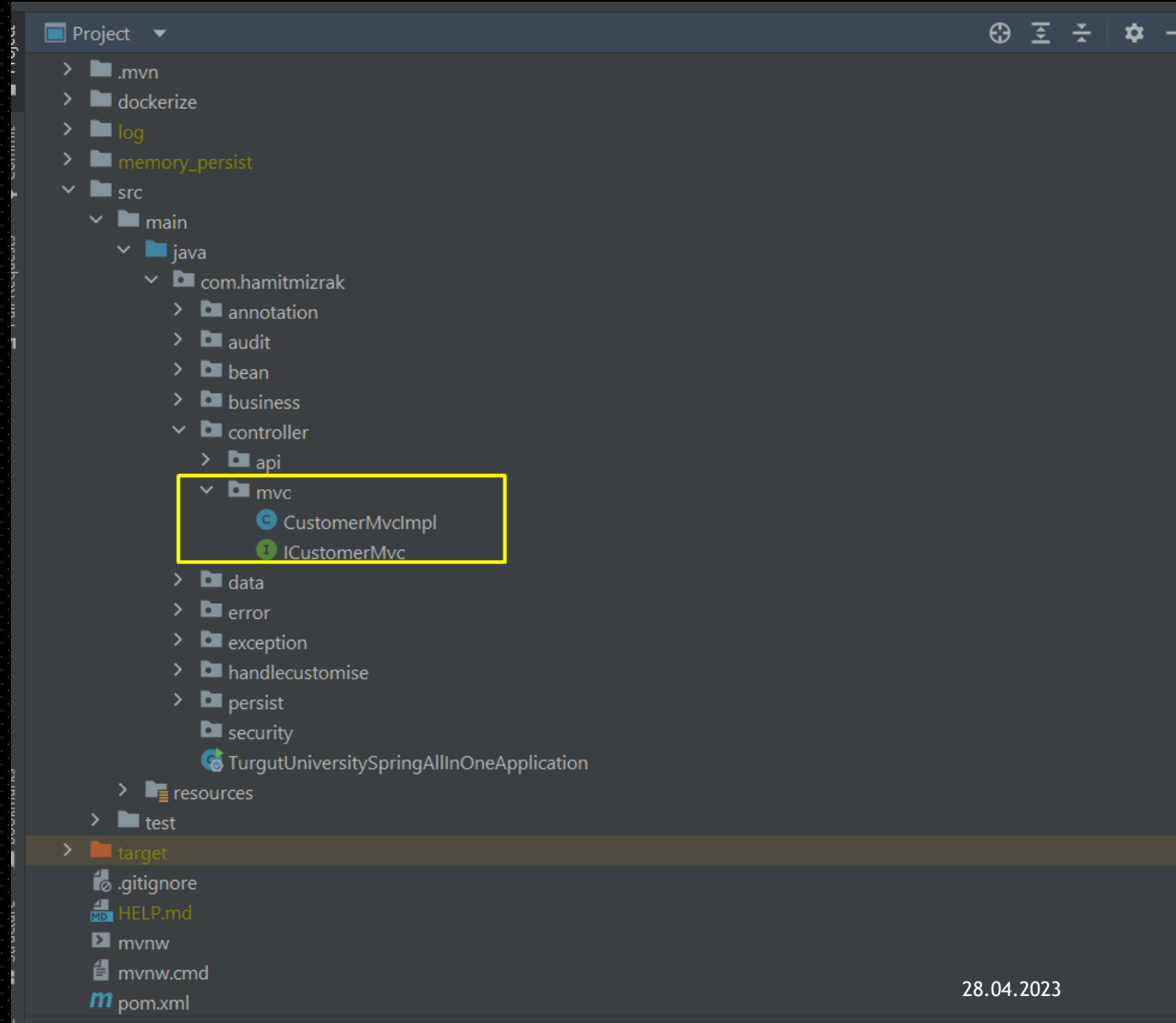
proje genel görüntüsü



proje genel görüntüsü



proje genel görüntüsü



pom.xml → 1

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <!--parent-->
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.0.6</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>

    <!--for project informartion-->
    <groupId>com.hamitmizrak</groupId>
    <artifactId>Turgut_University_SpringAllInOne</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>Turgut_University_SpringAllInOne</name>
    <description>Turgut_University_SpringAllInOne</description>

    <!--properties-->
    <properties>
        <encoding>UTF-8</encoding>
        <java.version>17</java.version>
        <!--for maven-->
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <maven.compiler.target>17</maven.compiler.target>
        <maven.compiler.source>17</maven.compiler.source>

        <!--Version-->
        <modelmapper.version>2.4.4</modelmapper.version>
        <swagger.version>3.0.0</swagger.version>
        <gson.version>2.8.6</gson.version>
    </properties>
```

pom.xml → 2

```
<!--dependencies-->
<dependencies>
  <!--Spring Web Mvc Rest-->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <!-- Spring data-jpa-->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>

  <!--security-->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-test</artifactId>
    <scope>test</scope>
  </dependency>

  <!--actuator-->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
  </dependency>
</dependencies>
```

pom.xml → 3

```
<!--lombok-->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>

<!--Validation-->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-validation</artifactId>
</dependency>

<!-- model mapper -->
<dependency>
  <groupId>org.modelmapper</groupId>
  <artifactId>modelmapper</artifactId>
  <version>${modelmapper.version}</version>
</dependency>

<!--Templates thymeleaf-->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

pom.xml → 4

```
<!-- swagger open-api-3.0-->
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-boot-starter</artifactId>
  <version>${swagger.version}</version>
</dependency>

<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-webflux-ui</artifactId>
  <version>1.6.12</version>
</dependency>

<!--Database-->
<!--h2DB-->
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>

<!--Mysql-->
<dependency>
  <groupId>com.mysql</groupId>
  <artifactId>mysql-connector-j</artifactId>
  <scope>runtime</scope>
</dependency>

<!--postgresql-->
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <scope>runtime</scope>
</dependency>
```

pom.xml → 5

```
<!--TEST-->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>

<!--tools-->
<!-- JSON işlemleri için gson-->
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>${gson.version}</version>
</dependency>

<!--devtools:otomatik compiler için-->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-devtools</artifactId>
  <optional>true</optional>
  <scope>runtime</scope>
</dependency>

<!--application.properties ==> application.yml-->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-configuration-processor</artifactId>
  <optional>true</optional>
</dependency>
</dependencies>
```

pom.xml → 6

```
<build>
  <plugins>
    <!-- for maven -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-resources-plugin</artifactId>
      <version>3.0.2</version>
    </plugin>

    <!-- for maven target 17 -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version>
      <configuration>
        <source>17</source>
        <target>17</target>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <excludes>
          <exclude>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
          </exclude>
        </excludes>
      </configuration>
    </plugin>
  </plugins>
</build>

</project>
```

gitignore

```
# special data
log
memory_persist
blog/node_modules
dockerize/JavaTutorialsCommonData-0.0.1-SNAPSHOT.jar
```

```
HELP.md
target/
!.mvn/wrapper/maven-wrapper.jar
!**/src/main/**/target/
!**/src/test/**/target/
```

```
### STS ###
.appt_generated
.classpath
.factorypath
.project
.settings
.springBeans
.sts4-cache
```

```
### IntelliJ IDEA ###
.idea
*.iws
*.iml
*.ipr
```

```
### NetBeans ###
/nbproject/private/
/nbbuild/
/dist/
/nbdist/
/.nb-gradle/
build/
!**/src/main/**/build/
!**/src/test/**/build/
```

```
### VS Code ###
.vscode/
```

application.properties → 1

```
#####  
### System config #####  
# Port  
# http://localhost:2222  
server.port=2222  
# terminalde => java -jar spring-5.jar --server.port=8083  
# spring.application.admin.enabled=true  
#logging.level.org.springframework.boot.autoconfigure=ERROR  
#spring.main.allow-bean-definition-overriding=true  
# kendi hatam ici message  
# security.basic.enable= false  
# security.ignored=/**  
# server.error.include-exception=true  
# server.error.include-binding-errors=always  
# server.error.include-message=always  
# uygulama adini degistirmek  
# spring.application.name=proje_adi  
# Uygulamanin yoneticisi ozelliklerini etkinlestirmek icin kullanilir  
# spring.application.adminPage.enabled=true  
## Sunucunun baglanmasi gereken ad adresini ayarlamak icin kullanilirerror.  
## http://127.1.1.1:2222/actuator  
# server.address=127.0.0.0  
springdoc.show-actuator=true
```


application.properties → 2

```
#####  
### White label #####  
server.error.whitelabel.enabled=true  
  
#####  
### Api Result #####  
api.result.error=Genel Exception  
api.result.message=Rest Global Handling Exception  
  
#####  
### Circular Referance #####  
spring.main.allow-circular-references=true  
#####  
###Pageable #####  
spring.data.web.pageable.page-parameter=currentPage  
spring.data.web.pageable.size-parameter=pageSize  
spring.data.web.pageable.default-page-size=10  
spring.data.web.pageable.max-page-size=1000  
  
#####  
###jackson #####  
spring.jackson.mapper.default-view-inclusion=true
```

application.properties → 3

```
#####  
### Context #####  
##localhost:2222/java  
# server.servlet.context-path=/java  
#Uygulama baz?ndan almak istiyorsak  
server.servlet.context-parameters.my_special.name=Hamit  
user.surname=Mizrak  
  
#####  
### Profiles #####  
spring.profiles.active=frontend  
#spring.profiles.active=backend  
  
#####  
### Prometheus: Monitoring #####  
#management.endpoints.web.exposure.include=*  
#management.endpoint.health.show-details=always  
### Redis config #####  
#spring.cache.type=redis  
#spring.cache.host=localhost  
#spring.cache.port=6379
```

application.properties → 4

```
### GraphQL config #####
# GraphQL
#graphql.servlet.mapping=/graphql
#graphql.servlet.enabled=true
#graphql.servlet.corsEnabled=true
## GraphiQL
#graphiql.mapping=/graphiql
#graphiql.endpoint=/graphql
#graphiql.enabled=true
#graphiql.cdn.enabled=true
#graphiql.cdn.version=0.11.11
#graphql.tools.schema-location-pattern=**/*.graphiql
#graphql.graphiql.enabled=true
#spring.graphql.graphiql.enabled=true
#spring.graphql.graphiql.path=/graphiql
### Session Time out #####
# adminPage bir ?ey yapmaazsa 30 dakika sonra kapan?r
# server.servlet.session_timeout=1m
server.servlet.session_timeout=30m
# server.servlet.session.cookie.http-only=true
# server.servlet.session.cookie.secure=true
```

