

24. csoport prjoektmunka

Rendszerfejlesztés korszerű módszerei



2021/22/2

Ficsor attila, selmeci ferenc, kaposi áron erik, bencsik ádám zoltán

Tartalom

[Projekt leírása 2](#_Toc28329)

[Adatmodell 3](#_Toc28471)

[Osztályok 3](#_Toc31163)

[Szerepek és jogosultságok 5](#_Toc30742)

[Adatbázis 6](#_Toc7168)

[Egyszerűsített diagram 6](#_Toc10557)

[Bővített diagram 7](#_Toc21704)

[Végpontok 8](#_Toc24016)

[Architechtúra 14](#_Toc20334)

# Projekt leírása

Majd egyszer

# Adatmodell

## Osztályok

1. Felhasználó:
   * Azonosító: szám
   * Felhasználónév: szöveg
   * Jelszó: szöveg
   * Szerepkör: összetett (szerepkör)
   * Képesítés: lista (összetett: képesítés)
2. Szepekör:
   * Azonosító: szám
   * Név: szöveg
   * Jogosultságok: lista (összetett: jogosultság)
3. Jogosultság:
   * Azonosító: szám
   * Név: szöveg
4. Képesítés:
   * Azonosító: szám
   * Név: szöveg
5. Feladat:
   * Azonosító: szám
   * Név: szöveg
   * Kategória: összetett (kategória)
   * Hozzárendelt felhasználó: összetett (felhasználó)
   * Prioritás: szám
   * EventLog: lista (összetett: log)
6. Log:
   * Azonosító: szám
   * Státusz: szöveg
   * Időpont: dátum
   * Felhasználó: összetett (felhasználó)
7. Kategória:
   * Azonosító: szám
   * Rendkívüli: logikai
   * Név: szöveg
   * Normaidő: szám
   * Intervallum: összetett (intervallum)
   * Szülőkategória: összetett (kategória)
   * Előírások: szöveg
8. Intervallum:
   * Azonosító: szám
   * Mérték: szöveg
9. Eszköz:
   * Azonosító: szám
   * Név: szöveg
   * Kategória: összetett (kategória)
   * Leírás: szöveg
   * Elhelyezkedés: összetett (elhelyezkedés)
10. Elhelyezkedés:
    * Azonosító: szám
    * Épület: szöveg
    * Terem/csarnok: szöveg

## Szerepek és jogosultságok

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Eszközfelelős | Operátor | Karbantartó |
| Felhasználó | Lekérés | - | Igen | - |
| Módosítás | - | Igen | - |
| Szerepkör | Lekérés | - | Igen | - |
| Módosítás | - | - | - |
| Jogosultság | Lekérés | - | Igen | - |
| Módosítás | - | - | - |
| Képesítés | Lekérés | - | Igen | - |
| Módosítás | - | Igen | - |
| Feladat | Lekérés | - | Igen | Igen |
| Módosítás | - | Igen | Igen |
| Log | Lekérés | - | Igen | Igen |
| Módosítás | - | - | - |
| Kategória | Lekérés | igen | Igen | Igen |
| Módosítás | Igen | - | - |
| Intervallum | Lekérés | Igen | Igen | Igen |
| Módosítás | Igen | - | - |
| Eszköz | Lekérés | Igen | Igen | Igen |
| Módosítás | Igen | - | - |
| Elhelyezkedés | Lekérés | Igen | Igen | Igen |
| Módosítás | igen | - | - |

## Adatbázis

### Egyszerűsített diagram



### Bővített diagram



## Végpontok

A Get és Post végpontok hívása GET, a Put és Delete végpontoké pedig ADMIN jogosultsághoz kötött.  
A Post és Delete végpontok paraméterként egy id-t kapnak.  
A Put végpont egy megfelelő típusú objektumot kap body-ban.

