VAS VÁRMEGYEI SZAKKÉPZÉSI CENTRUM
HORVÁTH BOLDIZSÁR
KÖZGAZDASÁGI ÉS INFORMATIKAI
TECHNIKUM

A 5 0613 12 03 számú Szoftverfejlesztő és –tesztelő vizsgaremek

Backend dokumentációja

Készítették:

HEGEDÜS JANKA

SALAMON SZINDI

SZABÓ MÁTÉ

SZOMBATHELY

2025

Phyton Flask:

A Flask egy nyílt forráskódú webes keretrendszer, amit Phyton nyelven írtak meg.

A Flask keretrendszert weboldalak, REST API-k gyors fejlesztéséhez. Egyik nagy előnye, hogy könnyen összekapcsolható más Frontend keretrendszerekkel.

A mi weblapunkon ezt arra használtuk, hogy ezzel könnyedén tudjunk képeket feltölteni. Hiszen lehetővé teszi, hogy a felhasználó a saját számítógépéről bárhonnan tud képeket feltölteni, és az elérhető lesz akkor is, ha azt letörli onnan, ahol volt vagy pedig valaki más egy másik eszközről szeretné azt megtekinteni. Mivel ez egy virtuális adatbázison keresztül bekerül a React mappába.

Mikor a képfeltöltés történik, akkor a python kód automatikusan a feltöltött kép neve elé beszúr egy 9számjegyből álló számsort, valamint az aktuális időpontot mikró szekundum pontossággal. Ezekre azért van szükség, hogy biztosítsuk, hogy a lép neve egyedi legyen, ugyanis nem lehet a kettő képnek ugyan az a neve.

```
eredeti_filename = secure_filename(file.filename)

veletlen_szam = str(random.randint(100000000, 999999999))

aktualis_time = datetime.now()

idoformazas = aktualis_time.strftime('%Y%m%d%H%M%S') + f'{aktualis_time.microsecond // 10000:02d}'

filename = f"{veletlen_szam}_{idoformazas}_{eredeti_filename}"

file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
```

1. ábra: Kép nevének egyedivé tétele

Backend:

Az alkalmazás azon része, amely a háttérben fut. Ez kezeli az authentikációt, adatbázist, illetve a teljesíti a frontendtől érkező kéréseket.

Ilyen például a Web API 2 ami Entity Framework 6-ot használ.

Ez úgy történik, hogy a backendben implementálva vannak metódusok, amelyeket a frontend egy végponton keresztül tud elérni.

Leggyakrabban implementált metódusok álltalános leírása:

Adatlekérő metódusok:

Paraméter nélküli Get metódus:
Egy olyan Get eljárás, amely adatot kér a szervertől a végponton keresztül, anélkül, hogy bármit is küldene az adatbázisnak. Ezért az összes adatot visszaadja, ami azon a végponton található.

Felépítése:

- Annotáció: GET
- Útvonal: api/…
- Válasz típusa: pl. (IHttpActionResult)
- Válasz: küldi az objektumot JSON-ként

Paraméteres Get metódus:
Egy olyan Get metódus, amely adatokat kér a szervertő úgy, hogy egy vagy akár több paramétert is küld a szervernek, ami alapján csak egy konkrét adatot is képes visszaadni. Az alábbi felépítési példánál Id alapján szeretnénk megkeresni egy konkrét adatot

Felépítése:

- Annotáció: GET
- Útvonal: api/valami/{id}
- Válasz típusa: pl. (IHttpActionResult)
- Válasz: küldi az objektumot JSON-ként

Leggyakrabban használt státuszkódok az Adatlekérő eljárásokban:

- 200 – OK
- 204 – No Content
- 400 – Bad Request
- 404 – Not Found
- 500 – Internal Server Error

Adatmódosító metódusok:

Post metódus:
Egy olyan eljárás, amelyet leginkább adatok küldésére használunk a végpont felé. Ezt a műveletet használják akkor, mikor egy új adatot szeretnénk létrehozni az adatbázisban.

Felépítése:

- Annotáció: Post
- Útvonal: api/valami
- Body: Adatok küldése általában JSON vagy XML formátumban

Leggyakrabban használt státuszkódok:

- 200 - OK
- 201 - Created
- 400 - Bad Request
- 409 - Conflict
- 500 – Internal Server Error

Patch metódus:
Egy olyan metódus, amely a már meglévő erőforrásokat tudja részlegesen módosítani. A fejlécben megadott adatok alapján tudja beazonosítani, a változtatni kívánt adatokat.

Felépítés:

- Annotáció: Patch
- Útvonal: api/valami/{id}
- Body: a módosítani kívánt részek
- Válasz Tipusa: pl. (IHttpActionResult)
- Válasz: státuszkód

Leggyakrabban használt státuszkódok:

- 200 - OK
- 204 - No Content
- 400 - Bad Request
- 404 – Not Found
- 409 – Conflict
- 500 – Internal Server Error

Put metódus:

Ez egy olyan módosító metódus, amely már a meglévő erőforrásokat tudja teljesen módosítani, miután a fejlécben kapott adatok után beazonosítja az erőforrást. Put eljárás hátránya, hogy a patchel szemben ez nem képes az adatok részleges módosítására. Ezért, ha a felhasználó nem küld valamilyen adatot akkor az null értékre változik meg az adatbázisban. A leggyakrabb státuszkódok megegyeznek, a patch-évelFelépítés

- Annotáció: Put
- Útvonal: api/valami/{id}
- Body: összes adat küldése
- Válasz Tipusa: pl. (IHttpActionResult)
- Válasz: státuszkód

Delete metódus:

Ez egy olyan eljárás, amely arra szolgál, hogy egy adott erőforrást véglegesen eltávolítson a szerverről. A sikeres törléshez szükség van arra, hogy a fejlécbe küldjünk megfelelő adatot/adatokat, ami alapján be lehet azonosítani a megfelelő adatot

Felépítése:

- Annotáció: Patch
- Útvonal: api/valami/{id}
- Válasz Tipusa: pl. (IHttpActionResult)
- Válasz: státuszkód

Leggyakrabban használt státuszkod:

- 200 – OK
- 204 - No Content
- 400 - Bad Request
- 404 – Not Found
- 500 – Internal Server Error

Frontendtől a kérés sikeresen megérkezése, utána feldolgozza a kérésben szereplő információkat, és azokat továbbítja az adatbázis felé, ahonnan egy választ vár vissza, amit státuszkódba csomagolva továbbít a frontend felé.