application.properties → 5

```
#####  
### validation database yazmadan kontrol edilmesin #####  
#spring.jpa.properties.javax.persistence.validation.mode=none  
  
#####  
### Debug #####  
# debug aktif etmek  
# debug=true  
  
#####  
### Swagger(Open Api config) #####  
# localhost:8080/swagger-ui.html  
# swagger open api 3.0  
# http://localhost:2222/swagger-ui.html  
# http://localhost:2222/api-docs/  
# http://localhost:2222/v3/api-docs/  
springdoc.api-docs.path=/api-docs\  
springdoc.swagger-ui.path=/swagger-ui/index.html\  
#springdoc.packagesToScan=com.hamitmizrak.api.DenemeApi\  
#springdoc.pathsToMatch=/**
```

application.properties → 6

```
#####  
### Log #####  
# Loglama Asenkron  
# sabit bir yerde saklamal?y?z.  
logging.file.name=./log/blog.log  
#spring hatalar? almak i?in  
#logging.level.org.springframework=DEBUG  
logging.level.org.springframework.web=DEBUG  
logging.level.org.hibernate=ERROR  
# logging.level.org.springframework.security.web.FilterChainProxy=DEBUG  
# Gunluk dosyasinin konumunu yap?land?r?r.  
# logging.file.path=./  
# logging.pattern.console=%d{yyyy-MM-dd HH:mm:ss} - %msg%n  
# Log LEVEL (7)  
# O F F W I D T  
# OFF > FATAL > ERROR > WARN > INFO > DEBUG > TRACE  
logging.level.root=INFO  
  
#####  
### JWT #####  
#jwt.secret=secretOfUsDontBeShy  
#jwt.expire.time=86400000  
#jwt.token.prefix=Bearer  
#jwt.header.string=Authorization  
#jwt.secret=hamitmizrak
```

application.properties → 7

```
#####  
### DevTools #####  
# DevTool restart Enable  
# spring.devtools.restart.enabled=true  
# spring.devtools.livereload.enabled=true  
# spring.devtools.livereload.port=35729  
#uzaktan yönetmek  
  
#####  
### Spring Security #####  
# Security 1.YOL KAPATMAK (@SpringBootApplication kapatmak istersek)  
# @SpringBootApplication(exclude = {SecurityAutoConfiguration.class})  
  
# Security 2.YOL KAPATMAK  
# spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.security.SecurityAutoConfiguration  
  
# WebSecurity için kullanacağım  
spring.security.user.name=root1  
spring.security.user.password=root1  
spring.security.user.roles=ROLES_ADMIN  
# securityConfigurations  
# .formLogin()  
# .loginPage("/login")  
# .usernameParameter("username")  
# .passwordParameter("password")
```

application.properties → 8

```
#####  
### Mail #####  
# Varsayilan MimeMessage kodlamasiniayarmak i?in kullanilir  
#spring.mail.default-encoding=UTF-8  
## Mail serverin adresi  
#spring.mail.host=smtp.gmail.com  
#spring.mail.port=587  
#spring.mail.username=hrpmuhendislik44@gmail.com  
#spring.mail.password=gkoerkwnwnyxxtku  
#spring.mail.properties.mail.smtp.auth=true  
#spring.mail.properties.mail.smtp.starttls.enable=true  
#spring.mail.properties.mail.smtp.starttls.required=true  
#spring.mail.properties.mail.smtp.connectiontimeout=5000  
#spring.mail.properties.mail.smtp.timeout=5000  
#spring.mail.properties.mail.smtp.writetimeout=5000
```

application.properties → 9

```
#####  
### Database #####  
## H2Database  
logging.level.org.springframework.jdbc.core=TRACE  
spring.jpa.properties.hibernate.format_sql=true  
spring.jpa.properties.hibernate.show_sql=true  
spring.jpa.show-sql=true  
  
# once create sonra durdur update yap  
#jpa > none update create, create-drop, validate  
spring.jpa.hibernate.ddl-auto=update
```


application.properties → 10

```
## Generic JNDI Data Source data 44
##in Memory
##spring.datasource.url=jdbc:h2:mem:denemedb
## Directory on Disk
## spring.datasource.url=jdbc:h2:file:C:/data/denemedb
##Current User Directory
##spring.datasource.url=jdbc:h2:file:~/denemedb
## Current Working Directory
## http://localhost:2222/h2-console
## following setting h2-console
spring.h2.console.settings.web-allow-others=true
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.url=jdbc:h2:file:./memory_persist/spring
##spring.datasource.url=jdbc:h2:file:./memory_persist/register;useSSL=falseDB_CLOSE_ON_EXIT=FALSE;AUTO_SERVER=TRUE
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
```

application.properties → 11

```
## Mysql
##CREATE SCHEMA `employee_database` DEFAULT CHARACTER SET utf8 COLLATE utf8_turkish_ci ;root
## use employee_database;
#spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
#spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
#spring.datasource.url=jdbc:mysql://localhost:3306/register?createDatabaseIfNotExist=true&autoReconnect=true&useSSL=false
#spring.datasource.username=root
#spring.datasource.password=root

## Mongo DB
## Robo 3T
## @SpringBootApplication içinde bunu yaz ==> @EnableMongoRepositories
## @Repository ==> public interface IDeneme extends MongoRepository<EntityName,Long>{}
#spring.data.mongodb.database=mongodb
#spring.data.mongodb.uri=mongodb://localhost:27017

## Postgresql
#spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation=true
#spring.datasource.url=jdbc:postgresql://localhost:5432/employee_database
#spring.datasource.username=postgres
#spring.datasource.password=root
#spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.PostgreSQLDialect

# Oracle
#spring.datasource.url=jdbc:oracle:thin:@localhost:1521:orcl
#spring.datasource.username=system
#spring.datasource.password=Password123
```

messages.properties

Site unicode character: <https://itpro.cz/juniconv/>

CUSTOMER

customer.id=ID

customer.name=Ad\u0131

customer.surname=Soyad\u0131

customer.email=Email

customer.password=\u015Eifre

customer.date=\Tarih

customer.update=Guncelle

customer.show=Goster

customer.delete=Sil

ValidationMessages.properties

Site unicode character: <https://itpro.cz/juniconv/>

ENGLISH

customer.name.validation.constraints.NotNull.message = Name cannot be null

customer.surname.validation.constraints.NotNull.message = Surname cannot be null

customer.email.validation.constraints.NotNull.message = Email address cannot be null

customer.email.validation.regex.constraints.NotNull.message = Email regex

customer.email.validation.unique.constraints.NotNull.message=There is such a user in the system, another email experiment

customer.password.validation.constraints.NotNull.message = Password cannot be null

customer.password.pattern.validation.constraints.NotNull.message =Password must least 1 uppercase , 1 lowercase , 1 special character

ValidationMessages_tr.properties

Site unicode character: <https://itpro.cz/juniconv/>

TURKISH

customer.name.validation.constraints.NotNull.message = kullan\u0131c\u0131 ad\u0131n\u0131 bo\u015f ge\u00e7emezsiniz.

customer.surname.validation.constraints.NotNull.message = kullan\u0131c\u0131 soyad\u0131n\u0131 bo\u015f ge\u00e7emezsiniz.

customer.email.validation.constraints.NotNull.message=kullan\u0131c\u0131 emailini bo\u015f ge\u00e7emezsiniz.\

customer.email.validation.regex.constraints.NotNull.message = Email regex

customer.email.validation.unique.constraints.NotNull.message=Sistemde bu email bulunuyor Farkli bir email address giriniz.

customer.password.validation.constraints.NotNull.message = kullan\u0131c\u0131 \u015fifresini bo\u015f ge\u00e7emezsiniz.

customer.password.pattern.validation.constraints.NotNull.message =kullan\u0131c\u0131 \u015fifresinde en az 7 karakter olmal\u0131, En az 1 tane k\u00fc\u00e7\u00fck harf olmal\u0131.

index.html → 1

```
<!doctype html>
<html th:lang="${#locale.language}" xmlns:th="http://www.thymeleaf.org" >
<head>
  <title>Thymeleaf String</title>

  <!-- Turkce karakter -->
  <meta charset="utf-8">

  <!-- responsive design -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- seo -->
  <meta name="keywords" content="html5,css3,js,jquery,bootstrap,react,javase" />
  <meta name="description" content="" />
  <meta name="author" content="" />

  <!-- font-awesome -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css" />

  <!-- google font -->
  <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:ital@1&display=swap" rel="stylesheet">

  <!-- Bootstrap CSS v5.0.2 -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">
```

index.html → 2

```
<!-- jquery -->
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

<!-- react libraries -->
<script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

<link rel="stylesheet" th:href="@{/css/style.css}">
<script th:src="@{/js/template.js}"></script>
</head>

<body class="container">

<!-- start codes -->
<h1 class="text-primary">Anasayfa Hoşgeldiniz</h1>

<!-- ends codes -->
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"></script>
</body>
</html>
```

ApiResult

```
package com.hamitmizrak.error;

import com.fasterxml.jackson.annotation.JsonInclude;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.extern.log4j.Log4j2;

import java.sql.Date;
import java.util.Map;

// LOMBOK
@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
@Log4j2

// jackson: objeyi json çevirir.
// validationError: null ise dönemsini istemeyebiliriz.
// Eğer Sistemde Null bir değer varsa bu null değeri frontentte göndermemek için
@JsonInclude(JsonInclude.Include.NON_NULL)
// Backentten Frontend' bilgi göndermek istersek biz bu Class yardımıyla hata mesajlarımızı gönderebileceğiz.
public class ApiResult {
```



```
public class ApiResult {  
    //Field  
    private int status;  
    private String path;  
    private String message;  
    private String error;  
    //tarih  
    private Date createdAt = new Date(System.currentTimeMillis());  
    //private String createdAt=nowDate();  
  
    //validationError  
    private Map<String, String> validationErrors;
```

```

//Parametreli constructor
public ApiResponse(int status, String path, String message, String error) {
    this.status = status;
    this.path = path;
    this.message = message;
    this.error = error;
}

//Parametreli constructor
public ApiResponse(int status, String path, String message) {
    this.status = status;
    this.path = path;
    this.message = message;
}

//Tarih Formatter
//Now Date
/*private String nowDate() {
    Locale locale=new Locale("tr","TR");
    SimpleDateFormat simpleDateFormat=new SimpleDateFormat("dd.MMMM.yyyy HH:mm:ss",locale);
    Date date=new Date();
    String change=simpleDateFormat.format(date);
    System.out.println(change);
    return change;
}*/
}

