

YAZILIM GELİŞTİRME VE ORTAM ARAÇLARI FİNAL PROJE RAPORU

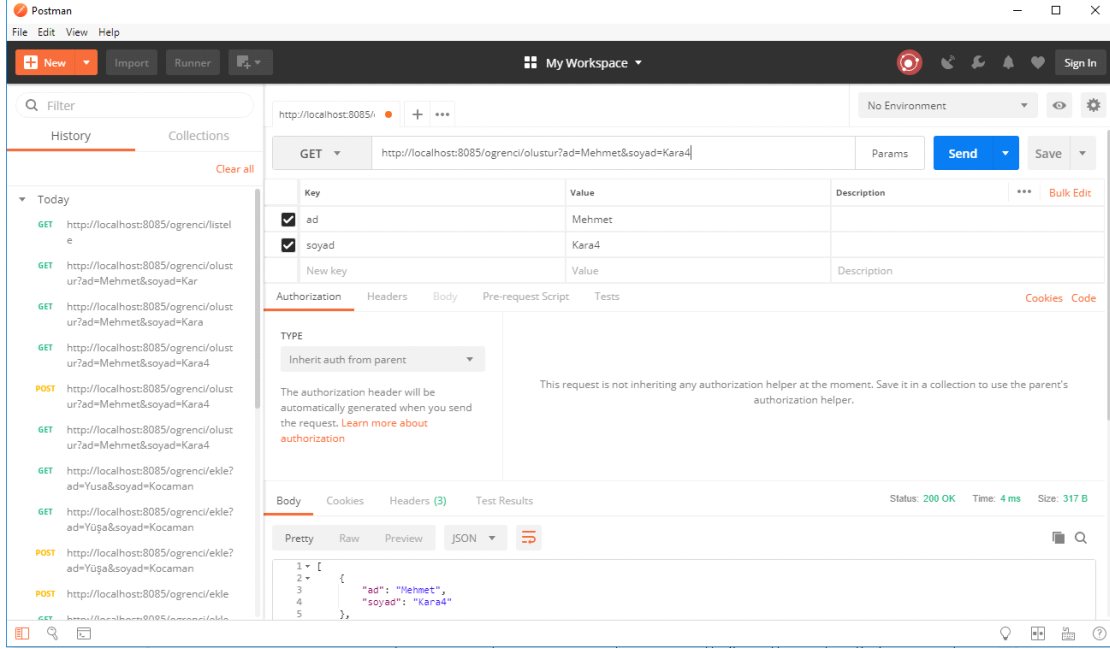
YÜŞA KOCAMAN H5160015

RETServisi:

```
SpringBootOrnek/pom.xml *RETServisiOrnek.java SpringBootOrnekApplication.java application.properties
1 package com.medipol.yazilimaraclari.SpringBootOrnek;
2
3 import java.util.*;
4 import org.springframework.web.bind.annotation.*;
5
6 @RestController
7 public class RETServisiOrnek {
8
9     private List<Ogrenci> ogrenciler = new ArrayList<Ogrenci>();
10
11     public RETServisiOrnek() {}
12
13 }
14 /**
15  *
16  * @param ogrenciler
17  */
18
19 public RETServisiOrnek(List<Ogrenci> ogrenciler) {
20     this.ogrenciler = ogrenciler;
21 }
22
23 class Ogrenci {
24     String ad, soyad;
25     public String getAd() { return ad;}
26     public String getSoyad() { return soyad;}
27 }
28
29 /** http://localhost:9031/ogrenci/listele */
30 @RequestMapping("/ogrenci/listele")
31 public List<Ogrenci> ogrenciListele() {
32     return ogrenciler;
33 }
34
35 /** http://localhost:9031/ogrenci/olustur?ad=Mehmet&soyad=Kara4 */
36 @RequestMapping(value="/ogrenci/olustur",method = RequestMethod.GET)
37 public synchronized Ogrenci ogrenciOlustur(String ad, String soyad) {
38     return ogrenciEkle(ogrenciler, ad, soyad);
39 }
40
41 @RequestMapping(value="/ogrenci/olustur/post",method = RequestMethod.POST)
42 public synchronized Ogrenci ogrenciOlusturPost(String ad, String soyad) {
43     return ogrenciEkle(ogrenciler, ad, soyad);
44 }
45
46 protected Ogrenci ogrenciEkle(List<Ogrenci> liste, String ad, String soyad) {
47     Ogrenci ogrenci = new Ogrenci();
48     ogrenci.ad=ad;
49     ogrenci.soyad=soyad;
50     liste.add(ogrenci);
51     return ogrenci;
52 }
53 }
```

