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 (GET)
* <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>

Son XXX dakikadaki oluşturulan bütün Userları getirir.

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

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

* DOCTOR (GET)
* <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>

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

* PATIENT (GET)
* <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.

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

YENİ : Id’si 8 olan hastanın Doctor verisini getirir.

* PATIENT-TIMER (GET)
* <http://localhost:8080/api/1.0/timers>

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

* [http://localhost:8080/api/1.0/timers/patients/9](http://localhost:8080/api/1.0/timers/doctors/patients/9)

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

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

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

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

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>

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

* [http://localhost:8080/api/1.0/bloodresults/patient/9/sixmonth](http://localhost:8080/api/1.0/bloodresults/patient/9/sixmonth/)

YENİ : 9 numarali hastanin son 6 ayındaki kan testi sonuçlarını getirir.

* [http://localhost:8080/api/1.0/bloodresults/patient/9/daily](http://localhost:8080/api/1.0/bloodresults/patient/9/daily/)

YENİ : 9 numarali hastanin son 24 saatteki kan testi sonuçlarını getirir.

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

YENİ : 9 numarali hastanin son 1 haftadaki kan testi sonuçlarını getirir.

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

YENİ : 9 numarali hastanin son 1 haftadaki kan testi sonuçlarını getirir.

* FIREBASE-TOKEN (GET)
* [http://localhost:8080/api/1.0/firebase/token/patient/6](http://localhost:8080/api/1.0/firebase/token/patient/6%20)

YENİ : 6 numarali hastanın token verisini getirir.

POST: (İstek Gönderildiği zaman çalışanlar)

* USER (POST)
* <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 (POST)
* <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 (POST)
* <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 (POST)
* <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 (POST)

GÜNCELLENDİ :

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

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

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

Şeklinde veri gönderildiğinde 9 numaralı hastanın kan değerleri kaydedilmiş olur.

* FIREBASE-TOKEN (GET)
* [http://localhost:8080/api/1.0/firebase/token/patient/6](http://localhost:8080/api/1.0/firebase/token/patient/6%20)

YENİ : 6 numarali hastanın token verisini getirir.

PUT: (İstek Gönderildiği zaman çalışanlar)

YENİ :

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

Veya (herhangi bir özellik mesela “name” i örnek alalım.)

{

    "name": "Bulut Akay"

}

Şeklinde veri gönderildiği takdirde admin’i günceller. Eğer username kayıtlı ise tekrar aynı username gönderildiği takdirde hata vericektir.

- Doctor (PUT)

YENİ :

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

{

    "id": 3,

    "roleId": 2,

    "name": "Ahmet Emin",

    "lastname": "SAGLIK",

    "password": "pass",

    "specialization": "Endocrin",

    "graduate": "KTU"

}

Veya (herhangi bir özellik mesela “graduate” i örnek alalım.)

{

    "graduate": "KTU"

}

Şeklinde veri gönderildiği takdirde doctor’u günceller. Eğer username kayıtlı ise tekrar aynı username gönderildiği takdirde hata vericektir.

- Patient (PUT)

YENİ :

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

{

    "id": 6,

    "name": "p22at1",

    "lastname": "pat222",

    "password": "pa22t1",

    "doctorId": 3,

    "diabeticTypeId": 1

}

Veya (herhangi bir özellik mesela “doctorId” i örnek alalım.)

{

    "doctorId": 5

}

Şeklinde veri gönderildiği takdirde patient’i günceller. Eğer username kayıtlı ise tekrar aynı username gönderildiği takdirde hata vericektir.