```

CustomExceptionHandler

```
package com.hamitmizrak.error;

import lombok.RequiredArgsConstructor;
import lombok.extern.log4j.Log4j2;
import org.springframework.boot.web.error.ErrorAttributeOptions;
import org.springframework.boot.web.servlet.error.ErrorAttributes;
import org.springframework.boot.web.servlet.error.ErrorController;
import org.springframework.validation.FieldError;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.context.request.WebRequest;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

//LOMBOK
@Log4j2
@RequiredArgsConstructor

// SpringBoot defaulttan gelen error'ı kendimize göre customize yapıyoruz.
@RestController
@CrossOrigin
public class CustomExceptionHandler implements ErrorController {

    //injection
    private final ErrorAttributes errorAttributes;
```

```

//http://localhost:4444/error
//spring'ten gelen /error yakalayıp custom handle yapmak için
@RequestMapping("/error")
public ApiResult handleError(WebRequest webRequest) {
    //ApiResult değişkenlerini atamak
    int status;
    String message, path;
    ApiResult error;

    //Spring 2.3>=
    Map<String, Object> attributes = this.errorAttributes.getErrorAttributes(
        webRequest, ErrorAttributeOptions.of(ErrorAttributeOptions.Include.MESSAGE, ErrorAttributeOptions.Include.BINDING_ERRORS)
    ); //end attributes

    status = (Integer) attributes.get("status");
    message = (String) attributes.get("message");
    path = (String) attributes.get("path");
    error = new ApiResult(status, path, message);

    //attributesta error varsa
    if (attributes.containsKey("errors")) {
        List<FieldError> fieldErrorList = (List) attributes.get("errors");
        Map<String, String> validationMistake = new HashMap<>();
        for (FieldError fieldError : fieldErrorList) {
            validationMistake.put(fieldError.getField(), fieldError.getDefaultMessage());
        }
        error.setValidationErrors(validationMistake);
    }
    return error;
} //end handleError
}

```

HamitMizrakException

```
package com.hamitmizrak.exception;  
  
// My special Exception  
public class HamitMizrakException extends RuntimeException{  
    public HamitMizrakException(String message) {  
        super(message);  
    }  
}
```

ResourceAuthorizedException

```
package com.hamitmizrak.exception;

import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ResponseStatus;

// api yakalanması için
// 401 : yetkisiz giriş exception
@ResponseStatus(value = HttpStatus.UNAUTHORIZED)
public class ResourceAuthorizedException extends RuntimeException {
    public ResourceAuthorizedException(String message) {
        super(message);
    }
}
```

ResourceBadRequestException

```
package com.hamitmizrak.exception;

import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ResponseStatus;

// 400: bad request
@ResponseStatus(value = HttpStatus.BAD_REQUEST)
public class ResourceBadRequestException extends RuntimeException {
    public ResourceBadRequestException(String message) {
        super(message);
    }
}
```

ResourceCreatedException

```
package com.hamitmizrak.exception;

import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ResponseStatus;

// 201: oluşturuldu
@ResponseStatus(value = HttpStatus.CREATED)
public class ResourceCreatedException extends RuntimeException {
    public ResourceCreatedException(String message) {
        super(message);
    }
}
```


ResourceNotFoundException

```
package com.hamitmizrak.exception;

import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ResponseStatus;

// 404
@ResponseStatus(value = HttpStatus.NOT_FOUND)
public class ResourceNotFoundException extends RuntimeException{
    public ResourceNotFoundException(String message) {
        super(message);
    }
}
```

Genel İstisna Yakala

```
package com.hamitmizrak.handlecustomise;

import com.hamitmizrak.error.ApiResult;
import com.hamitmizrak.exception.*;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.RestControllerAdvice;

// Exception Spring tarafından yakalanması için kullanıyoruz.
@RestControllerAdvice
public class RestGlobalHandlingException {
```

```

//application.properties veri al
@Value("${api.result.error}")
private String errorApplicationProperties;

@Value("${api.result.message}")
private String messageApplicationProperties;
// HamitMizrakException
// KONTROL ==> http://localhost:2222/customer/api/v1/find
@ExceptionHandler({HamitMizrakException.class})
public ApiResult adviceHamitMizrakException() {
    ApiResult apiResult = ApiResult.builder()
        .error(errorApplicationProperties)
        .message(messageApplicationProperties)
        .path("/api")
        .status(44)
        .build();
    return apiResult;
}

```

404

```
// status codes: 404
// ResourceNotFoundException
@ExceptionHandler({ResourceNotFoundException.class})
public ApiResult adviceResourceNotFoundException() {
    ApiResult apiResult = ApiResult.builder()
        .error("Bulunamadı")
        .message("404 Hatası")
        .path("/api")
        .status(404)
        .build();
    return apiResult;
}
```

```
// status codes: 401
@ExceptionHandler({ResourceAuthorizedException.class})
public ApiResult adviceResourceAuthorizedException() {
    ApiResult apiResult = ApiResult.builder()
        .error("Yetkisiz Giriş")
        .message("401 Hatası")
        .path("/api")
        .status(401)
        .build();
    return apiResult;
}
```

```
// status codes: 400
@ExceptionHandler({ResourceBadRequestException.class})
public ApiResult adviceResourceBadRequestException() {
    ApiResult apiResult = ApiResult.builder()
        .error("Kötü istek")
        .message("400 Hatası")
        .path("/api")
        .status(400)
        .build();
    return apiResult;
}
```

```
// status codes: 201
@ExceptionHandler({ResourceCreatedException.class})
public ApiResult adviceResourceCreatedException() {
    ApiResult apiResult = ApiResult.builder()
        .error("Oluşturuldu")
        .message("201")
        .path("/api")
        .status(201)
        .build();
    return apiResult;
}
```

L18N

```
package com.hamitmizrak.bean;

import jakarta.servlet.http.HttpServletRequest;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.util.StringUtils;
import org.springframework.web.servlet.i18n.AcceptHeaderLocaleResolver;
import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;

import java.util.Arrays;
import java.util.List;
import java.util.Locale;

@Configuration
public class LocaleI18NBean extends AcceptHeaderLocaleResolver {
    // Default Accept-Language: tr YAPTIM
    @Bean
    public LocaleChangeInterceptor localeChangeInterceptorBean() {
        LocaleChangeInterceptor interceptor = new LocaleChangeInterceptor();
        interceptor.setParamName("tr");
        return interceptor;
    }

    List<Locale> localeList = Arrays.asList(new Locale("tr"), new Locale("en"), new Locale("ge"));

    @Override
    public Locale resolveLocale(HttpServletRequest request) {
        List<Locale.LanguageRange> list = Locale.LanguageRange.parse(request.getHeader("Accept-Language"));
        //org.springframework.util.StringUtils;
        if (StringUtils.isEmpty(request.getHeader("Accept-Language"))) {
            return Locale.getDefault();
        }
        Locale locale = Locale.lookup(list, localeList);
        return locale;
    }
}
```


ModelMapper

```
package com.hamitmizrak.bean;

import org.modelmapper.ModelMapper;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class ModelMapperBean {

    @Bean
    public ModelMapper modelMapperMethod(){
        return new ModelMapper();
    }
}
```

OpenApiMethod

```
package com.hamitmizrak.bean;

import io.swagger.v3.oas.models.OpenAPI;
import io.swagger.v3.oas.models.info.Contact;
import io.swagger.v3.oas.models.info.Info;
import io.swagger.v3.oas.models.info.License;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class OpenApiConfigurationBean {

    //localhost:4444/swagger-ui.html
    //swagger: API Document
    @Bean
    public OpenAPI openAPIMethod(){
        return new OpenAPI().info( new Info()
            .title("title bilgisi")
            .description("description bilgisi")
            .version("V1.0")
            .contact(new Contact()
                .name("Hamit")
                .url("www.hamitmizrak.com.tr")
                .email("hamitmizrak@gmail.com"))
            .termsOfService(" Software INC.")
            .license(new License()
                .name("Licence ")
                .url("www."))
        );
    }
}
```

PasswordEncoderBean

```
package com.hamitmizrak.bean;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

@Configuration
public class PasswordEncoderBean {

    @Bean
    public PasswordEncoder passwordEncoderMethod(){
        return new BCryptPasswordEncoder();
    }
}
```


CustomerDto → 1

```
package com.hamitmizrak.business.dto;

import jakarta.validation.constraints.*;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.extern.log4j.Log4j2;
import java.io.Serializable;
import java.util.Date;

// LOMBOK
@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
@Log4j2

// CustomerDto
// Validation
public class CustomerDto implements Serializable {
```

CustomerDto ➡ 2

```
// CustomerDto
// Validation
public class CustomerDto implements Serializable {
    //Serileştirme
    public static final Long serialVersionUID = 1L;

    //ID
    private Long id;

    //NAME
    @NotEmpty(message = "{customer.name.validation.constraints.NotNull.message}")
    private String name;

    //SURNAME
    @NotEmpty(message = "{customer.surname.validation.constraints.NotNull.message}")
    private String surname;
```

CustomerDto → 3

```
//EMAIL
@NotEmpty(message = "{customer.email.validation.constraints.NotNull.message}")
>Email(message = "{customer.email.validation.regex.constraints.NotNull.message}")
// Anotation
private String email;

//PASSWORD
//@Min(value = 7,message = "Şifreyi 7 küçük giremezsiniz")
//@Max(value = 7,message = "Şifreyi 10 büyük giremezsiniz")
@NotEmpty(message = "{customer.password.validation.constraints.NotNull.message}")
@Size(min = 7, max = 10, message = "{customer.password.pattern.validation.constraints.NotNull.message}")
@Pattern(regexp = "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&()-[{]}:;',?/*~$^+=<>]).*$", message = "{customer.password.pattern.validation.constraints.NotNull.message}")
private String password;

private String image;
private Object specialObject;

//DATE
private Date createDate;
} // end class
```

AuditorAwareImpl

```
package com.hamitmizrak.audit;

import org.springframework.data.domain.AuditorAware;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.context.SecurityContextHolder;

import java.util.Optional;

public class AuditorAwareImpl implements AuditorAware<String> {
    @Override
    public Optional<String> getCurrentAuditor() {
        //sistemdeki kullanıcı ismini Session ile almalısın
        //sayfaya giriş yapmış userlar
        Authentication authentication= SecurityContextHolder.getContext().getAuthentication();
        if(authentication!=null && authentication.isAuthenticated()){
            System.out.println(authentication.getName());
            System.out.println(authentication.getPrincipal());
            return Optional.of(authentication.getName());
        }
        return Optional.of("HamitM44.");
    }
}
```


AuditorAwareBean

```
package com.hamitmizrak.audit;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.domain.AuditorAware;

//AuditorAware: Sistemdeki kullanıcı loglama yapacak
@Configuration
public class AuditorAwareBean {

    @Bean
    public AuditorAware<String> auditorAwareMethod(){
        return new AuditorAwareImpl();
    }
}
```