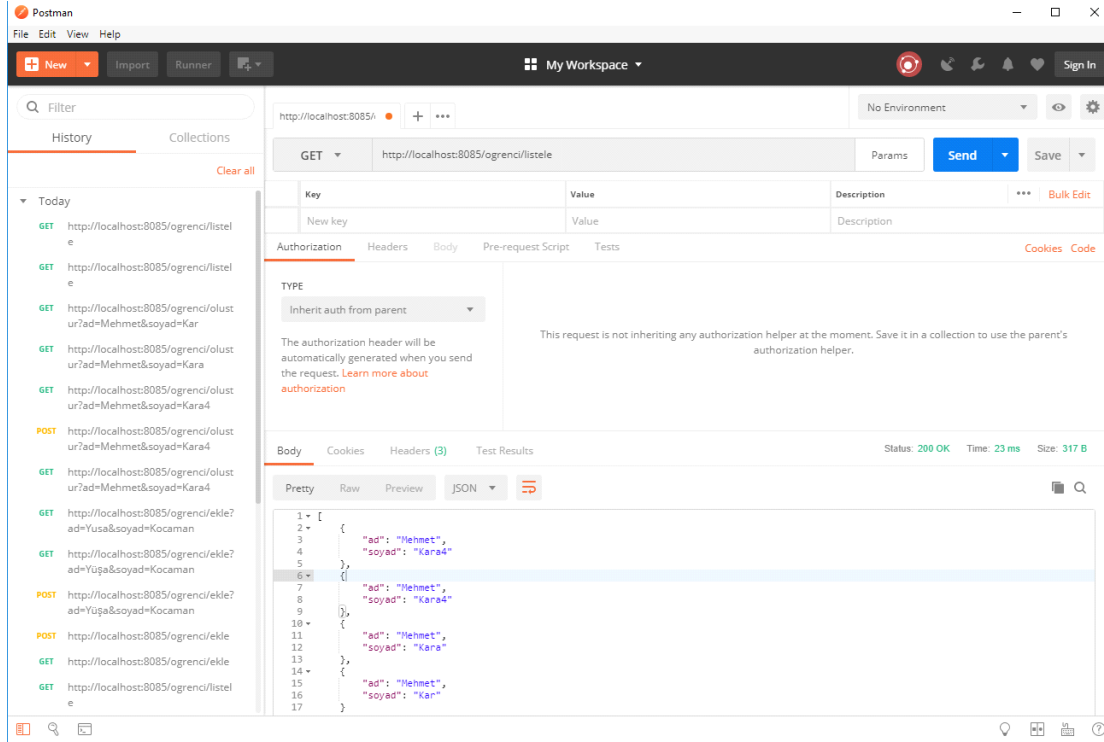
Maven projesidir. Öğrenci ekleme ve öğrenci listeleme vardır. Öğrenci silme yoktur.

POSTMAN'DE EKLEME:



Postmanda params'dan kişi eklemek.

POSTMAN'DE LİSTELEME:



Postmanda eklediğimiz kişilerin listesi.

JAVADOC KULLANIMI:

```
15 /**
16  *
17  * @param ogrenciler
18  *
19  */
20
```

Javadoc olarak projenin içerisinde @param etiketi kullanılmıştır. Bu @param etiketi metod parametrelerini gösterir. Java bu yorumun ismi dökümantasyon yorumu olarak adlandırılır.

GİTHUB'A REPOSİTORY NASIL COMMITLENİR:

Github hesabımıza giriş yapıp Repositories diyip ordan New diyoruz Repositoriese bi isim veriyoruz ve "Initialize this repository with a README"yi işaretliyoruz Create repository diyoruz.Clone or

download'dan linki kopyalıyoruz .

