**ClientsController:**

* **registerClient(String firstName, String lastName,String userName, String userPassword, String address, String phone, String email): adauga un nou client in baza de date;**
* **getByAccount(String username, String password): ia un client din baza de date dupa username si parola;**
* **getByUsername(String username): ia un client din baza de date dupa username**
* **getById(int id): ia un client din baza de date dupa id;**
* **changePassword(String oldPassword, String newPassword): face un update in baza de date pentru a schimba vechea parola cu una noua;**
* **updateData(String email, String phoneNumber): modifica email-ul si numarul de telefon al unui client dat;**
* **getAll() :aceasta metoda interogheaza tabelul clients si ia toate intrarile si le pune intr-o lista de clienti;**
* **getAllOnlyGeneralData(): aceasta metoda interogheaza tabelul clients si ia toate id-urile si username-urile clientilor si le pune intr-o lista de pair-uri.**

**ClientsContractsController:**

* **createClientContract(int idClient, int idCompany, Date startDate, Date endDate): adauga un nou contract in baza de date;**
* **getById(int id): selecteaza contractul dupa id-ul specificat;**
* **getByClient(int id): selecteaza contractele dupa id-ul clientului specificat si le pune intr-o lista;**
* **getByCompany(int id): selecteaza contractele dupa id-ul companiei specificat si le pune intr-o lista.**

**CompaniesContoller:**

* **getByAccount(String username, String password): selecteaza o companie dupa username si parola;**
* **getByUsername(String username): selecteaza o companie dupa username;**
* **getById(int id): selecteaza o companie dupa id;**
* **getAll(): selecteaza toate companiile si le pune intr-o lista.**

**EmployeeContractsController :**

* **createEmployeeContract(int companyId, int employeeId, java.sql.Date startDate, java.sql.Date endDate) : adauga un nou contract al angajatului in baza de date;**
* **getById(int** **id**) : **ia un contract al angajatului din baza de date dupa id-ul contractului;**
* **getByEmployee(int id) : interogheaza baza de date si returneaza o lista cu toate contractele angajatului cu id-ul dat;**
* **getByCompany(int id) : interogheaza baza de date si returneaza o lista cu toate contractele angajatilor care lucreaza la compania cu id-ul dat.**

**EmployeesController :**

* **registerEmployee(String firstName, String lastName, String userName, String password, String phone, String email) : adauga un nou angajat in baza de date;**
* **getByAccount(String username, String password) : ia un angajat din baza de date care are contul (user-ul si parola) dat;**
* **getByUsername(String username) : ia angajatul din baza de date care are user-ul dat;**
* **getById(int id) : ia angajatul din baza de date care are id-ul dat;**
* **changePassword(String oldPassword, String newPassword) : face un update in baza de date pentru a schimba vechea parola cu una noua;**
* **updateData(String email, String phoneNumber) : modifica e-mail-ul si numarul de telefon al unui angajat;**
* **getAll() : interogheaza baza de date si returneaza o lista cu toti angajatii din baza de date;**
* **getAllOnlyGeneralData() : interogheaza tabelul angajati si returneaza o lista cu id-urile si username-urile tuturor angajatilor (ca perechi).**

**LoginController :**

* **connectClient(String username, String password) : daca clientul cu user-ul si parola date este in baza de date,atunci adauga intr-o structura de date globala user-ul si id-ul acestuia;**
* **connectEmployee(String username, String password) : daca angajatul cu user-ul si parola date este in baza de date,atunci il adauga intr-o structura de date globala user-ul si id-ul acestuia;**
* **connectCompany(String username, String password) : daca compania cu user-ul si parola date este in baza de date,atunci il adauga intr-o structura de date globala user-ul si id-ul acesteia.**

**MeetingsController :**

* **createMeeting(int clientId, int employeeId, Date date, String Description) : adauga o noua intalnire in baza de date;**
* **getAll() : interogheaza baza de date si returneaza o lista cu toate intalnirile din tabelul intalniri;**
* **getByClient(int clientId) : returneaza o intalnire a clientului cu id-ul dat;**
* **getAllByEmployee(int employeeId) : interogheaza baza de date si returneaza o lista cu toate intalnirile angajatului cu id-ul dat;**
* **getAllOpenedByEmployee** **(int employeeId) : interogheaza baza de date si returneaza o lista cu toate intalnirile deschise ale unui angajat cu id-ul dat;**
* **getAllClosedByEmployee(int employeeId) : interogheaza baza de date si returneaza o lista cu toate intalnirile inchise ale unui angajat cu id-ul dat;**
* **getByMeeting(int id) : retunreaza intalnirea cu id-ul dat.**

**ReportsController :**

* **createReport(int idMeeting, String description) : adauga un nou raport(id-ul intalnirii si descrierea) in baza de date;**
* **getAll() : returneaza o lista cu toate rapoartele din tabelul raport;**
* **getById(int id) : interogheaza baza de date si returneaza raportul cu id-ul dat;**
* **getByMeeting(int id) : interogheaza baza de date si returneaza raportul cu id-ul intalnirii dat.**