BaseEntity → 1

```
package com.hamitmizrak.data.entity;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.hibernate.annotations.CreationTimestamp;
import org.springframework.data.annotation.CreatedBy;
import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedBy;
import org.springframework.data.annotation.LastModifiedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;
import java.util.Date;

//lombok
@Data
@NoArgsConstructor
@AllArgsConstructor

//audit
@EntityListeners(AuditingEntityListener.class)

//Json bunları parse etmesin => created_date,update_date
@JsonIgnoreProperties(value = {"created_date,update_date"},allowGetters = true)

//Super Class
@MappedSuperclass
// eğer class abstract olursa interface ==> implements Method zorunluluktan kaldırır.
abstract public class BaseEntity {
```

BaseEntity → 2

```
abstract public class BaseEntity {  
  
    //ID  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id",updatable = false,nullable = false)  
    private Long id;  
  
    //Tarihi otomatik sisteme ekleme  
    @Column(name = "system_created_date",updatable = false)  
    @Temporal(TemporalType.TIMESTAMP)  
    @CreationTimestamp  
    private Date systemCreateDate;
```

BaseEntity → 3

```
////// LOGLAMA ////////////////////////////////////////  
// AUDITING  
// Kim ekledi ?  
@Column(name = "created_by")  
@CreatedBy  
private String createdBy;  
  
// Kim ne zaman ekledi ?  
@Column(name = "created_date")  
@CreatedDate  
private Date createDate;  
  
// Kim güncelledi ?  
@Column(name = "update_by")  
@LastModifiedBy  
private String updateBy;  
  
// Kim ne zaman güncelledi ?  
@Column(name = "update_date")  
@LastModifiedDate  
private Date updateDate;  
}
```

CustomerEntity → 1

package com.hamitmizrak.data.entity;

```
import jakarta.persistence.*;
import jakarta.validation.constraints.Email;
import jakarta.validation.constraints.NotEmpty;
import jakarta.validation.constraints.Size;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.extern.log4j.Log4j2;
import java.io.Serializable;
import java.util.Date;
```

```
// LOMBOK
```

```
@Data
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
@Builder
```

```
@Log4j2
```

```
@Entity
```

```
@Table(name = "customer")
```

```
public class CustomerEntity extends BaseEntity implements Serializable {
```

CustomerEntity ➡ 2

```
// LOMBOK
@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
@Log4j2

@Entity
@Table(name = "customer")
public class CustomerEntity extends BaseEntity implements Serializable {
    // Serileştirme
    public static final Long serialVersionUID = 1L;

    // NAME
    private String name;

    // SURNAME
    private String surname;

    // PASSWORD
    private String password;
```

CustomerEntity → 3

```
// LOB : büyük veriler
@Lob
@Column( columnDefinition = "varchar(255) default 'picture.png'")
private String image;

// Javada olsun ancak Database olmasın
@Transient
private Object specialObject;

// EMAIL
// virgüllü sayı: columnDefinition = "Decimal(10,2) default '100.00'")
@Column(
    name = "email",
    nullable = true,
    unique = true,
    length = 100,
    insertable = true,
    updatable = true,
    columnDefinition = "varchar(255) default 'email yazmadiniz'")
private String email;
}
```

@Repository

```
package com.hamitmizrak.data.repository;

import com.hamitmizrak.data.entity.CustomerEntity;
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface ICustomerRepository extends CrudRepository<CustomerEntity, Long> {

    //Delivered Query
    CustomerEntity findByEmail(String email);
}
```


CustomerUniqueEmail

```
package com.hamitmizrak.annotation;

import jakarta.validation.Constraint;
import jakarta.validation.Payload;
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

@Target({ElementType.FIELD})
@Retention(RetentionPolicy.RUNTIME)
@Constraint(validatedBy = {UniqueEmailValidation.class})
// My special Annotation
public @interface CustomerUniqueEmail {
    String message() default "{customer.email.validation.unique.constraints.NotNull.message}";
    Class<?>[] groups() default {};
    Class<? extends Payload>[] payload() default {};
}
```

UniqueEmailValidation

```
package com.hamitmizrak.annotation;

import com.hamitmizrak.data.entity.CustomerEntity;
import com.hamitmizrak.data.repository.ICustomerRepository;
import jakarta.validation.ConstraintValidator;
import jakarta.validation.ConstraintValidatorContext;
import lombok.RequiredArgsConstructor;

// LOMBOK
@RequiredArgsConstructor
public class UniqueEmailValidation implements ConstraintValidator<CustomerUniqueEmail, String> {

    //1.YOL => Field Injection
    //@Autowired
    //IUserRegisterRepository iUserRegisterRepository;

    //2.YOL ==> Constructor Injection
    /* public UniqueEmailValidation(IUserRegisterRepository iUserRegisterRepository) {
        this.iUserRegisterRepository = iUserRegisterRepository;
    }*/

    //3.YOL ==> Lombok Injection
    private final ICustomerRepository iCustomerRepository;

    //Email adresinde database böyle bir email var mı yok mu
    public boolean isValid(String email, ConstraintValidatorContext context) {
        CustomerEntity customerEntity = iCustomerRepository.findByEmail(email);
        //eğer bu email database varsa return false
        if (customerEntity != null){
            return false;
        }
        return true;
    }
}
```

CustomerDto → 1

```
package com.hamitmizrak.business.dto;

import com.hamitmizrak.annotation.CustomerUniqueEmail;
import jakarta.validation.constraints.*;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.extern.log4j.Log4j2;
import java.io.Serializable;
import java.util.Date;

// LOMBOK
@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
@Log4j2

// CustomerDto
// Validation
public class CustomerDto implements Serializable {
```

CustomerDto → 2

```
// CustomerDto
// Validation
public class CustomerDto implements Serializable {
    //Serileştirme
    public static final Long serialVersionUID = 1L;

    //ID
    private Long id;

    //NAME
    @NotEmpty(message = "{customer.name.validation.constraints.NotNull.message}")
    private String name;

    //SURNAME
    @NotEmpty(message = "{customer.surname.validation.constraints.NotNull.message}")
    private String surname;

    //EMAIL
    @NotEmpty(message = "{customer.email.validation.constraints.NotNull.message}")
    @Email(message = "{customer.email.validation.regex.constraints.NotNull.message}")
    @CustomerUniqueEmail
    private String email;
```

CustomerDto ➔ 3

```
//PASSWORD
//@Min(value = 7,message = "Şifreyi 7 küçük giremezsiniz")
//@Max(value = 7,message = "Şifreyi 10 büyük giremezsiniz")
@NotEmpty(message = "{customer.password.validation.constraints.NotNull.message}")
@Size(min = 7, max = 10, message = "{customer.password.pattern.validation.constraints.NotNull.message}")
@Pattern(regexp = "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&()-[{]}:;','/*~$^+=<>]).*$", message = "{customer.password.pattern.validation.constraints.NotNull.message}")
private String password;

private String image;
private Object specialObject;

//DATE
private Date systemDate;

//Audit
// Kim ekledi ?
private String createdBy;

// Kim ne zaman ekledi ?
private Date createdAt;

// Kim güncelledi ?
private String updateBy;

// Kim ne zaman güncelledi ?
private Date updateDate;

} // end class
```

CustomerEntity ➔ 1

```
package com.hamitmizrak.data.entity;

import jakarta.persistence.*;
import jakarta.validation.constraints.Email;
import jakarta.validation.constraints.NotEmpty;
import jakarta.validation.constraints.Size;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.extern.log4j.Log4j2;
import java.io.Serializable;
import java.util.Date;

// LOMBOK
@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
@Log4j2

@Entity
@Table(name = "customer")
public class CustomerEntity extends BaseEntity implements Serializable {
```

CustomerEntity → 2

```
@Entity
@Table(name = "customer")
public class CustomerEntity extends BaseEntity implements Serializable {
    // Serileştirme
    public static final Long serialVersionUID = 1L;

    // NAME
    private String name;

    // SURNAME
    private String surname;

    // PASSWORD
    private String password;

    // LOB : büyük veriler
    @Lob
    @Column(columnDefinition = "varchar(255) default 'picture.png'")
    private String image;

    // Javada olsun ancak Database olmasın
    @Transient
    private Object specialObject;
```

CustomerEntity → 3

```
// EMAIL
// virgüllü sayı: columnDefinition = "Decimal(10,2) default '100.00'"
@Column(
    name = "email",
    nullable = true,
    //unique = true, // bunun yerine kendi Annotation yazdım
    length = 100,
    insertable = true,
    updatable = true,
    columnDefinition = "varchar(255) default 'email yazmadiniz'"
)
private String email;
}
```


ICustomerServices → 1

```
package com.hamitmizrak.business.services;
```

```
import com.hamitmizrak.business.dto.CustomerDto;
```

```
import com.hamitmizrak.data.entity.CustomerEntity;
```

```
import java.util.List;
```

```
import java.util.Map;
```

```
public interface ICustomerServices {
```

```
    // SPEED DATA
```

```
    public CustomerDto speedData();
```

```
    // DELETE ALL
```

```
    public String customerAllDelete();
```

ICustomerServices → 2

```
public interface ICustomerServices {  
  
    // MODEL MAPPER  
    public CustomerDto EntityToDto(CustomerEntity customerEntity );  
    public CustomerEntity DtoToEntity(CustomerDto customerDto);  
  
    // CREATE  
    public CustomerDto createRegister(CustomerDto customerDto);  
  
    // LIST  
    public List<CustomerDto> getAllCustomers();  
  
    // FIND ID  
    public CustomerDto getFindByCustomerId(Long id);  
  
    // DELETE  
    public Map<String, Boolean> deleteCustomer(Long id);  
  
    // UPDATE  
    public CustomerDto updateCustomer(Long id, CustomerDto customerDto);  
}
```

CustomerServices → 1

```
package com.hamitmizrak.business.services.impl;

import com.hamitmizrak.bean.ModelMapperBean;
import com.hamitmizrak.bean.PasswordEncoderBean;
import com.hamitmizrak.business.dto.CustomerDto;
import com.hamitmizrak.business.services.ICustomerServices;
import com.hamitmizrak.data.entity.CustomerEntity;
import com.hamitmizrak.data.repository.ICustomerRepository;
import com.hamitmizrak.exception.HamitMizrakException;
import com.hamitmizrak.exception.ResourceNotFoundException;
import lombok.RequiredArgsConstructor;
import lombok.SneakyThrows;
import lombok.extern.log4j.Log4j2;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

CustomerServices ➔ 2