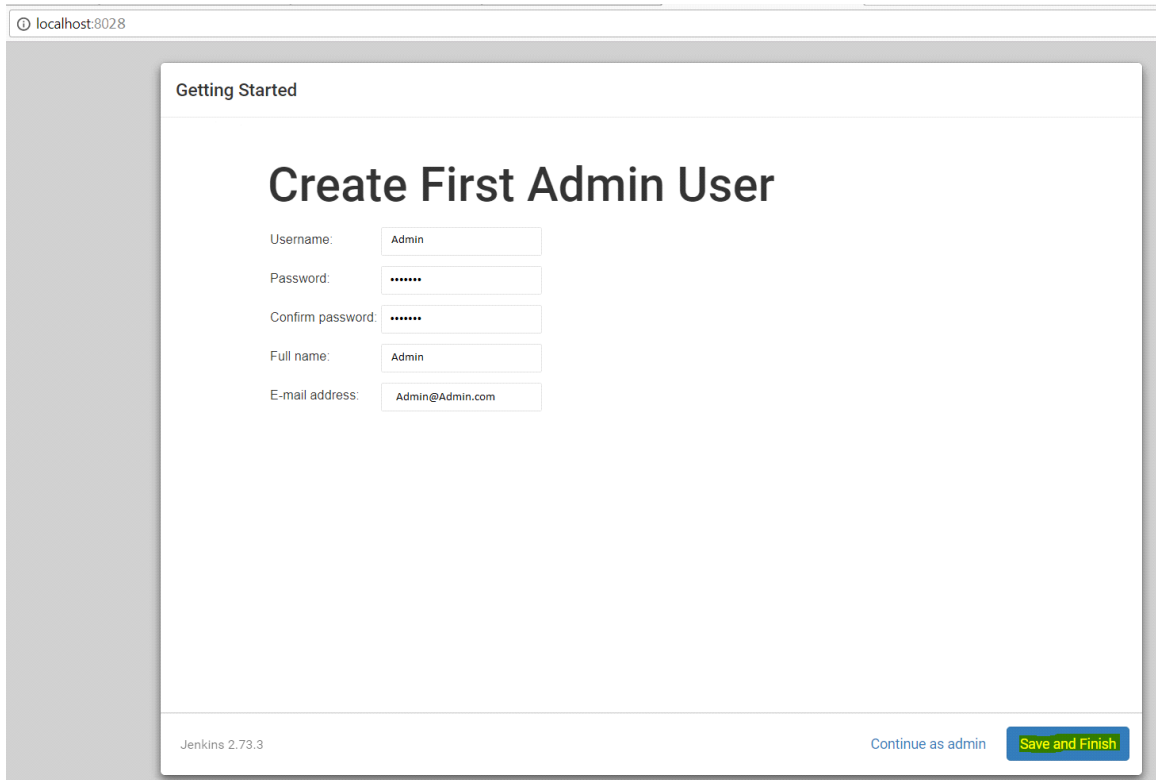
Window->Show view->Other ' e tıklıyoruz ordan açılan kareden Git Repositories ve Git Staging 'i açıyoruz.

Github hesabımıza yüklemek istediğimiz projenin üstüne gelip sağ tıklıyoruz Team->Share Project diyoruz karşımıza çıkan Repository'yi seçiyoruz ve Finish diyoruz.

Git Staging kısmına geliyor bütün dosyalar Seç sürükle işlemi ile hepsini Staged Changes'e atıyoruz ve bi Commit Message yazmamız gerekiyor bu işlemleri yaptıktan sonra Commit and Push... diyoruz. Eclipse sizden Github hesabına giriş yapmanızı istiyor Github'a giriş yaptıktan sonra işleminiz sonlanmıştır.

<https://github.com/Yusakocaman1997/Yeni>

JENKINS KULLANIM:



The screenshot shows the Jenkins 'Getting Started' page in a web browser. The page title is 'Getting Started'. The main heading is 'Create First Admin User'. Below the heading, there are five input fields with labels and pre-filled values:

- Username: Admin
- Password: *****
- Confirm password: *****
- Full name: Admin
- E-mail address: Admin@Admin.com

At the bottom left, it says 'Jenkins 2.73.3'. At the bottom right, there are two links: 'Continue as admin' and a blue button labeled 'Save and Finish'.

Burayı istediğiniz bi şekilde doldurduktan sonra sağ alttaki Save and Finish diyoruz.

Global Tool Configuration

Maven Configuration

Default settings provider

Default global settings provider

JDK

JDK installations

☐ JDK

Isim

JAVA_HOME

☒ Install automatically

☐ Install from java.sun.com

Version

☐ I agree to the Java SE Development Kit License Agreement

☒ Installing JDK requires Oracle account. [Please enter your username/password](#)

List of JDK installations on this system

Global Tool Configuration'a tıkladıktan sonra jdk'ya gelip isimi 'Java 8' yapıyoruz ve program C:\'deki java jdk dosyasının uzantısını kopyalayıp java_home olan yere yapıştırıyoruz.