Státuszkódok öt osztályra bonthatók:

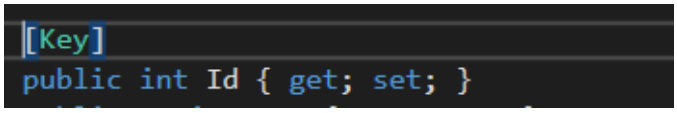
1. tájékoztató: 100-199
2. sikeres: 200-299
3. átirányítás: 300-399
4. ügyfélhiba: 400-499
5. kiszolgáló hiba: 500-599

Leggyakrabban használt státuszkódok összefoglalva:

- 200 - OK: Sikeresen végrehajtódott, a kérés
- 201 - Created: Új elem létrehozása sikerült
- 204 - No Content: A kérés befejeződött, de nem szeretnénk megjeleníteni semmit
- 400 - Bad request: A kérésben szintaktikai hiba található
- 401 - Unauthorized: Hozzáféréshez kötött
- 404 - Not Found: Az erőforrás nem található
- 409 - Conflict: Ütközés az adatbázisban
- 500 - Internal Server Error: probléma történt a szerveren

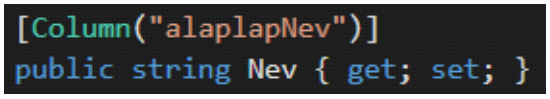
A backendben megtalálhatók modellek, ugyanis ezek fognak megfelelni az adatbázisban szereplő mezőknek.

A modellekben, ha az elsődleges kulcsot nem Id-nak nevezzük, akkor ezt annotálni kell a key kulcs szóval.



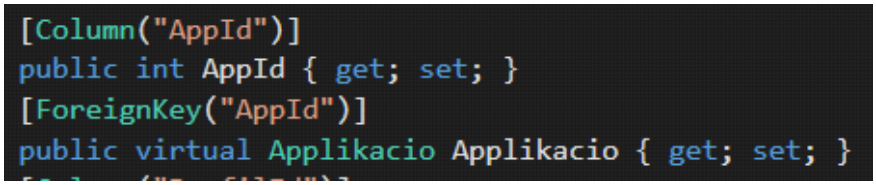
2. ábra: Anotacio

Illetve érdemes a többi mezőnek ugyan azt a nevet adni, mint ami az adatbázisban szerepel, különben azokat pedig egy schemat kell beállítani, ami az adatbázisban szereplő névre mutat.



3. ábra: Schema

Abban a modellben, ahol egy kapcsolat idegen kulcsa található, ott azt is jelölni kell, hogy melyik modell-vel van összekapcsolni:



4. ábra: Idegenkulcsok jelölése

Mint ahogy már korábban szó volt róla, hogy a backendbe vannak implementálva azok az eljárások, amiken szükségesek azok, hogy a kliens oldalról eljussanak a kérések az adatbázishoz. Ezeket a Controllerben helyezik el.

Végpontjaink részeletes bemutatása:

VideókártyaController:

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Videokartya

Kimeneti értékek:

- Nev - típusa: string – Videókártya neve
- alaplapiCsatlakozas – típusa: string - Alaplapi csatlakozó neve
- ajanlottTapegyseg – típusa: int – Ajánlott tápegység watt-ban
- monitorCsatlakozas – típusa: string – Monitor csatlakoztatási lehetőség
- chipGyartoja – típusa: string – Chip gyártójának neve
- vram – típusa: int – Videókártya virtuális memóriája
- kepnev – típusa: string – Feltöltött videókártya kép neve

Használt státuszkó:

- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(VideokartyaModel))]
public IHttpActionResult Get()
{
    IEnumerable<VideokartyaModel> result = null;
    result = ctx.Videokartyak.Select(x => new VideokartyaModel
    {
        Nev = x.Nev,
        alaplapiCsatlakozas = x.AlaplapiCsatlakozas,
        ajanlottTapegyseg = x.AjanlottTapegyseg,
        monitorCsatlakozas = x.MonitorCsatlakozas,
        chipGyartoja = x.ChipGyartoja,
        vram = x.Vram,
        kepnev=x.KepNev
    }).ToList();

    return Ok(result);
}
```

5. ábra: Videókártya paraméter nélküli Get metódus

Paraméteres Get:

Végpontja: https://localhost:44316/api/Videokartya/0?name=&vram=

Bemeneti értékek:

- id – tipusa: int - ez statikus nulla lesz mindig
- name – tipusa: string - Videókártya neve
- vram – tipusa: int – A videókértyához tartozó vram

Kimeneti értékek:

- Nev - típusa: string – Videókártya neve
- alaplapiCsatlakozas – típusa: string - Alaplapi csatlakozó neve
- ajanlottTapegyseg – típusa: int – Ajánlott tápegység watt-ban
- monitorCsatlakozas – típusa: string – Monitor csatlakoztatási lehetőség
- chipGyartoja – típusa: string – Chip gyártójának neve

- vram – típusa: int – Videókártya virtuális memóriája
- kepnev – típusa: string – Feltöltött videókártya kép neve

Használt státuszkód:

- 404 – Not Found (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(VideokartyaModel))]
public IHttpActionResult Get(int id, string name, int vram )
{
    VideokartyaModel result = null;
    result = ctx.Videokartyak.Where(x => x.Nev == name && x.Vram==vram).Select(x => new VideokartyaModel
    {
        Nev = x.Nev,
        alaplapiCsatlakozas = x.AlaplapiCsatlakozas,
        ajanlottTapegyseg = x.AjanlottTapegyseg,
        monitorCsatlakozas = x.MonitorCsatlakozas,
        chipGyartoja = x.ChipGyartoja,
        vram = x.Vram,
        kepnev=x.KepNev
    }).FirstOrDefault();
    if (result == null) return NotFound();
    return Ok(result);
}
```