```
// LOMBOK
@RequiredArgsConstructor // injection
@Log4j2
//@SneakyThrows //throws

@Service
// Asıl iş yükünü yapan class
public class CustomerServices implements ICustomerServices {

    // INJECTION
    private final ICustomerRepository iCustomerRepository;
    private final ModelMapperBean modelMapperBean;
    private final PasswordEncoderBean passwordEncoderBean;
```

Speed Data

```
// SPEED DATA
@Override
public CustomerDto speedData() {
    CustomerDto customerDto=null;
    for (int i = 1; i <= 5; i++) {
        customerDto = CustomerDto.builder()
            .name("adı " + i)
            .surname("soyadı " + i)
            .email(UUID.randomUUID().toString().concat("@gmail.com"))
            .password("root")
            .build();
        customerDto.setPassword(passwordEncoderBean.passwordEncoderMethod().encode(customerDto.getPassword()));
        // DTO ==> ENTITY Model Mapper
        CustomerEntity customerEntityMapper = DtoToEntity(customerDto);
        CustomerEntity customerEntityRepository = iCustomerRepository.save(customerEntityMapper);
    }
    return customerDto;
}
```

DeleteAll

```
// DELETE ALL
@Override
public String customerAllDelete() {
    iCustomerRepository.deleteAll();
    return "Bütün veriler silindi";
}
```

CustomerServices ➔ 3

```
//MODEL MAPPER
@Override
public CustomerDto EntityToDto(CustomerEntity customerEntity) {
    return modelMapperBean.modelMapperMethod().map(customerEntity, CustomerDto.class);
}

@Override
public CustomerEntity DtoToEntity(CustomerDto customerDto) {
    return modelMapperBean.modelMapperMethod().map(customerDto, CustomerEntity.class);
}
```


CustomerServices ➔ 4

```
// CREATE
@Override
@Transactional // create,delete,update
public CustomerDto createRegister(CustomerDto customerDto) {
    if (customerDto != null) {
        // Password Masking
        customerDto.setPassword(passwordEncoderBean.passwordEncoderMethod().encode(customerDto.getPassword()));
        // DTO ==> ENTITY Model Mapper
        CustomerEntity customerEntityMapper = DtoToEntity(customerDto);
        CustomerEntity customerEntityRepository = iCustomerRepository.save(customerEntityMapper);
        // Kaydedilen verinin ID dönsün
        customerDto.setId(customerEntityRepository.getId());
        customerDto.setCreateDate(customerEntityRepository.getCreateDate());
    } else if (customerDto == null) {
        throw new HamitMizrakException("Customer Dto Null geldi");
    }
    return customerDto;
}
```

CustomerServices → 5

```
// LIST
@Override
public List<CustomerDto> getAllCustomers() {
    Iterable<CustomerEntity> customerEntityList = iCustomerRepository.findAll();
    // EntityList ==> DtoList çevir
    List<CustomerDto> entityToDtoList = new ArrayList<>();
    for (CustomerEntity temp : customerEntityList) {
        CustomerDto customerDto = EntityToDto(temp);
        entityToDtoList.add(customerDto);
    }
    return entityToDtoList;
}
```

CustomerServices → 6

```
// FIND
@Override
public CustomerDto getFindByCustomerId(Long id) {
    CustomerEntity customerEntity=null;
    if(id!=null){
        customerEntity = iCustomerRepository.findById(id)
            .orElseThrow(() -> new ResourceNotFoundException(id + " nolu ID yoktur"));
    }else if (id == null) {
        throw new HamitMizrakException("Customer Dto Null geldi");
    }
    return EntityToDto(customerEntity);
}
```

CustomerServices ➡ 7

```
// DELETE
@Override
@Transactional // create,delete,update
public Map<String, Boolean> deleteCustomer(Long id) {
    Map<String, Boolean> mapDelete = new HashMap<>();
    // FIND
    CustomerDto customerDto = getFindByCustomerId(id);
    // Dto ==> Entity
    CustomerEntity customerEntity = DtoToEntity(customerDto);
    if (customerEntity != null) {
        iCustomerRepository.delete(customerEntity);
        mapDelete.put("Silindi", Boolean.TRUE);
    }
    return mapDelete;
}
```

CustomerServices → 8

```
// UPDATE
@Override
@Transactional // create,delete,update
public CustomerDto updateCustomer(Long id, CustomerDto customerDto) {
    // FIND
    CustomerDto customerDtoFind = getFindByCustomerId(id);
    // Dto ==> Entity
    CustomerEntity customerEntity = DtoToEntity(customerDtoFind);
    if (customerEntity != null) {
        customerEntity.setName(customerDto.getName());
        customerEntity.setSurname(customerDto.getSurname());
        customerEntity.setEmail(customerDto.getEmail());
        customerEntity.setImage(customerDto.getImage());
        customerEntity.setPassword(passwordEncoderBean.passwordEncoderMethod().encode(customerDto.getPassword()));
        iCustomerRepository.save(customerEntity);
        // Kaydedilen verinin ID dönsün
        customerDto.setId(customerEntity.getId());
        customerDto.setCreatedDate(customerEntity.getCreatedDate());
    }
    return customerDto;
}
} //end class
```

ICustomerApi → 1

```
package com.hamitmizrak.controller.api;

import com.hamitmizrak.business.dto.CustomerDto;
import org.springframework.http.ResponseEntity;
import java.util.List;
import java.util.Map;

public interface ICustomerApi {

    //SPEED DATA
    public ResponseEntity<CustomerDto> speedData();

    // DELETE ALL
    public ResponseEntity<String> customerAllDelete();
}
```

ICustomerApi → 2

```
// CREATE
public ResponseEntity<CustomerDto> createRegister(CustomerDto customerDto);

// LIST
public ResponseEntity<List<CustomerDto>> getAllCustomers();

// FIND ID
public ResponseEntity<?> getFindByCustomerId(Long id);

// DELETE
public ResponseEntity<Map<String, Boolean>> deleteCustomer(Long id);

// UPDATE
public ResponseEntity<CustomerDto> updateCustomer(Long id, CustomerDto customerDto);
}
```

CustomerApi → 1

```
package com.hamitmizrak.controller.api;

import com.hamitmizrak.business.dto.CustomerDto;
import com.hamitmizrak.business.services.impl.CustomerServices;
import com.hamitmizrak.controller.ICustomerApi;
import com.hamitmizrak.error.ApiResult;
import jakarta.annotation.PostConstruct;
import jakarta.validation.Valid;
import lombok.RequiredArgsConstructor;
import lombok.extern.log4j.Log4j2;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import java.util.List;
import java.util.Map;
```


CustomerApi → 2

```
// LOMBOK
@Log4j2
@RequiredArgsConstructor // Injection

// Api
@RestController
@CrossOrigin(origins = "http://localhost:3000")
// CORS: eğer reactta package.json'da proxy yazarsam @CrossOrigin yazmasamda olur
@RequestMapping("/customer/api/v1")
public class CustomerApi implements ICustomerApi {

    // ERROR
    private ApiResult apiResult;

    // @PostConstruct
    @PostConstruct
    public void springData(){
        apiResult=new ApiResult();
    }
}
```

CustomerApi → 3

```
//Injection
//1.YOL => Field Injection
/*
@Autowired
private CustomerServices customerServices;
*/

//2.YOL => Constructor Injection
/*
private final CustomerServices customerServices;
@Autowired
public CustomerApi(CustomerServices customerServices) {
    this.customerServices = customerServices;
}
*/

//3.YOL => Constructor Injection (LOMBOK)
private final CustomerServices customerServices;
```

Speed All

```
// SPEED ALL
// http://localhost:2222/customer/api/v1/speedData
@Override
@GetMapping("/speedData")
public ResponseEntity<CustomerDto> speedData() {
    return ResponseEntity.ok(customerServices.speedData());
}
```

Delete All

```
// DELETE ALL
// http://localhost:2222/customer/api/v1/customerAllData
@Override
@GetMapping("/customerAllData")
public ResponseEntity<String> customerAllDelete() {
    return ResponseEntity.ok(customerServices.customerAllDelete());
}
```

CustomerApi ➡ 4

```
// CREATE
// http://localhost:2222/customer/api/v1/create
@Override
@PostMapping(value="create")
public ResponseEntity<CustomerDto> createRegister( @Valid @RequestBody CustomerDto customerDto) {
    // return new ResponseEntity<>(adminDto, HttpStatus.OK);
    // return ResponseEntity.status(HttpStatus.OK).body(adminDto);
    // return ResponseEntity.status(200).body(adminDto);
    // return ResponseEntity.ok().body(adminDto);
    return ResponseEntity.ok(customerServices.createRegister(customerDto));
}
```

CustomerApi → 5

```
// LIST
// http://localhost:2222/customer/api/v1/list
@Override
@GetMapping("list")
public ResponseEntity<List<CustomerDto>> getAllCustomers() {
    return ResponseEntity.ok(customerServices.getAllCustomers());
}
```

CustomerApi → 6

```
// FIND
// http://localhost:2222/customer/api/v1/find
// http://localhost:2222/customer/api/v1/find/0
// http://localhost:2222/customer/api/v1/find/-1
// http://localhost:2222/customer/api/v1/find/1

@Override
@GetMapping({"/find", "/find/{id}"})
public ResponseEntity<?> getFindByCustomerId(@PathVariable(name = "id", required = false) Long id) {
    if (id == null) {
        log.error("API => 404 NOT FOUND");
        //return ResponseEntity.notFound().build();
    } else if (id == 0) {
        log.error("API => 400 BAD REQUEST");
        apiResult = new ApiResult(400, "localhost:2222/customer/api/v1/register/0", "Kötü istek", "Bad Request");
        //return ResponseEntity.badRequest().build();
        return ResponseEntity.badRequest().body(apiResult);
    } else if (id < 0) {
        log.error("API => 401 UNAUTHROZED");
        apiResult = ApiResult.builder()
            .error("unAuthorized")
            .message("Yetkisiz Giriş")
            .path("localhost:2222/customer/api/v1/register/-1")
            .status(401)
            .build();
        return ResponseEntity.status(401).body(apiResult);
    }
    System.out.println(customerServices.getFindByCustomerId(id));
    return ResponseEntity.ok(customerServices.getFindByCustomerId(id));
}
```

CustomerApi ➔ 7

```
// DELETE
// http://localhost:2222/customer/api/v1/delete/1
@Override
@DeleteMapping({"delete","delete/{id}"})
public ResponseEntity<Map<String, Boolean>> deleteCustomer(@PathVariable(name = "id", required = false) Long id) {
    return ResponseEntity.ok(customerServices.deleteCustomer(id));
}
```


CustomerApi ➔ 8