Git

Git installations

☐ Git

Name

Path to Git executable

☒ There's no such executable git.exe in PATH: C:\ProgramData\Oracle\Java\javapath, C:\WINDOWS\system32, C:\WINDOWS, C:\WINDOWS\System32\Wbem, C:\WINDOWS\System32\WindowsPowerShell\v1.0/, C:\WINDOWS\system32\config\systemprofile\dnx\bin, C:\Program Files\Microsoft DNX\Dnvm/, C:\Program Files\Microsoft SQL Server\130\Tools\Binn/, C:\Program Files (x86)\Windows Kits\8.1\Windows Performance Toolkit/, C:\Program Files\MATLAB\R2015b\runtime\win64, C:\Program Files\MATLAB\R2015b\bin, C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn/, C:\Program Files (x86)\Microsoft SQL Server\140\Tools\Binn/, C:\Program Files\Microsoft SQL Server\140\Tools\Binn/, C:\Program Files\Microsoft SQL Server\140\DTs\Binn/, C:\Program Files (x86)\Microsoft SQL Server\140\Tools\Binn\ManagementStudio/, C:\Program Files (x86)\Microsoft SQL Server\140\DTs\Binn/, C:\Program Files (x86)\Microsoft SQL Server\140\Tools\Binn\ManagementStudio/, C:\Program Files\Common Files\Autodesk Shared/, C:\Program Files (x86)\Autodesk\Backburner/, C:\Users\bprtab\AppData\Local\Microsoft\WindowsApps.

☐ Install automatically

☐ JGit

☐ JGit with Apache HTTP client

Git kısmına geldiğimizde iste sadece Add Git diyoruz ve onlardan JGit ve JGit with Apache HTTP client'e tıklarız.

Maven

Maven Installations

Maven

İsim:

☒ Install automatically

Install from Apache


Version:


List of Maven installations on this system


Maven'a gelip isimi 'Maven 3.5.2' yapıyoruz tiki işaretliyoruz ve Versiou 3.5.2'yi seçip sol alttaki save'e tıklıyoruz.


Enter an item name


» Required field


 **Serbest-stil yazılım projesi yapilandir**
Jenkins'in merkezi özelliği, projelerinizi yapılandırmanıza yardım etmesidir. Bu proje türünü kullanarak, herhangi bir yapılandırma sistemini herhangi bir Kaynak Kodu Yönetimi aracı ile birleştirebilirsiniz ve hatta yazılım yapılandırmanın dışında başka tür projeler için dahi kullanabilirsiniz.

 **Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 **Çoklu-konfigürasyona sahip proje yapilandir**
Platforma Özgü yapılandırmalara sahip veya çeşitli ortamlarda test işlemi yapan projeler gibi çok fazla sayıda konfigürasyona ihtiyaç duyan projeler için uygundur.

 **Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

 **GitHub Organization**
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

 **Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Seçilmiş olan yeri seçip ismine Maven Build diyoruz ve sol alttaki OK'a basıyoruz.

Kaynak Kodu Yönetimi

☐ Hiçbiri
☒ Git

Repositories

Repository URL: ?

Credentials: ?

Branches to build

Branch Specifier (blank for 'any'): X ?

Git executable: ?

Repository tarayıcısı: ?

Additional Behaviours:

☐ Subversion ?

Github'ındaki kendi projenin URL'sini kopyalayıp buraya yapıştırıyoruz ve Git executable'ı jgitapache'yi seçiyoruz.

Yapılandırma

☒ En üst seviye Maven hedeflerini çalıştır X ?

Maven Versiyonu: ?

Hedefler: ?

POM: ?

Properties: ?

JVM Options: ?

Inject build variables: ☐ ?

Use private Maven repository: ☐ ?

Settings file: ?

Global Settings file: ?

Windows'u Etki

Maven Version'u Maven 3.5.2 yapıyoruz alttaki hedefler olan yere install yazıyoruz ve pom'da kendi projemizin Pom.xml'i kopyalayıp yapıştırıyoruz ve sonra kaydet diyoruz.

```
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/guava/guava/18.0/guava-18.0.jar (2.3 MB at 739 kB/s)
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ SpringBootOrnek ---
[INFO] Installing C:\Users\bpirlab\.jenkins\workspace\Maven_Build\SpringBootOrnek\target\SpringBootOrnek-0.0.1-SNAPSHOT.jar to
C:\Users\bpirlab\.m2\repository\com\medipol\yazilimaraclari\SpringBootOrnek\0.0.1-SNAPSHOT\SpringBootOrnek-0.0.1-SNAPSHOT.jar
[INFO] Installing C:\Users\bpirlab\.jenkins\workspace\Maven_Build\SpringBootOrnek\pom.xml to C:\Users\bpirlab\.m2\repository\com\medipol\yazilimaraclari\SpringBootOrnek\0.0.1-
SNAPSHOT\SpringBootOrnek-0.0.1-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 47.562 s
[INFO] Finished at: 2018-05-30T11:20:07+03:00
[INFO] Final Memory: 31M/141M
[INFO] -----
Finished: SUCCESS
```