1. /user
   * Get: lista
     1. Kérés: nem kap semmit
     2. Válasz: egy lista a megfelelő objektumokkal
     3. Kivételek:
   * Post: felhasználó
     1. Kérés: egy id-t kap
     2. Válasz: egy user objektumot ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Put:módosítás
     1. Kérés: egy user objektumot kap
     2. Válasz: egy státusz üzenetet ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Delete: törlés
     1. Kérés: egy id-t kap/egy objektumot
     2. Válasz: egy státusz üzenettel tér vissza
     3. Kivételek:
        1. DEPENDENT: ha valamilyen másik objektum hivatkozik rá
2. /role
   * Get: lista
     1. Kérés: nem kap semmit
     2. Válasz: egy lista a megfelelő objektumokkal
     3. Kivételek:
   * Post: szerepkör
     1. Kérés: egy id-t kap
     2. Válasz: egy role objektumot ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Put:módosítás
     1. Kérés: egy role objektumot kap
     2. Válasz: egy státusz üzenetet ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Delete: törlés
     1. Kérés: egy id-t kap/egy objektumot
     2. Válasz: egy státusz üzenettel tér vissza
     3. Kivételek:
        1. DEPENDENT: ha valamilyen másik objektum hivatkozik rá
3. /permission
   * Get: lista
     1. Kérés: nem kap semmit
     2. Válasz: egy lista a megfelelő objektumokkal
     3. Kivételek:
   * Post: jogosultság
     1. Kérés: egy id-t kap
     2. Válasz: egy permission objektumot ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Put:módosítás
     1. Kérés: egy permission objektumot kap
     2. Válasz: egy státusz üzenetet ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Delete: törlés
     1. Kérés: egy id-t kap/egy objektumot
     2. Válasz: egy státusz üzenettel tér vissza
     3. Kivételek:
        1. DEPENDENT: ha valamilyen másik objektum hivatkozik rá
4. /tool
   * Get: lista
     1. Kérés: nem kap semmit
     2. Válasz: egy lista a megfelelő objektumokkal
     3. Kivételek:
   * Post: eszköz
     1. Kérés: egy id-t kap
     2. Válasz: egy tool objektumot ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Put:módosítás
     1. Kérés: egy tool objektumot kap
     2. Válasz: egy státusz üzenetet ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Delete: törlés
     1. Kérés: egy id-t kap/egy objektumot
     2. Válasz: egy státusz üzenettel tér vissza
     3. Kivételek:
        1. DEPENDENT: ha valamilyen másik objektum hivatkozik rá
5. /issue
   * Get: lista
     1. Kérés: nem kap semmit
     2. Válasz: egy lista a megfelelő objektumokkal
     3. Kivételek:
   * Post: feladat
     1. Kérés: egy id-t kap
     2. Válasz: egy issue objektumot ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Put: módosítás
     1. Kérés: egy issue objektumot kap
     2. Válasz: egy státusz üzenetet ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Delete: törlés
     1. Kérés: egy id-t kap/egy objektumot
     2. Válasz: egy státusz üzenettel tér vissza
     3. Kivételek:
        1. DEPENDENT: ha valamilyen másik objektum hivatkozik rá
6. /qualification
   * Get: lista
     1. Kérés: nem kap semmit
     2. Válasz: egy lista a megfelelő objektumokkal
     3. Kivételek:
   * Post: képesítést
     1. Kérés: egy id-t kap
     2. Válasz: egy qualification objektumot ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Put:módosítás
     1. Kérés: egy qualification objektumot kap
     2. Válasz: egy státusz üzenetet ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Delete: törlés
     1. Kérés: egy id-t kap/egy objektumot
     2. Válasz: egy státusz üzenettel tér vissza
     3. Kivételek:
        1. DEPENDENT: ha valamilyen másik objektum hivatkozik rá
7. /log (egy feladat id-t kap)
   * Get:visszaadja az adott id-hoz tartozó logokat
     1. Kérés: egy feladat id-t kap
     2. Válasz: egy lista a megfelelő log objektumokkal
     3. Kivételek:
8. /category
   * Get: lista
     1. Kérés: nem kap semmit
     2. Válasz: egy lista a megfelelő objektumokkal
     3. Kivételek:
   * Post: kategória
     1. Kérés: egy id-t kap
     2. Válasz: egy category objektumot ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Put:módosítás
     1. Kérés: egy category objektumot kap
     2. Válasz: egy státusz üzenetet ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Delete: törlés
     1. Kérés: egy id-t kap/egy objektumot
     2. Válasz: egy státusz üzenettel tér vissza
     3. Kivételek:
        1. DEPENDENT: ha valamilyen másik objektum hivatkozik rá
9. /interval
   * Get: lista
     1. Kérés: nem kap semmit
     2. Válasz: egy lista a megfelelő objektumokkal
     3. Kivételek:
   * Post: intervallum
     1. Kérés: egy id-t kap
     2. Válasz: egy interval objektumot ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Put:módosítás
     1. Kérés: egy interval objektumot kap
     2. Válasz: egy státusz üzenetet ad vissza
     3. Kivételek:
        1. NOT\_FOUND: ha nincs ilyen id-val objektum
   * Delete: törlés
     1. Kérés: egy id-t kap/egy objektumot
     2. Válasz: egy státusz üzenettel tér vissza
     3. Kivételek:
        1. DEPENDENT: ha valamilyen másik objektum hivatkozik rá
10. /location
    * Get: lista
      1. Kérés: nem kap semmit
      2. Válasz: egy lista a megfelelő objektumokkal
      3. Kivételek:
    * Post: elhelyezkedés
      1. Kérés: egy id-t kap
      2. Válasz: egy location objektumot ad vissza
      3. Kivételek:
         1. NOT\_FOUND: ha nincs ilyen id-val objektum
    * Put:módosítás
      1. Kérés: egy location objektumot kap
      2. Válasz: egy státusz üzenetet ad vissza
      3. Kivételek:
         1. NOT\_FOUND: ha nincs ilyen id-val objektum
    * Delete: törlés
      1. Kérés: egy id-t kap/egy objektumot
      2. Válasz: egy státusz üzenettel tér vissza
      3. Kivételek:
         1. DEPENDENT: ha valamilyen másik objektum hivatkozik rá
11. /login
    * Post:
      1. Kérés:egy emailt/usernevet és egy jelszót kap 'password' ‘email’
      2. Válasz: egy user objektummal tér vissza
      3. Kivételek:
         1. USER\_NOT\_FOUND: ha nincs ilyen nevű felhasználó
         2. BAD\_CREDENTIALS: ha a felh. + jelszó páros nem megfelelő, úgy hogy van ilyen nevű felhasználó
12. /logout
    * Post:
      1. Kérés: egy user objektumot kap
      2. Válasz: egy státusz üzenettel tér vissza
      3. Kivételek:
13. /sign-up