```
// UPDATE
// http://localhost:2222/customer/api/v1/update/1
@Override
@PutMapping({"/update","/update/{id}"})
public ResponseEntity<CustomerDto> updateCustomer(@PathVariable(name = "id", required = false) Long id, @Valid @RequestBody CustomerDto customerDto) {
    return ResponseEntity.ok(customerServices.updateCustomer(id, customerDto));
}
} // end update
```

@SpringBootApplication ➔ 1

```
package com.hamitmizrak;

import jakarta.annotation.PostConstruct;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.domain.EntityScan;
import org.springframework.boot.autoconfigure.security.servlet.SecurityAutoConfiguration;
import org.springframework.data.jpa.repository.config.EnableJpaAuditing;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import java.util.Collections;
import java.util.TimeZone;

// auditorAware için
@EnableJpaAuditing(auditorAwareRef = "auditorAwareMethod")

//SCAN
@EntityScan(basePackages = "com.hamitmizrak.data.entity") //Entity bulamadığı zaman
@EnableJpaRepositories(basePackages = "com.hamitmizrak.data.repository" ) //Repository bulamadığı zaman

// Spring Security

@SpringBootApplication(exclude = {
    //SecurityAutoConfiguration.class,
    org.springframework.boot.autoconfigure.security.servlet.SecurityAutoConfiguration.class,
    org.springframework.boot.actuate.autoconfigure.security.servlet.ManagementWebSecurityAutoConfiguration.class
})

//@SpringBootApplication
public class TurgutUniversitySpringAllInOneApplication {
```

@SpringBootApplication → 2

```
//@SpringBootApplication
public class TurgutUniversitySpringAllInOneApplication {

    @PostConstruct
    public void init(){
        TimeZone.setDefault(TimeZone.getTimeZone("IST"));
    }

    public static void main(String[] args) {
        // devtools active isActive
        // System.setProperty("spring.devtools.restart.enabled", "true");

        // PORT Ayarlamak
        /*
        SpringApplication app = new SpringApplication(TurgutUniversitySpringAllInOneApplication.class);
        app.setDefaultProperties(Collections
            .singletonMap("server.port", "8083"));
        app.run(args);
        */

        //Disables headless JOptionPane
        System.setProperty("java.awt.headless", "false");

        SpringApplication.run(TurgutUniversitySpringAllInOneApplication.class, args);
    } //end PSVM
} // end TurgutUniversitySpringAllInOneApplication
```

CommandLineRunner

```
package com.hamitmizrak.bean;

import com.hamitmizrak.business.services.ICustomerServices;
import lombok.RequiredArgsConstructor;
import lombok.extern.log4j.Log4j2;
import org.springframework.boot.CommandLineRunner;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

// LOMBOK
@RequiredArgsConstructor // Injection
@Log4j2

@Configuration
public class CustomerCommandLineRunner {

    private final ICustomerServices customerServices;

    // proje ayağa kalkar kalmaz otomatik veri eklesin
    @Bean
    public CommandLineRunner projectDataProcess(){ // parametre olarak verebilirsin==> ICustomerServices customerServices
        return args -> {
            customerServices.speedData();
        };
    }
}
```

Create

```
package com.hamitmizrak.persist;

public class PostmanDocument {
    // CREATE
    // http://localhost:2222/customer/api/v1/create
    /*
    POST /customer/api/v1/create HTTP/1.1
    Host: localhost:2222
    Content-Type: application/json
    Content-Length: 121

    {
      "name": "Hamit",
      "surname": "Mizrak",
      "email": "hamitmizrak45444@gmail.com",
      "password": "Hm123456@"
    }
    */
}
```

List

```
// LIST
//http://localhost:2222/customer/api/v1/list
/*
  GET /customer/api/v1/list HTTP/1.1
  Host: localhost:2222
*/
```

Find

```
//FIND
// http://localhost:2222/customer/api/v1/find/-1
/*

GET /customer/api/v1/find/-1 HTTP/1.1
Host: localhost:2222
*/
```

Update

```
//UPDATE
//http://localhost:2222/customer/api/v1/update/1
/*
PUT /customer/api/v1/update/3HTTP/1.1
Host:localhost:2222
Content-Type:application/json
Content-Length:125

{
  "name":"Hamit22",
    "surname":"Mizrak22",
    "email":"hamitmizrak44522@gmail.com",
    "password":"Hm123456@"
}
*/
```


Delete

```
//DELETE
// http://localhost:2222/customer/api/v1/delete/1
/*
DELETE /customer/api/v1/delete/1 HTTP/1.1
Host: localhost:2222
*/
```

```
}
```


ICustomerMvc

```
package com.hamitmizrak.controller.mvc;
import com.hamitmizrak.business.dto.CustomerDto;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;

// for Spring MVC
public interface ICustomerMvc {

    // SPEEAD DATA => GET
    // http://localhost:2222/customer/speedData
    String customerSpeedDataGet();

    // ALL DELETE => GET
    // http://localhost:2222/customer/customerAllData
    String customerAllDeleteGet();
}
```

```
// CREATE => GET
// http://localhost:2222/customer/create
String customerCreateGet(Model model);

// CREATE => POST
String customerCreatePost( CustomerDto customerDto, BindingResult bindingResult, Model model);

// LIST
// http://localhost:2222/customer/list
String customerListGet(Model model);

// FIND
// http://localhost:2222/customer/find/1
String customerFindGet( Long id, Model model);
```

```
// DELETE
// http://localhost:2222/customer/delete/1
String customerDeleteGet( Long id, Model model);

// UPDATE
// http://localhost:2222/customer/update/1
String customerUpdateGet( Long id, Model model);

// UPDATE
// http://localhost:2222/customer/update/1
String customerUpdatePost( Long id, CustomerDto customerDto, BindingResult bindingResult, Model model);
}
```

CustomerMVCImpl → 1

```
package com.hamitmizrak.controller.mvc;

import com.hamitmizrak.business.dto.CustomerDto;
import com.hamitmizrak.business.services.ICustomerServices;
import jakarta.validation.Valid;
import lombok.RequiredArgsConstructor;
import lombok.extern.log4j.Log4j2;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Map;

// LOMBOK
@RequiredArgsConstructor // Injection
@Log4j2

// Spring MVC
@Controller
@RequestMapping("/customer")
public class CustomerMvcImpl implements ICustomerMvc {
```

CustomerMVCImpl → 2

```
// LOMBOK
@RequiredArgsConstructor // Injection
@Log4j2

// Spring MVC
@Controller
@RequestMapping("/customer")
public class CustomerMvcImpl implements ICustomerMvc {

    // List, Create, Delete => gelen modelAddAttributes
    private String modelAttributesTemp=null;

    // Injection
    private final ICustomerServices customerServices;
```

CustomerMVCImpl → 3

```
// SPEED DATA => GET
// http://localhost:2222/customer/speedData
@Override
@GetMapping("/speedData")
public String customerSpeedDataGet() {
    customerServices.speedData();
    modelAttributesTemp="5 tane veri eklendi";
    return "redirect:/customer/list";
}
```


CustomerMVCImpl ➔ 4 (DeleteAll)

```
// ALL DELETE => GET
// http://localhost:2222/customer/customerAllData
@Override
@GetMapping("/customerAllData")
public String customerAllDeleteGet() {
    customerServices.customerAllDelete();
    modelAttributesTemp="Bütün Veriler Silindi";
    return "redirect:/customer/list";
}
```

CustomerMVCImpl ➔ 5 (Create)

```
// CREATE => GET
// http://localhost:2222/customer/create
@Override
@GetMapping("/create")
public String customerCreateGet(Model model) {
    model.addAttribute("customer_create", new CustomerDto());
    return "customer/create";
}

// CREATE => POST
@Override
@PostMapping("/create")
public String customerCreatePost(@Valid @ModelAttribute("customer_create") CustomerDto customerDto, BindingResult bindingResult, Model model) {
    if (bindingResult.hasErrors()) {
        log.error("HATA CREATED" + bindingResult);
        return "customer/create";
    }
    //eğer hata yoksa
    model.addAttribute("success_create", "Müşteri Eklendi" + customerDto);
    log.info(customerDto);
    // Services Create
    customerServices.createRegister(customerDto);
    modelAttributesTemp="Eklendi "+customerDto;
    return "redirect:/customer/list";
}
```

CustomerMVCImpl → 6 (list)

```
// LIST
// http://localhost:2222/customer/list
@Override
@GetMapping("/list")
public String customerListGet(Model model) {
    // Services List
    List<CustomerDto> list = customerServices.getAllCustomers();
    model.addAttribute("customer_list", list);
    list.forEach((temp) -> {
        System.out.println(temp);
    });
    model.addAttribute("model_list", modelAttributesTemp);
    return "customer/list";
}
```

CustomerMVCImpl → 7(Find)

```
// FIND
// http://localhost:2222/customer/find/1
@Override
@GetMapping("/find/{id}")
public String customerFindGet(@PathVariable(name = "id") Long id, Model model) {
    // Services List
    CustomerDto find = customerServices.getFindByCustomerId(id);
    model.addAttribute("customer_find", find);
    System.out.println(find);
    return "customer/detail";
}
```

CustomerMVCImpl ➔ 8 (Delete)

```
// DELETE
// http://localhost:2222/customer/delete/1
@Override
@GetMapping("/delete/{id}")
public String customerDeleteGet(@PathVariable(name = "id") Long id, Model model) {
    // Services List
    Map<String, Boolean> mapDelete = customerServices.deleteCustomer(id);
    model.addAttribute("customer_find", mapDelete);
    // Ekranda Göster
    modelAttributesTemp="Silindi";
    System.out.println(mapDelete);
    //return "redirect:/customer/list?sil=tamam";
    return "redirect:/customer/list";
}
```

CustomerMVCImpl ➔ 9 (Update -1)

```
// UPDATE
// http://localhost:2222/customer/update/1
@Override
@GetMapping("/update/{id}")
public String customerUpdateGet(@PathVariable(name = "id") Long id, Model model) {
    CustomerDto customerDto = customerServices.getFindByCustomerId(id);
    if (customerDto != null) {
        model.addAttribute("customer_update", customerDto);
    } else
        model.addAttribute("customer_update", id + " numaralı veri yoktur");
    // Ekranda Göster
    modelAttributesTemp=id + " numaralı veri yoktur";
    return "customer/update";
}
```

CustomerMVCImpl ➔ 10 (Update-2)