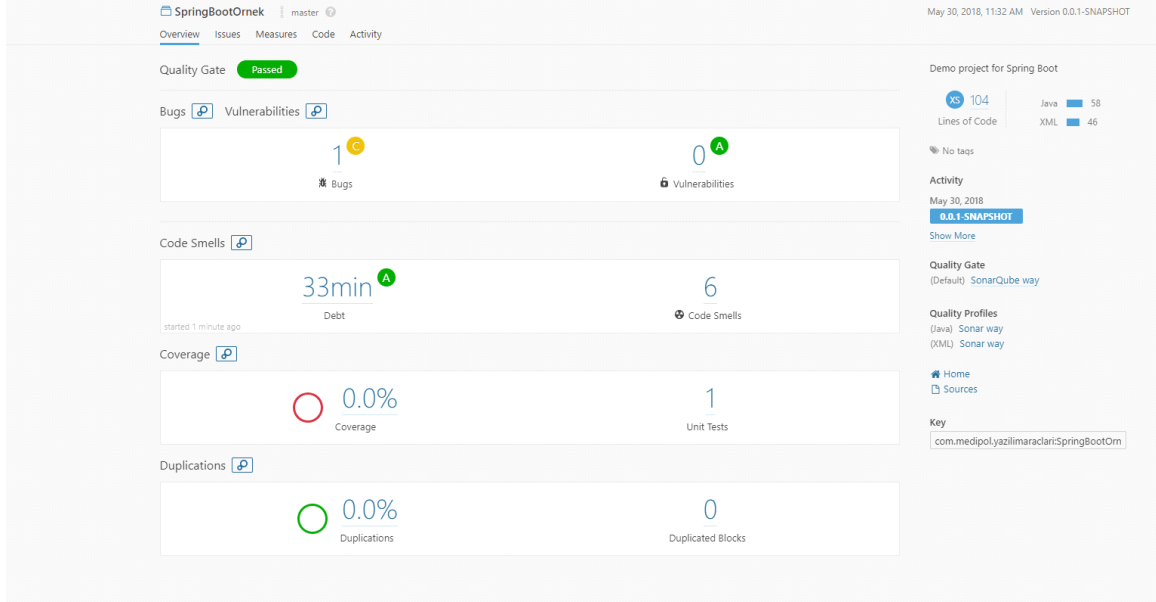
BUILD SUCCESS 'derse eğer herşey yolunda gitmiştir ve jenkins sorunsuz çalışmıştır demektir.

SONARQUBE KULLANIMI:

SonarQubeyi çalıştırmak için jenkins'e ihtiyacımız vardır bu yeride jenkins'ten açıyoruz ve Maven 3.5.2 yapıyoruz hedefleri ise install sonar:sonar yapıyoruz.

```
[INFO] Sensor Zero Coverage Sensor
[INFO] Sensor Zero Coverage Sensor (done) | time=54ms
[INFO] Sensor CPD Block Indexer
[INFO] Sensor CPD Block Indexer (done) | time=30ms
[INFO] SCM provider for this project is: git
[INFO] 4 files to be analyzed
[INFO] 4/4 files analyzed
[INFO] 1 file had no CPD blocks
[INFO] Calculating CPD for 1 file
[INFO] CPD calculation finished
[INFO] Analysis report generated in 256ms, dir size=46 KB
[INFO] Analysis reports compressed in 39ms, zip size=16 KB
[INFO] Analysis report uploaded in 1744ms
[INFO] ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard/index/com.medipol.yazilimaraclari:SpringBootOrnek
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://localhost:9000/api/ce/task?id=AN0uLVa9F53s1NSE_G7s
[INFO] Task total time: 9.328 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 42.098 s
[INFO] Finished at: 2018-05-30T11:32:31+03:00
[INFO] Final Memory: 54M/499M
[INFO] -----
Finished: SUCCESS
```


ANALYSIS SUCCESSFUL yazan yerin yanındaki linke tıklayınca alttaki sayfa açılıyor.



Burada ise yazdığımız kodun bi hatası var mı yoksa hepsi doğru mu diye kontrol eder.