* Post:

1. Kell:
   1. ‘email
   2. ‘password’
   3. ‘trade’
   4. ‘level’
2. vissza:
   1. JSON a sikerességgel / hibával ‘Data’ kulccsal

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

POST : új rekord hozzáadása

GET : adott táblából minden lekérése

PUT: id alapján objektum módosítása

DELETE: id alapján objektum törlése

/role :

POST:

in:

role\_name : string

return:

‘Data’:’Sikeres’

GET:

in: -

return: ‘Roles’: {role\_id : role}

DELETE:

in:

id : ‘int’

return:

‘Data’ : ‘Sikeres’

PUT:

in:

id : int

name : string

return:

‘Data’: ‘Sikeres’ //módosítja az adott role nevét

/location:

Post:

in:

‘building’ : string

‘rooms’: string

return:

‘Data’: ‘Sikeres’

GET:

in:-

return: ‘Locations’ : {location\_id : [building, room]}

DELETE:

in:

id : ‘int’

return:

‘Data’ : ‘Sikeres’

PUT:

in:

id : int

building : string

room : string

return:

‘ Data’: ‘Sikeres’ //módosítja az adott locationt

/trade

POST:

in:

name : string

return:

‘Data’: ‘Sikeres’

GET:

in: -

return : ‘Data’: {trade\_id : trade\_name}

DELETE:

in:

id : int

return:

‘Data’ : ‘Sikeres’

put:

in:

id: int

name : string

return:

‘Data’ : ‘Sikeres’

/category:

POST:

in:

name : string

norma\_time : int //Percekben

interval : int // napok

descript : string

qualification : int (trade\_id)

parent\_id : int (category\_id) //ha nincs, azt is küldd le üres string- ként és akkor beállítja, és az lesz a legfelső kategória szint az adott hierarchiában return:

‘Data’:’Sikeres’

GET:

in:-

return: ‘Data’ : {Minden kategória id a kulcs}

DELETE:

in

id: int

return:

‘Data’:’Sikeres’

PUT:

in:

id : int

name : string

norma\_time : int //Percekben

interval : int // napok

descript : string

qualification : int (trade\_id)

parent\_id : int (category\_id)

return :

‘Data’. ‘Sikeres’

/item:

POST:

in:

name : string

descript: string

category: int (category\_id)

location : int (location\_id)

return:

‘Data’. ‘Sikeres’

GET:

in: -

return: minden location ‘Data’ kulccsal.

DELETE:

in:

id: int

return: ‘Data’: ‘Sikeres’

PUT:

in:

id : int

name : string

descript: string

category: int (category\_id)

location : int (location\_id)

return:

‘Data’ : ‘Sikeres’

Hibát még nem küld vissza a legtöbb, ezt még meg kell csinálnunk, de általánosan az lesz hogyha hiba van akkor return ‘Data’ : ‘Error\_type’ mindenhol vagy valami ilyesmi. Amelyik nem ‘Data’ kulccsal küld vissza adatot majd lecserélem még, de szerintem most alapokhoz ez jó lesz.

# Architechtúra