```
// UPDATE
// http://localhost:2222/customer/update/1
@Override
@PostMapping("/update/{id}")
public String customerUpdatePost(
    @PathVariable(name = "id") Long id,
    @Valid @ModelAttribute("customer_update") CustomerDto customerDto,
    BindingResult bindingResult,
    Model model) {
    if (bindingResult.hasErrors()) {
        log.error("HATA: " + bindingResult);
        return "customer/update";
    }
    customerServices.updateCustomer(id, customerDto);
    model.addAttribute("success_update", "Güncellendi");
    modelAttributesTemp=id + " Güncellendi "+customerDto;
    return "redirect:/customer/list";
}
}
```

list.html → 1

```
<!doctype html>
<html th:lang="{#locale.language}" xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Thymeleaf String</title>

  <!-- Turkce karakter -->
  <meta charset="utf-8">

  <!-- responsive design -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- seo -->
  <meta name="keywords" content="html5,css3,js,jquery,bootstrap,react,javase"/>
  <meta name="description" content=""/>
  <meta name="author" content=""/>

  <!-- font-awesome -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"/>

  <!-- google font -->
  <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:ital@1&display=swap" rel="stylesheet">

  <!-- Bootstrap CSS v5.0.2 -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

  <!-- jquery -->
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

  <!-- react libraries -->
  <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

  <link rel="stylesheet" th:href="@{/css/style.css}">
  <script th:src="@{/js/template.js}"></script>

  <!-- dataTables CSS: paging and sorting -->
  <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/v/bs4/dt-1.10.25/datatables.min.css"/>
</head>
```


list.html → 2

```
<body class="container">

<!-- http://localhost:2222/customer/list -->

<div th:if="${model_list}!=null" class="alert alert-primary alert-dismissible fade show mt-5" role="alert">
    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
    <strong th:text="${model_list}"></strong>
</div>

<!-- start codes -->
<h1 class="text-primary mt-5">LIST</h1>

<a th:href="@{/customer/create}" class="btn btn-primary">EKLE</a>
<a th:href="@{/customer/speedData}" class="btn btn-secondary">Hızlı Ekle</a>
<a th:href="@{/customer/customerAllData}" th:onclick="if(!confirm('Bütün Verileri silmek istiyor musunuz ?')) return false" class="btn btn-danger">
    <i class="fa-solid fa-trash"></i>
</a>
</td>
<br><br>
```

list.html → 3

```
<div th:if="${customer_list}!=null" class="container my-5">
  <table class="table table-hover table-striped" id="datalistregisterTable">
    <thead>
      <tr>
        <th class="text-uppercase" th:text="#{customer.id}"></th>
        <th class="text-uppercase" th:text="#{customer.name}"></th>
        <th class="text-uppercase" th:text="#{customer.surname}"></th>
        <th class="text-uppercase" th:text="#{customer.email}"></th>
        <th class="text-uppercase" th:text="#{customer.password}"></th>
        <th class="text-uppercase" th:text="#{customer.date}"></th>
        <th class="text-uppercase">Kim Ekledi</th>
        <th class="text-uppercase">Kim Günledi</th>
        <th class="text-uppercase" th:text="#{customer.update}"></th>
        <th class="text-uppercase" th:text="#{customer.show}"></th>
        <th class="text-uppercase" th:text="#{customer.delete}"></th>
      </tr>
    </thead>
```

list.html → 3

```
<tbody>
<tr th:if="{cust.id>0}" th:each="cust:{customer_list}">
  <td th:text="{cust.id}"></td>
  <td th:text="{cust.name}"></td>
  <td th:text="{cust.surname}"></td>
  <td th:text="{cust.email}"></td>
  <td th:text="{cust.password}"></td>
  <td th:text="{cust.systemDate}"></td>
  <td th:text="{cust.createdBy}"></td>
  <td th:text="{cust.updateBy}"></td>
  <td>
    <a class="btn btn-primary" th:href="@{/customer/update/{id}(id={cust.id})}">
      <i class="fa-solid fa-wrench"></i>
    </a>
  </td>
  <td>
    <a class="btn btn-success" th:href="@{/customer/find/{id}(id={cust.id})}">
      <i class="fa-solid fa-eye btn-sm"></i>
    </a>
  </td>
  <!--
  <td>
    <a th:href="@{/customer/delete/{id}(id={cust.id})}" th:onclick="if(!confirm('silme  istiyor musunuz ?')) return false" class="btn btn-danger">
      <i class="fa-solid fa-trash"></i>
    </a>
  </td>
  -->
  <td>
    <a class="btn btn-danger"
      th:onclick="deleteSweetAlertRegister('/customer/delete/' + {cust.id} + '\')">
      <i class="fa-solid fa-trash-can"></i>
    </a>
  </td>
</tr>
</tbody>
</table>
</div>
```

list.html → 4

```
<!-- dataTables JS: paging and sorting -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript" src="https://cdn.datatables.net/v/bs4/dt-1.10.25/datatables.min.js"></script>
<script>
    $(document).ready(function () {
        $('#datalistregisterTable').DataTable({
            'aoColumnDefs': [{
                'bSortable': false,
                'aTargets': [-1] /* 1st one, start by the right */
            }]
        });
    })
</script>
```

list.html → 5

```
<!-- ends codes -->
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"></script>

<!-- Sweet Alert -->
<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
<script type="text/javascript">
    function deleteSweetAlertRegister(url) {
        swal({
            title: "Silmek istediğinizden emin misiniz?",
            text: "Silindikten sonra tekrar geri alınamaz.",
            icon: "warning",
            buttons: ["İptal", "Sil"],
            dangerMode: true,
        })
        .then((willDelete) => {
            if (willDelete) {
                swal("Kayıt başarılı bir şekilde silindi.", {
                    icon: "success",
                });
                window.location.href = url;
            } else {
                swal("İptal edildi.", {
                    icon: "warning",
                });
            }
        });
    }
</script>
</body>
</html>
```

detail.html → 1

```
<!doctype html>
<html th:lang="{#locale.language}" xmlns:th="http://www.thymeleaf.org" >
<head>
  <title>Thymeleaf String</title>

  <!-- Turkce karakter -->
  <meta charset="utf-8">

  <!-- responsive design -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- seo -->
  <meta name="keywords" content="html5,css3,js,jquery,bootstrap,react,javase" />
  <meta name="description" content="" />
  <meta name="author" content="" />

  <!-- font-awesome -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css" />

  <!-- google font -->
  <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:ital@1&display=swap" rel="stylesheet">

  <!-- Bootstrap CSS v5.0.2 -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

  <!-- jquery -->
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

  <!-- react libraries -->
  <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

  <link rel="stylesheet" th:href="@{/css/style.css}">
  <script th:src="@{/js/template.js}"></script>
</head>
```

detail.html → 2

```
<body class="container bg-dark text-dark">

  <!-- start codes -->
  <a th:href="@{/customer/list}" class="btn btn-primary">Customer List</a>
  <h1 class="text-primary">DETAILS</h1>

  <div class="card text-center" style="width: 18rem;">
    
    <div class="card-body">
      <h5 th:text="{customer_find.id}" class="card-title"></h5>
      <p th:text="{customer_find.name}" class="card-text"></p>
      <p th:text="{customer_find.surname}" class="card-text"></p>
      <p th:text="{customer_find.email}" class="card-text"></p>
      <p th:text="{customer_find.password}" class="card-text"></p>
      <p th:text="{customer_find.createdDate}" class="card-text"></p>
      <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
  </div>

  <!-- ends codes -->
  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"></script>
</body>
</html>
```

create.html → 1

```
<!doctype html>
<html th:lang="${#locale.language}" xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Thymeleaf String</title>

  <!-- Turkce karakter -->
  <meta charset="utf-8">

  <!-- responsive design -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- seo -->
  <meta name="keywords" content="html5,css3,js,jquery,bootstrap,react,javase"/>
  <meta name="description" content=""/>
  <meta name="author" content=""/>

  <!-- font-awesome -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"/>

  <!-- google font -->
  <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:ital@1&display=swap" rel="stylesheet">

  <!-- Bootstrap CSS v5.0.2 -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

  <!-- jquery -->
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

  <!-- react libraries -->
  <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

  <link rel="stylesheet" th:href="@{/css/style.css}">
  <script th:src="@{/js/template.js}"></script>
</head>
```


create.html → ?

```
<body class="container">

<a th:href="@{/customer/list}" class="btn btn-primary">Customer List</a>
<!-- start codes -->
<h1 class="text-primary">CREATE</h1>

<div class="container">
  <div class="row">
    <h1 class="text-uppercase mt-5 mb-3">Üye Kayıt</h1>
    <form th:action="@{/customer/create}" th:object="${customer_create}" method="post">

      <div class="col-md col-sm">
        <div class="form-group mb-4">
          <input type="text" class="form-control"
            placeholder="Adı" autofocus="true"
            id="name" name="name"
            th:field="*{name}"/>
          <span class="text-danger" th:if="${#fields.hasErrors('name')}}"
            th:errors="*{name}"></span>
        </div>
      </div>

      <div class="col-md col-sm">
        <div class="form-group mb-4">
          <input type="text" class="form-control" placeholder="soyadı" id="surname"
            name="surname" th:field="*{surname}"/>
          <span class="text-danger" th:if="${#fields.hasErrors('surname')}}"
            th:errors="*{surname}"></span>
        </div>
      </div>
    </div>
  </div>
```

create.html → 3

```
<div class="col-md col-sm">
  <div class="form-group mb-4">
    <input type="text" class="form-control" placeholder="Email" id="email" name="email"
      th:field="*{email}"/>
    <span class="text-danger" th:if="${#fields.hasErrors('email')}" th:errors="*{email}"></span>
  </div>
</div>

<div class="col-md col-sm">
  <div class="form-group mb-4">
    <input type="text" class="form-control" placeholder="password" id="password" name="password"
      th:field="*{password}"/>
    <span class="text-danger" th:if="${#fields.hasErrors('password')}" th:errors="*{password}"></span>
  </div>
</div>

<div class="col-md col-sm">
  <div class="form-group mb-4">
    <button type="reset" class="btn btn-warning">Temizle</button>
    <button type="submit" class="btn btn-primary">Gönder</button>
  </div>
</div>
</form>
</div>
```

create.html → 4

```
<!-- ends codes -->  
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"></script>  
</body>  
</html>
```

update.html → 1

```
<!doctype html>
<html th:lang="${#locale.language}" xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Thymeleaf String</title>

  <!-- Turke karakter -->
  <meta charset="utf-8">

  <!-- responsive design -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- seo -->
  <meta name="keywords" content="html5,css3,js,jquery,bootstrap,react,javase"/>
  <meta name="description" content=""/>
  <meta name="author" content=""/>

  <!-- font-awesome -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"/>

  <!-- google font -->
  <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:ital@1&display=swap" rel="stylesheet">