6. ábra: Videokártya paraméteres Get

Post:

Végpontja: https://localhost:44316/api/Videokartya

Bementi érték:

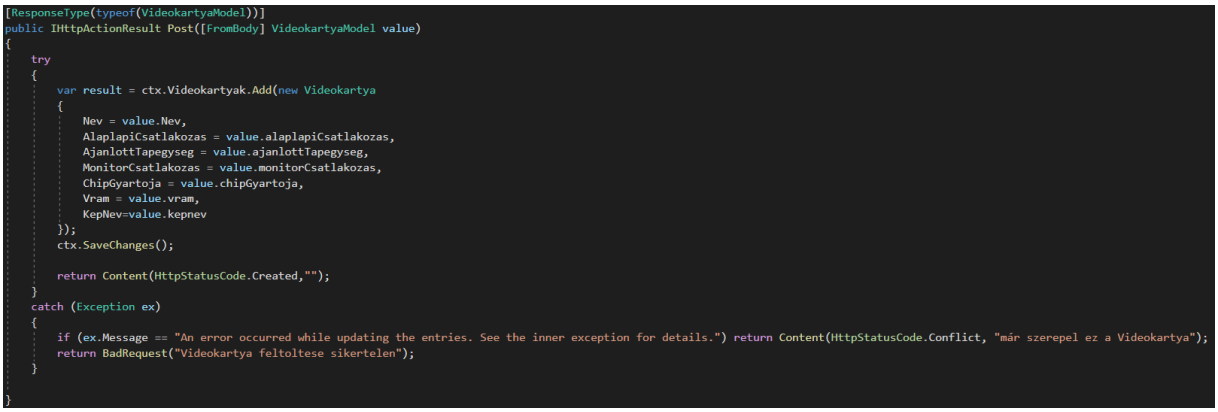
- Nev – tipusa: string - Videókértya neve
- alaplapiCsatlakozas – tipusa: string – Alaplapi csatlakozás neve
- ajanlottTapegyseg – tipusa: int-Ajánlott tápegység watt-ban
- MonitorCsatlakozas - típusa: string – Monitor csatlakoztatási lehetőség
- chipGyartoja – tipus: string - Chip gyártójának neve
- vram – tipusa: int - Videókártya virtuális memóriája
- kepnev – típusa: string – Feltöltött videókártya kép neve

Kimeneti érték:

- Szöveges üzenet

Használt státuszkód:

- 409 – Conflict (Ütközés az adatbázisban)
- 400 – BadRequest (Valamilyen oknál fogva nem sikerül a feltöltés)
- 201 – Created (Sikeres feltöltés)



```
[ResponseType(typeof(VideokartyaModel))]
public IHttpActionResult Post([FromBody] VideokartyaModel value)
{
    try
    {
        var result = ctx.Videokartyak.Add(new Videokartya
        {
            Nev = value.Nev,
            AlaplapiCsatlakozas = value.alaplapiCsatlakozas,
            AjanlottTapegyseg = value.ajanlottTapegyseg,
            MonitorCsatlakozas = value.monitorCsatlakozas,
            ChipGyartoja = value.chipGyartoja,
            Vram = value.vram,
            KepNev=value.kepnev
        });
        ctx.SaveChanges();

        return Content(HttpStatusCode.Created,"");
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.") return Content(HttpStatusCode.Conflict, "már szerepel ez a Videokartya");
        return BadRequest("Videokartya feltoltese sikertelen");
    }
}
```

7. ábra: Videókártya Post metódusa

Patch:

Végpontja: https://localhost:44316/api/Videokartya/1?name=&vram=

Bemeneti érték:

- id – tipusa: int – mindig statikus nulla
- name- tipusa: string - Videókártya neve
- vram - tipusa: int - Videókártya virtuális memóriája
- A modositani kivánt paraméterek

Kimeneti érték:

- Nev - típusa: string – Videókártya neve
- alaplapiCsatlakozas – típusa: string - Alaplapi csatlakozó neve
- ajanlottTapegyseg – típusa: int – Ajánlott tápegység watt-ban
- monitorCsatlakozas – típusa: string – Monitor csatlakoztatási lehetőség
- chipGyartoja – típusa: string – Chip gyártójának neve
- vram – típusa: int – Videókártya virtuális memóriája
- kepnev – típusa: string – Feltöltött videókártya kép neve
- Hiba esetén szöveges üzenet

Használt státuszkód:

- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(VideokartyaModel))]
public IHttpActionResult Patch(int id,string name,int vram, [FromBody] VideokartyaModel value)
{
    try
    {
        var result = ctx.Videokartyak.Where(x => x.Nev == name && x.Vram == vram).FirstOrDefault();
        if (result == null) return NotFound();
        if(value.Nev!=null) result.Nev = value.Nev;
        if (value.alaplapiCsatlakozas != null) result.AlaplapiCsatlakozas = value.alaplapiCsatlakozas;
        if (value.ajanlottTapegyseg != 0) result.AjanlottTapegyseg = value.ajanlottTapegyseg;
        if(value.monitorCsatlakozas!=null) result.MonitorCsatlakozas = value.monitorCsatlakozas;
        if(value.chipGyartoja!=null) result.ChipGyartoja = value.chipGyartoja;
        if(value.vram!=0) result.Vram = value.vram;
        if (value.kepnev != null) result.KepNev = value.kepnev;

        ctx.SaveChanges();
        return Ok(result);
        //return Ok(result);
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.") return Content(HttpStatusCode.Conflict,"üttközé
        return InternalServerError(ex);
    }
}
```

8. ábra: Videókártya Post metódusa

Delete:

Végpontja: https://localhost:44316/api/Videokartya/0?name=&vram=

Bemeneti értékek:

- id – tipusa: int – mindig statikus nulla
- name- tipusa: string - Videókártya neve
- vram – tipusa: int - Videókártya virtuális memóriája

Kimeneti érték:

- Hiba esetén szöveges üzenet

Használt státuszkód:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(VideokartyaModel))]
public IHttpActionResult Delete(int id,string name, int vram)
{
    var vidId = ctx.Videokartyak.Where(x => x.Nev == name && x.Vram == vram).Select(x=>x.Id).FirstOrDefault();
    var set = ctx.Setupok.Where(x => x.VidkaId == vidId).ToList();

    foreach (var item in set)
    {
        item.VidkaId = null;
    }


    var result = ctx.Videokartyak.Where(x => x.Nev == name&& x.Vram==vram).FirstOrDefault();
    if (result!=null)
    {
        ctx.Videokartyak.Remove(result);
        ctx.SaveChanges();
        return Ok("Törlés sikeresen megtörtént");
    }
    ctx.SaveChanges();
    return NotFound();

}
```

9. ábra: Videókártya Delete metódusa

SetupController:

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Setup

Kimeneti értékek:

- ApplikacioNeve – típusa: string - Melyik applikációhoz tartozik
- Tarhely – tipusa: int – Az applikáció mennyi tárhelyet igényel
- Gepigeny – típusa: string – Melyik gépigényes osztályba sorolható pl.(min/opt)
- VideokartyaNev – típusa: string - Setuphoz tartozó Videókártya neve
- VideokartyaVram – típusa: int – Videókártyához tartozó virtuális ram
- ProcesszorNev – típusa: string – Setuphoz tartozó processzor neve
- ProcesszorSzalakSzama – típusa: int – Processzorban található szálak mennyisége
- ProcesszorMagokSzama – típusa: int – Processzorban található magok száma
- ProcesszorFrekvencia – típusa: double – Processzorhoz tartozó GHz érték
- RamNev – típusa: string – Setuphoz tartozó ram neve
- RamFrekvencia – típusa: int – Ramhoz tartozó Hz érték
- RamMeret – típusa: int – Mekkora gyári kiszerelésben található a ram
- OprendszerNev – típusa: string – A setuphoz tartozó operációs rendszer neve
- AlaplapNeve – típusa: string – A setuphoz tartozó alaplap neve
- AlaplapCpuFoglalat – típusa: string – Az alaplapra milyen processzorok kompatibilisek
- AlaplapMemoriaMaxFrekvencia – típusa: double – Az alaplap mekkora Hz-vel rendelkező ramot tud kezelni
- AlaplapRamTipusa – típusa: string – Az alaplapra milyen foglalatú memóriát tud befogadni

Használt státuszkódok:

- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(SetupModel))]
public IHttpActionResult Get()
{
    IEnumerable<SetupModel> result = null;
    result = ctx.Setupok.Include(x => x.Alaplap).Include(x => x.Oprendszer).Include(x => x.Processzor).Include(x => x.Ram)
                        .Include(x => x.Videokartya).Include(x => x.Applikacio).Select(x => new SetupModel
    {
        ApplikacioNeve = x.Applikacio.Nev,
        Gepigeny = x.Gp,
        VidekortyaNev = x.Videokartya.Nev,
        VideokartyaVram = x.Videokartya.Vram,
        ProcesszorNev = x.Processzor.Nev,
        ProcesszorSzalakSzama = x.Processzor.SzalakSzama,
        ProcesszorMagokSzama = x.Processzor.ProcesszormagokSzama,
        ProcesszorFrekvencia = x.Processzor.ProcesszorFrekvencia,
        RamNeve=x.Ram.Nev,
        RamMeret = x.Ram.Meret,
        RamFrekvencia = x.Ram.Frekvencia,
        OprendszerNev = x.Oprendszer.Nev,
        Tarhely = x.Applikacio.Tarhely,
        AlaplapNeve=x.Alaplap.Nev,
        AlaplapCpuFoglalat=x.Alaplap.CpuFoglalat,
        AlaplapMemoriaMaxFrekvencia=x.Alaplap.MaxFrekvencia,
        AlaplapRamTipusa=x.Alaplap.MemoriaTipusa
    }).ToList();
    return Ok(result);
}
```

10. ábra: Setup paraméter nélküli Get metódusa

Paraméteres Get:

Végpontja: https://localhost:44316/api/Setup/0?name=

Bemeneti értékek:

id – tipusa: int – mindig statikus nulla

name – tipusa: string – Alkalmazás neve, amihez keressük a setupot

Kimeneti értékek:

- ApplikacioNeve – típusa: string - Melyik applikációhoz tartozik
- Tarhely – tipusa: int – Az applikáció mennyi tárhelyet igényel
- Gepigeny – típusa: string – Melyik gépigényes osztályba sorolható pl.(min/opt)
- VideokartyaNev – típusa: string - Setuphoz tartozó Videókártya neve
- VideokartyaVram – típusa: int – Videókártyához tartozó virtuális ram
- ProcesszorNev – típusa: string – Setuphoz tartozó processzor neve
- ProcesszorSzalakSzama – típusa: int – Processzorban található szálak mennyisége
- ProcesszorMagokSzama – típusa: int – Processzorban található magok száma
- ProcesszorFrekvencia – típusa: double – Processzorhoz tartozó GHz érték
- RamNev – típusa: string – Setuphoz tartozó ram neve
- RamFrekvencia – típusa: int – Ramhoz tartozó Hz érték
- RamMeret – típusa: int – Mekkora gyári kiszerelésben található a ram

- OprendszerNev – típusa: string – A setuphoz tartozó operációs rendszer neve
- AlaplapNeve – típusa: string – A setuphoz tartozó alaplap neve
- AlaplapCpuFoglalat – típusa: string – Az alaplapra milyen processzorok kompatibilisek
- AlaplapMemoriaMaxFrekvencia – típusa: double – Az alaplap mekkora Hz-vel rendelkező ramot tud kezelni
- AlaplapRamTipusa – típusa: string – Az alaplapra milyen foglalatú memóriát tud befogadni

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 - Ok (Sikerült a kérés)

```
[ResponseType(typeof(SetupModel))]
public IHttpActionResult Get(int id, string name)
{
    IEnumerable<SetupModel> result = null;
    result = ctx.Setupok.Include(x => x.Alaplap).Include(x => x.Oprendszer)
        .Include(x => x.Processzor).Include(x => x.Ram)
        .Include(x => x.Videokartya).Include(x => x.Applikacio)
        .Where(x => x.Applikacio.Nev == name).Select(x => new SetupModel
    {
        ApplikacioNeve = x.Applikacio.Nev,
        Gepigeny = x.Gp,
        VidekortyaNev = x.Videokartya.Nev,
        VideokartyaVram = x.Videokartya.Vram,
        ProcesszorNev = x.Processzor.Nev,
        ProcesszorSzalakSzama = x.Processzor.SzalakSzama,
        ProcesszorMagokSzama = x.Processzor.ProcesszormagokSzama,
        ProcesszorFrekvencia = x.Processzor.ProcesszorFrekvencia,
        RamNeve = x.Ram.Nev,
        RamMeret = x.Ram.Meret,
        RamFrekvencia = x.Ram.Frekvencia,
        OprendszerNev = x.Oprendszer.Nev,
        Tarhely = x.Applikacio.Tarhely,
        AlaplapNeve = x.Alaplap.Nev,
        AlaplapCpuFoglalat = x.Alaplap.CpuFoglalat,
        AlaplapMemoriaMaxFrekvencia = x.Alaplap.MaxFrekvencia,
        AlaplapRamTipusa = x.Alaplap.MemoriaTipusa
    }).ToList();
    if (result == null) return NotFound();
    return Ok(result);
}
```

11. ábra: Setuphoz tartozó paraméteres Get

Post:

Végpontja: https://localhost:44316/api/Setup

Bemeneti értékek

- ApplikacioNeve – típusa: string - Melyik applikációhoz tartozzon
- Gepigeny – típusa: string – Melyik gépigényes osztályba sorolható pl.(min/opt)

- VideokartyaNev – típusa: string - Setuphoz tartozó Videókártya neve
- Vram – típusa: int – Videókártyához tartozó virtuális ram
- ProcesszorNev – típusa: string – Setuphoz tartozó processzor neve
- OprendszerNev – típusa: string – A setuphoz tartozó operációs rendszer neve
- RamNev – típusa: string – Setuphoz tartozó ram neve
- RamFrekvencia – típusa: int – Ramhoz tartozó Hz érték
- RamMeret – típusa: int – Mekkora gyári kiszerelésben található a ram
- AlaplapNeve – típusa: string – A setuphoz tartozó alaplap neve

Kimeneti érték:

- Szöveges üzenet

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat. például. videókártya nev)
- 201 – Created (Sikeres feltöltés)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)

```
[ResponseType(typeof(SetupPostModel))]
public IHttpActionResult Post([FromBody] SetupPostModel value)
{
    var ApplikacioId = ctx.Applikaciok.Where(x => x.Nev == value.ApplikacioNeve)
        .Select(x => x.Id).FirstOrDefault();
    if (ApplikacioId == 0) return Content(HttpStatusCode.NotFound, "Nincs ilyen applikacio");
    var VideokartyaId = ctx.Videokartyak.Where(x => x.Nev == value.VidekortyaNev && x.Vram==value.Vram)
        .Select(x => x.Id).FirstOrDefault();
    if (VideokartyaId == 0) return Content(HttpStatusCode.NotFound, "Nincs ilyen Videokartya");
    var ProcesszorId = ctx.Processzorok.Where(x => x.Nev == value.ProcesszorNev)
        .Select(x => x.Id).FirstOrDefault();
    if (ProcesszorId == 0) return Content(HttpStatusCode.NotFound, "Nincs ilyen Processzor");
    var OprendszerId = ctx.Oprendszerek.Where(x => x.Nev == value.OprendszerNev)
        .Select(x => x.Id).FirstOrDefault();
    if (OprendszerId == 0) return Content(HttpStatusCode.NotFound, "Nincs ilyen Operációs rendszer");
    var RamoId = ctx.Ramok.Where(x => x.Nev == value.RamNeve && x.Frekvencia==value.RamFrekvencia && x.Meret==value.RamMeret)
        .Select(x => x.Id).FirstOrDefault();
    if (RamoId == 0) return Content(HttpStatusCode.NotFound, "Nincs ilyen Ram");
    var AlaplapId = ctx.Alaplapok.Where(x => x.Nev == value.AlaplapNeve)
        .Select(x => x.Id).FirstOrDefault();
    if (AlaplapId == 0) return Content(HttpStatusCode.NotFound, "Nincs ilyen Alaplap");
    try
    {var result = ctx.Setupok.Add(new Setup
        {
            ApplikacioId=ApplikacioId,
            VidkaId=VideokartyaId,
            ProcId=ProcesszorId,
            OpId=OprendszerId,
            RamId=RamoId,
            AlaplId=AlaplapId,
            Gp=value.Gepigeny
        });
        ctx.SaveChanges();
        return Created($"api/Setup/{result}", result);}
    catch (Exception ex) {return InternalServerError(ex);}}
```

12. ábra: Setuphoz tartozo Post

Patch:

Végpontja: https://localhost:44316/api/Setup/0?applikacionev=&igeny=

Bemeneti értékek:

- id – tipusa: int – mindig statikus nulla
- applikacionev – típusa: string – Applikáció neve, aminél szeretnénk megváltoztatni, a setupot
- igeny – típusa: string – A módosítani kívánt setup gépigénye
- A modositani kivánt paraméterek

Kimeneti értékek:

- Alaplap, Applikacio, Kategoria, Oprendszer, Processzor, Ram, Videokartya és a Setup tábla összes tulajdonságát
- Hiba esetén szöveges üzenet

Használt státuszkódok

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat. például. videókártya nev)
- 200 – Ok (Sikeres kérés)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)

```
[ResponseType(typeof(SetupPatchModel))]
public IHttpActionResult Patch(int id,string applikacionev,string igeny, [FromBody] SetupPatchModel value)
{
    try
    {
        var ApplikacioId = ctx.Applikaciok.Where(x => x.Nev == applikacionev)
            .Select(x => x.Id).FirstOrDefault();
        if (ApplikacioId == 0) return Content(HttpStatusCode.NotFound, "Nincs ilyen applikacio");
        var result = ctx.Setupok.Where(x => x.ApplikacioId == ApplikacioId && x.Gp == igeny).FirstOrDefault();
        if (result == null) return NotFound();
        if (value.AlaplapNeve != null) result.AlaplId = ctx.Alaplapok
                .Where(x=>x.Nev==value.AlaplapNeve)
                .Select(x=>x.Id).FirstOrDefault();
        if (value.VidekortyaNev != null) result.VidkaId = ctx.Videokartyak
                .Where(x => x.Nev == value.VidekortyaNev &&x.Vram==value.Vram)
                .Select(x => x.Id).FirstOrDefault();
        if (value.OprendszerNev != null) result.OpId = ctx.Oprendszerek
                .Where(x => x.Nev == value.OprendszerNev)
                .Select(x => x.Id).FirstOrDefault();
        if (value.ProcesszorNev != null) result.ProcId = ctx.Processzorok
                .Where(x => x.Nev == value.ProcesszorNev)
                .Select(x => x.Id).FirstOrDefault();
        if (value.RamNeve != null) result.RamId = ctx.Ramok
                .Where(x => x.Nev == value.RamNeve && x.Frekvencia==value.RamFrekvencia && x.Meret==value.RamMeret)
                .Select(x => x.Id).FirstOrDefault();

        ctx.SaveChanges();
        return Ok(result);
    }
    catch (Exception ex)
    {
        return InternalServerError(ex);
    }
}
```

13. ábra: Setuphoz tartozó Patch

Delete

Végpontja: https://localhost:44316/api/Setup/0?applikacionev=&igeny=

Bemeneti értékek:

- id – tipusa: int – mindig statikus nulla

- applikacionev – típusa: string - Applikáció neve, aminél szeretnénk megváltoztatni, a setupot
- igeny – tipusa: string - A módosítani kívánt setup gépigénye

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikeres kérés)

```
[ResponseType(typeof(SetupModel))]
public IHttpActionResult Delete(int id,string applikacionev,string igeny)
{
    var ApplikacioId = ctx.Applikaciok.Where(x => x.Nev == applikacionev).Select(x => x.Id).FirstOrDefault();
    if (ApplikacioId == 0) return Content(HttpStatusCode.NotFound, "Nincs ilyen applikacio");
    var result = ctx.Setupok.Where(x => x.ApplikacioId == ApplikacioId && x.Gp == igeny).FirstOrDefault();
    if (result != null)
    {
        ctx.Setupok.Remove(result);
        ctx.SaveChanges();
        return Ok("Törlés sikeresen megtörtént");
    }
    ctx.SaveChanges();
    return NotFound();
}
```

14. ábra: Setuphoz tartozó Delete

RamController

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Ram

Kimeneti értékek:

- Nev – típusa: string – Ram neve
- MemoriaTipus – típusa: string – Ram foglalatának típusa
- Frekvencia – típusa: int – Ram sebessége Hz-ben
- Meret – típusa: int - Mekkora gyári kiszerelésben található a ram
- Kepnev – típusa: string – Ramhoz tartozó kép neve

Használt státuszkód:

- 200 – Ok (Sikeres a kérés)

```
[ResponseType(typeof(RamModel))]
1 reference
public IHttpActionResult Get()
{
    IEnumerable<RamModel> result = null;
    result = ctx.Ramok.Select(x => new RamModel
    {
        Nev = x.Nev,
        MemoriaTipus = x.MemoriaTipus,
        Frekvencia = x.Frekvencia,
        Meret = x.Meret,
        Kepnev=x.KepNev
    }).ToList();
    return Ok(result);
}
```

15. ábra: Ramhoz tartozó paraméter nélküli Get

Paraméteres Get:

Végpontja: https://localhost:44316/api/Ram/0?name=&frekvencia=&meret=

Bemeneti értékek:

- id – tipusa: int – mindig statikus nulla
- name – típusa: string – Ram neve
- frekvencia – típusa: int - Ram sebessége Hz-ben
- meret – típusa: int - Mekkora gyári kiszerelésben található a ram

Kimeneti értékek:

- Nev – típusa: string – Ram neve
- MemoriaTipus – típusa: string – Ram foglalatának típusa
- Frekvencia – típusa: int – Ram sebessége Hz-ben
- Meret – típusa: int - Mekkora gyári kiszerelésben található a ram
- Kepnev – típusa: string – Ramhoz tartozó kép neve

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikeres a kérés)

```
[ResponseType(typeof(RamModel))]
1 reference
public IHttpActionResult Get(int id, string name, int frekvencia, int meret)
{
    RamModel result = null;
    result = ctx.Ramok.Where(x => x.Nev == name && x.Frekvencia==frekvencia&& x.Meret==meret)
        .Select(x => new RamModel
    {
        Nev = x.Nev,
        MemoriaTipus = x.MemoriaTipus,
        Frekvencia = x.Frekvencia,
        Meret = x.Meret,
        Kepnev = x.KepNev
    }).FirstOrDefault();
    if (result == null) return NotFound();
    return Ok(result);
}
```

16. ábra:Ramhoz tartozó paraméteres Get

Post:

Végpontja: https://localhost:44316/api/Ram

Bemeneti értékek:

- Nev – típusa: string – Ram neve
- MemoriaTipus – típusa: string – Ram foglalatának típusa
- Frekvencia – típusa: int – Ram sebessége Hz-ben
- Meret – típusa: int - Mekkora gyári kiszerelésben található a ram
- Kepnev – típusa: string – Ramhoz tartozó kép neve

Kimeneti értékek:

- Nev – típusa: string – Ram neve
- MemoriaTipus – típusa: string – Ram foglalatának típusa
- Frekvencia – típusa: int – Ram sebessége Hz-ben
- Meret – típusa: int - Mekkora gyári kiszerelésben található a ram
- Kepnev – típusa: string – Ramhoz tartozó kép neve
- Hiba esetén szöveges üzenet

Használt státuszkódok:

- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 201 – Created (Sikeres feltöltés)

```
[ResponseType(typeof(RamModel))]
1 reference
public IHttpActionResult Post([FromBody] RamModel value)
{
    try
    {
        var result = ctx.Ramok.Add(new Ram
        {
            Nev = value.Nev,
            MemoriaTipus=value.MemoriaTipus,
            Frekvencia=value.Frekvencia,
            Meret=value.Meret,
            KepNev=value.Kepnev
        });
        ctx.SaveChanges();

        return Created($"api/Ram/{result}", result);
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "már szerepel ez a ram");
        return InternalServerError(ex);
    }

}
```

17. ábra: Ramhoz tartozó Post

Patch:

Végpontja: https://localhost:44316/api/Ram/0?name=&frekvencia=&meret=

Bemeneti értékek:

- id – tipusa: int – mindig statikus nulla
- name – típusa: string – Ram neve
- frekvencia – típusa: int - Ram sebessége Hz-ben
- meret – típusa: int - Mekkora gyári kiszerelésben található a ram
- A modositani kivánt paraméterek

Kimeneti értékek:

- Szöveges megjegyzés:

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 200 – Ok (Sikeres kérés)

```
[ResponseType(typeof(RamModel))]
1 reference
public IHttpActionResult Patch(int id, string name,int frekvencia,int meret,[FromBody] RamModel value)
{
    try
    {
        var result = ctx.Ramok.Where(x => x.Nev == name && x.Frekvencia == frekvencia && x.Meret == meret).FirstOrDefault();
        if (result == null) return NotFound();
        if(value.Nev!=null) result.Nev = value.Nev;
        if (value.MemoriaTipus!=null) result.MemoriaTipus = value.MemoriaTipus;
        if (value.Frekvencia!=0) result.Frekvencia = value.Frekvencia;
        if (value.Meret!=0) result.Meret = value.Meret;
        if (value.Kepnev != null) result.KepNev = value.Kepnev;

        ctx.SaveChanges();
        return Content(HttpStatusCode.OK, "sikeres update");
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict,"Már létezik ez a ram");
        return InternalServerError(ex);
    }
}
```

18. ábra: Ramhoz tartozó Patch

Delete:

Végpontja: https://localhost:44316/api/Ram/0?name=&frekvencia=&meret=

Bemeneti értékek:

- id – tipusa: int – mindig statikus nulla
- name – típusa: string – Ram neve
- frekvencia – típusa: int - Ram sebessége Hz-ben
- meret – típusa: int - Mekkora gyári kiszerelésben található a ram

Kimeneti érték:

- Szöveges megjegyzés

Használt státuszkodok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikeres kérés)

```
[ResponseType(typeof(RamModel))]
1 reference
public IHttpActionResult Delete(int id,string name, int frekvencia, int meret)
{
    var ramId = ctx.Ramok.Where(x=>x.Nev==name && x.Frekvencia==frekvencia && x.Meret==meret)
        .Select(x => x.Id).FirstOrDefault();
    var set = ctx.Setupok.Where(x => x.RamId == ramId).ToList();

    foreach (var item in set)
    {
        item.RamId = null;
    }

    var result = ctx.Ramok.Where(x => x.Nev == name && x.Frekvencia == frekvencia && x.Meret == meret)
        .FirstOrDefault();
    if (result != null)
    {
        ctx.Ramok.Remove(result);
        ctx.SaveChanges();
        return Ok("Törlés sikeres");
    }
    ctx.SaveChanges();
    return NotFound();
}
```

19. ábra: Ramhoz tartozó Delete

ProfilController

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Profil

Kimeneti értékek:

- Id – típusa: int -Felhasználó egyedi azonosítója
- Felhasznalonev – típusa: string – Felhasználó név
- Email – típusa: string – Felhasználó email címe
- Jogosultsag – típusa: int – Jogosultsági szint
- Tema – típus: string – Felhasználó által választott háttérszín
- LogoEleresiUtja – típus: string – Felhasználó profilképe

Használt státuszkód:

- 200 – Ok (Sikeres kérés)

```
[ResponseType(typeof(ProfilResponseModel))]
1 reference
public IHttpActionResult Get()
{
    IEnumerable<ProfilResponseModel> result = null;

    result = ctx.Profilok.Select(x => new ProfilResponseModel
    {
        Id=x.Id,
        Felhasznalonev = x.Felhasznalonev,
        Email = x.Email,
        Jogosultsag = x.Jogosultsag,
        Tema = x.Tema,
        LogoEleresiUtja = x.LogoEleresiUtja
    }).ToList();
    return Ok(result);
}
```

20. ábra: Profilhoz tartozó paraméter nélküli Get

Paraméteres Get:

Végpontja: https://localhost:44316/api/Profil/0?name=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Felhasználónév

Kimeneti értékek:

- Id – típusa: int -Felhasználó egyedi azonosítója
- Felhasznalonev – típusa: string – Felhasználó név
- Email – típusa: string – Felhasználó email címe
- Jogosultsag – típusa: int – Jogosultsági szint
- Tema – típus: string – Felhasználó által választott háttérszín
- LogoEleresiUtja – típus: string – Felhasználó profilképe

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikeres kérés)

```
[ResponseType(typeof(ProfilResponseModel))]
1 reference
public IHttpActionResult Get(int id, string name)
{
    ProfilResponseModel result = null;

    result = ctx.Profilok.Where(x => x.Felhasznalonev == name).Select(x => new ProfilResponseModel
    {
        Id=x.Id,
        Felhasznalonev = x.Felhasznalonev,
        Email = x.Email,
        Jogosultsag = x.Jogosultsag,
        Tema = x.Tema,
        LogoEleresiUtja = x.LogoEleresiUtja
    }).FirstOrDefault();
    if (result == null) return NotFound();
    return Ok(result);
}
```

21. ábra: Profilhoz tartozó paraméteres Get

Post

Végpontja: https://localhost:44316/api/Profil

Bemeneti értékek:

- Felhasznalonev – típusa: string – Felhasználó név
- Email – típusa: string – Felhasználó email címe
- Jogosultsag – típusa: int – Jogosultsági szint
- Tema – típus: string – Felhasználó által választott háttérszín
- LogoEleresiUtja – típus: string – Felhasználó profilképe

Kimeneti érték:

- Szöveges megjegyzés

Használt státuszkódok:

- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 201 – (Sikeres feltölté)

```
[ResponseType(typeof(ProfilResponseModel))]
1 reference
public IHttpActionResult Post([FromBody] ProfilPostModel value)
{
    try
    {
        var email = ctx.Profilok.Where(x => x.Email == value.Email).FirstOrDefault();
        var felhasz = ctx.Profilok.Where(x => x.Felhasznalonev == value.Felhasznalonev).FirstOrDefault();
        if (email != null) return Content(HttpStatusCode.Conflict, "Ezzel az email-lal már regisztráltak");
        if (felhasz != null) return Content(HttpStatusCode.Conflict, "Ezzel az email-lal már regisztráltak");

        ctx.Profilok.Add(new Profil(value.Felhasznalonev, value.Email, value.Jelszo, value.Jogosultsag, value.Tema, value.LogoEleresiUtja));
        ctx.SaveChanges();
        return Content(HttpStatusCode.Created,"Regisztráció Sikeres!");
    }
    catch (Exception ex)
    {
        return InternalServerError(ex);
    }
}
```

22. ábra: Profilhoz tartozó Post

Patch:

Végpontja: https://localhost:44316/api/Profil/0?name=

Bementi értékek:

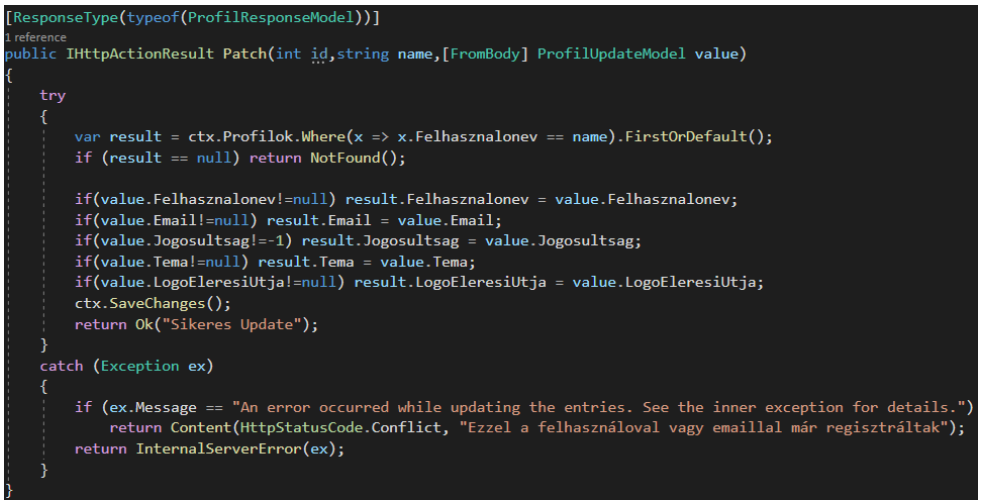
- id – típusa: int – mindig statikus nulla
- name – típusa: string – Felhasználónév
- A modositani kivánt paraméterek

Kimeneti érték:

- Szöveges megjegyzés

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 200 – Ok (Sikerült a kérés)



23. ábra: Profilhoz tartozó Patch

Delete:

Végpontja: https://localhost:44316/api/Profil/0?name=

Bemeneti értékek:

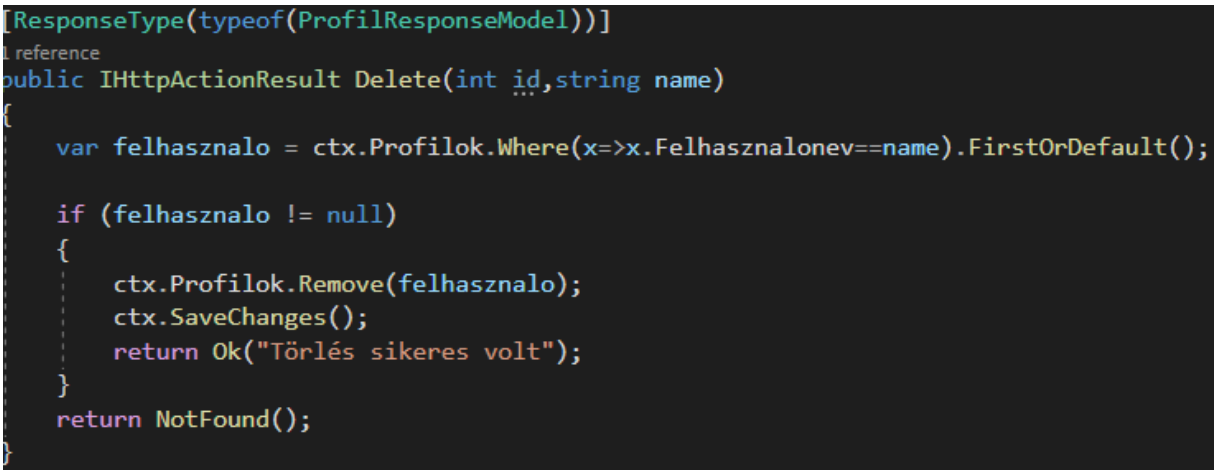
- id – típusa: int – mindig statikus nulla
- name – típusa: string – Felhasználónév

Kimeneti érték:

- Szöveges megjegyzés

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)



24. ábra: Profilhoz tartozó Delete

Authenticate:

Végpontja: https://localhost:44316/api/Profil/Authenticate

Bemeneti értékek:

- Email – típusa: string – Felhasználó email címe
- Jelszo – típusa: string – Felhasználó jelszava

Kimeneti értékek:

- Id – típusa: int -Felhasználó egyedi azonosítója
- Felhasznalonev – típusa: string – Felhasználó név
- Email – típusa: string – Felhasználó email címe
- Jelszo – típusa: string – Felhasználó titkosított jelszava
- JelszoUjra – típusa: string – Felhasználó titkosított jelszava máshogy
- Jogosultsag – típusa: int – Jogosultsági szint
- