main>resources >application.properties :

server.port=8080  
server.servlet.context-path=/api/1.0  
spring.datasource.url=jdbc:mysql://localhost:3306/harpia  
spring.datasource.username=root  
spring.datasource.password=  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true

spring.datasource.username ve password

MYSQL username ve password’u temsil etmektedir. (Sifresi yok sadece username’i var)

Birde **harpia** adında veri tabanı oluşturulması gerekmektedir.

GET : (browserdan url girince çalışanlar)

* USER
* <http://localhost:8080/api/1.0/users>

Bütün Userlar’ı (Admin, Doktor, Hasta) getirir.

* <http://localhost:8080/api/1.0/users/7>

Id’si 7 olan user’ı getirir. (Bu id’de user yoksa hata verir)

* <http://localhost:8080/api/1.0/users/time/minutes/XXX>

YENİ : Son XXX dakikadaki oluşturulan bütün Userları getirir.

* ADMIN
* <http://localhost:8080/api/1.0/admins/1>

Id’si 1 olan Admin’i getirir. (Bu id’de admin yoksa hata verir)

* DOCTOR
* <http://localhost:8080/api/1.0/doctors>

Bütün doktorları getirir.

* <http://localhost:8080/api/1.0/doctors/3>

Id’si 3 olan doktoru getirir. (Bu id’de admin yoksa hata verir)

* <http://localhost:8080/api/1.0/doctors/3/patients>

Id’si 3 olan doktorun Hastalarını getirir.

* <http://localhost:8080/api/1.0/timers/doctors/patients>

YENİ : Bütün Ölçüm Periot verilerini getirir. (id, hours,minutes,patientId)

* PATIENT
* <http://localhost:8080/api/1.0/patients>

Bütün hastaları getirir.

* <http://localhost:8080/api/1.0/patients/8>

Id’si 8 olan hastanın verisini getirir.

--------------------------

* PATIENT-TIMER
* <http://localhost:8080/api/1.0/timers/doctors/patients>

YENİ : Bütün Ölçüm Periot verilerini getirir. (id, hours,minutes,patientId)

* <http://localhost:8080/api/1.0/timers/doctors/patients/9>

YENİ : 9 id’li hastanın Ölçüm Periot verisini getirir. (id, hours,minutes,patientId)

* BLOOD-RESULT
* <http://localhost:8080/api/1.0/bloodresults>

YENİ : Bütün kan sonuçlarını getirir. (Güncel olandan en eskiye doğru sıralı bir şekilde gelir.)

* <http://localhost:8080/api/1.0/bloodresults/patient/11>

YENİ : 11 Numaralı hastanın bütün kan sonuçlarını getirir. (Güncel olandan en eskiye doğru sıralı bir şekilde gelir.)

* <http://localhost:8080/api/1.0/bloodresults/patient/9/minutes/XXX>

YENİ : 9 numaralı hastanın, son XXX dakikasındaki kan testi sonuçlarını getirir.

POST: (İstek Gönderildiği zaman çalışanlar)

* USER
* <http://localhost:8080/api/1.0/users/login>
* {
* "username":"aes",
* "password":"pass"
* }

Şeklinde veri gönderildiği takdirde eğer username ve password eşleşirse login işlemi yapılır ve user bilgisi dönülür. Aksi halde hata döner.

* ADMIN
* <http://localhost:8080/api/1.0/admins/save>
* {
* "name":"Bulut Akay",
* "lastname":"Ahmet Emin SAGLIK",
* "username":"aes2",
* "password":"pass"
* }

Şeklinde veri gönderildiği takdirde username kayıtlı değilse admin’i kaydeder, aksi halde hata verir.

* DOCTOR
* <http://localhost:8080/api/1.0/doctors>
* {
* "name":"Bulut1",
* "lastname":"Aka1y",
* "username":"akay12",
* "password":"pass",
* "totalPatientNumber" : 5
* }

Şeklinde veri gönderildiği takdirde username kayıtlı değilse doktoru kaydeder, aksi halde hata verir.

* PATIENT
* <http://localhost:8080/api/1.0/patients>
* {
* "name":"ahmet Emin",
* "lastname":"SAGLIK",
* "username":"pat2",
* "password":"pat",
* "diabeticType" : 1
* }

Şeklinde veri gönderildiği takdirde username kayıtlı değil ise hasta kaydeder. Username kayıtlı ise hata verir.

* PATIENT-TIMER
* <http://localhost:8080/api/1.0/patients>
* {
* "patientId": 9,
* "hours" : 1,
* "minutes" : 10
* }

YENİ : Şeklinde veri gönderildiği takdirde patient id’e timer yoksa eklicek. Varsa güncelleyecektir.

* BLOOD-RESULT
* <http://localhost:8080/api/1.0/bloodresults>
* {
* "patientId":9,
* "bloodSugar":150,
* "bloodPressure":200
* }

YENİ : Şeklinde veri gönderildiğinde 9 numaralı hastanın kan değerleri kaydedilmiş olur.