  <!-- Bootstrap CSS v5.0.2 -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

  <!-- jquery -->
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

  <!-- react libraries -->
  <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

  <link rel="stylesheet" th:href="@{/css/style.css}">
  <script th:src="@{/js/template.js}"></script>
</head>
```

update.html → 2

```
<body class="container">

<a th:href="@{/customer/list}" class="btn btn-primary">Customer List</a>
<!-- start codes -->
<h1 class="text-primary">UPDATE</h1>

<div class="container">
  <div class="row">
    <h1 class="text-uppercase mt-5 mb-3">Üye Güncelle</h1>
    <form th:action="@{/customer/update/{id}}(id=${customer_update.id})" th:object="${customer_update}" th:method="post">

      <div class="col-md col-sm">
        <div class="form-group mb-4">
          <input type="text" class="form-control"
            placeholder="Adı" autofocus="true"
            id="name" name="name"
            th:field="*{name}"/>
          <span class="text-danger" th:if="${#fields.hasErrors('name')}"
            th:errors="*{name}"></span>
        </div>
      </div>

      <div class="col-md col-sm">
        <div class="form-group mb-4">
          <input type="text" class="form-control"
            placeholder="soyadı" id="surname"
            name="surname" th:field="*{surname}"/>
          <span class="text-danger" th:if="${#fields.hasErrors('surname')}"
            th:errors="*{surname}"></span>
        </div>
      </div>
    </div>
  </div>
```

update.html → 1

```
<div class="col-md col-sm">
  <div class="form-group mb-4">
    <input type="text" class="form-control"
      placeholder="Email" id="email" name="email"
      th:field="*{email}"/>
    <span class="text-danger" th:if="${#fields.hasErrors('email')}" th:errors="*{email}"></span>
  </div>
</div>

<div class="col-md col-sm">
  <div class="form-group mb-4">
    <input type="text" class="form-control"
      placeholder="password" id="password" name="password"
      th:field="*{password}"/>
    <span class="text-danger" th:if="${#fields.hasErrors('password')}" th:errors="*{password}"></span>
  </div>
</div>

<div class="col-md col-sm">
  <div class="form-group mb-4">
    <button type="reset" class="btn btn-warning">Temizle</button>
    <button type="submit" class="btn btn-primary">Gönder</button>
  </div>
</div>
</form>
</div>
</div>
```

update.html → 1

```
<!-- ends codes -->  
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"></script>  
</body>  
</html>
```

```

<!doctype html>
<html th:lang="${#locale.language}" xmlns:th="http://www.thymeleaf.org" >
<head>
  <title>Thymeleaf String</title>

  <!-- Turke karakter -->
  <meta charset="utf-8">

  <!-- responsive design -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- seo -->
  <meta name="keywords" content="html5,css3,js,jquery,bootstrap,react,javase" />
  <meta name="description" content="" />
  <meta name="author" content="" />

  <!-- font-awesome -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css" />

  <!-- google font -->
  <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:ital@1&display=swap" rel="stylesheet">

  <!-- Bootstrap CSS v5.0.2 -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

  <!-- jquery -->
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

  <!-- react libraries -->
  <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

  <link rel="stylesheet" th:href="@{/css/style.css}">
  <script th:src="@{/js/template.js}"></script>
</head>

```



```
<body class="container">

  <!-- start codes -->
  <h1 class="text-primary">Logout</h1>
  <p th:text="${admin_logout}"></p>
  <a th:href="@{/customer/list}" class="btn btn-primary">Customer List</a>

  <!-- ends codes -->
  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"></script>
</body>
</html>
```


Spring Security

```
<!--security-->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>

<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-test</artifactId>
  <scope>test</scope>
</dependency>
```

application.properties

```
#####  
### Spring Security #####  
# Security 1.YOL KAPATMAK (@SpringBootApplication kapatmak istersek)  
# @SpringBootApplication(exclude = {SecurityAutoConfiguration.class})  
  
# Security 2.YOL KAPATMAK  
# spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.security.SecurityAutoConfiguration  
  
# WebSecurity için kullanacağım  
spring.security.user.name=hamitmizrak@gmail.com  
spring.security.user.password=root  
spring.security.user.roles=ADMIN  
# securityConfigurations  
# .formLogin()  
# .loginPage("/login")  
# .usernameParameter("username")  
# .passwordParameter("password")
```

@SpringBootApplication

```
// Spring Security
/*
@SpringBootApplication(exclude = {
    //SecurityAutoConfiguration.class,
    org.springframework.boot.autoconfigure.security.servlet.SecurityAutoConfiguration.class,
    org.springframework.boot.actuate.autoconfigure.security.servlet.ManagementWebSecurityAutoConfiguration.class
})
*/

@SpringBootApplication
public class TurgutUniversitySpringAllInOneApplication {
```

SecurityMvc

```
package com.hamitmizrak.controller.mvc;

import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.web.authentication.logout.SecurityContextLogoutHandler;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class SecurityMvc {

    // HOME
    // http://localhost:2222
    // http://localhost:2222/home
    @GetMapping("/home")
    public String homePage(){
        return "index";
    }
}
```

logout

```
// LOGOUT
// http://localhost:2222/logout44
@GetMapping("/logout44")
public String logoutPage(HttpServletRequest request, HttpServletResponse response, Model model){
    //sistemdeki kullanıcı ismini Session ile almalısın
    //sayfaya giriş yapmış userlar
    Authentication authentication= SecurityContextHolder.getContext().getAuthentication();
    if(authentication!=null && authentication.isAuthenticated()) {
        System.out.println(authentication);
        System.out.println(authentication.getName());
        System.out.println(authentication.getPrincipal());
        new SecurityContextLogoutHandler().logout(request,response,authentication);
        model.addAttribute("admin_logout","Sistemden çıkış yapıldı");
    }else{
        model.addAttribute("admin_logout","Sistemden çıkış yapılmadı !!!");
    }
    return "/logout";
}
} //end class
```

```
<!doctype html>
<html th:lang="${#locale.language}" xmlns:th="http://www.thymeleaf.org" >
<head>
  <title>Thymeleaf String</title>

  <!-- Turkce karakter -->
  <meta charset="utf-8">

  <!-- responsive design -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- seo -->
  <meta name="keywords" content="html5,css3,js,jquery,bootstrap,react,javase" />
  <meta name="description" content="" />
  <meta name="author" content="" />

  <!-- font-awesome -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css" />

  <!-- google font -->
  <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:ital@1&display=swap" rel="stylesheet">

  <!-- Bootstrap CSS v5.0.2 -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

  <!-- jquery -->
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

  <!-- react libraries -->
  <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

  <link rel="stylesheet" th:href="@{/css/style.css}">
  <script th:src="@{/js/template.js}"></script>
</head>

<body class="container">
```



```
<body class="container">
```

```
<!-- start codes -->
```

```
<h1 class="text-primary">Logout</h1>
```

```
<p th:text="${admin_logout}"></p>
```

```
<a th:href="@{/customer/list}" class="btn btn-primary">Customer List</a>
```

```
<!-- ends codes -->
```

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"></script>
```

```
</body>
```

```
</html>
```

WebSecurityConfig

```
package com.hamitmizrak.security;

import lombok.RequiredArgsConstructor;
import lombok.SneakyThrows;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.autoconfigure.condition.ConditionalOnProperty;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.crypto.factory.PasswordEncoderFactories;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.web.SecurityFilterChain;

// LOMBOK
@RequiredArgsConstructor

// WebSecurityConfig
@Configuration
@EnableWebSecurity

//h2-console
@ConditionalOnProperty(
    value="spring.h2.console.enabled",
    havingValue = "true",
    matchIfMissing = false)
public class WebSecurityConfig {
```

application.properties → Data

```
//h2-console
@ConditionalOnProperty(
    value="spring.h2.console.enabled",
    havingValue = "true",
    matchIfMissing = false)
public class WebSecurityConfig {

    // Global Variable ==> application.properties
    @Value("${spring.security.user.name}")
    private String username;

    @Value("${spring.security.user.password}")
    private String password;

    @Value("${spring.security.user.roles}")
    private String roles;
```

InMemoryUserDetailsManager

```
// database olmadan kullanım : InMemoryUserDetailsManager
// Hashsiz hali
@Bean
public InMemoryUserDetailsManager inMemoryUserDetailsManager(PasswordEncoder passwordEncoder) {
    //admin 2 rolü var
    UserDetails admin = User
        .withDefaultPasswordEncoder() //withUserDetails(username)
        .username(username)
        .password(passwordEncoder.encode(password))
        .password(password)
        .roles(roles, "ROLES_USER")
        .build();

    //user 1 rolü var
    UserDetails user = User
        .withDefaultPasswordEncoder() //withUserDetails(username)
        .username("root2")
        .password("root2")
        .roles("ROLES_USER2")
        .build();

    return new InMemoryUserDetailsManager(admin, user);
}
```

```
// 2.YOL
// Bunu yazarsak; InMemoryUserDetailsManager yukarıdaki yazdığımız kabul etmedi.
//injection
/*
@Autowired
private PasswordEncoderBean passwordEncoderBean;

@SneakyThrows
@Autowired // injection
// {noop} ==> maskeleymeden sakla
public void myUserPasswordRoles(AuthenticationManagerBuilder authenticationManagerBuilder){
    authenticationManagerBuilder
        .inMemoryAuthentication()
        .withUser("root3")
        //.password("{noop}root3")
        .password(passwordEncoderBean.passwordEncoderMethod().encode("root3"))
        .roles("ROLES_USER3");
}
*/
```

SecurityFilterChain

```
// securityFilterChain
@Bean
@SneakyThrows // throws Exception
public SecurityFilterChain securityFilterChain(HttpSecurity httpSecurity) {
    httpSecurity
        .csrf().disable()
        .authorizeRequests()
        .requestMatchers("/login").permitAll()
        .requestMatchers("/home", "/").permitAll()
        .requestMatchers("/logout").permitAll()
        .requestMatchers("/h2-console/**").permitAll()
        .anyRequest()
        .authenticated()
        .and().csrf().ignoringRequestMatchers("/h2-console/**")
        .and().headers().frameOptions().sameOrigin()
        .and()
        //.httpBasic();
        .formLogin()
        .and()
        .logout().logoutUrl("/logout44").invalidateHttpSession(true); // var olan sessionları kapatıyor.
    return httpSecurity.build();
}
```

PasswordEncoder

```
@Bean
public PasswordEncoder passwordEncoder() {
    PasswordEncoder passwordEncoder = PasswordEncoderFactories.createDelegatingPasswordEncoder();
    return passwordEncoder;
}

} //end class
```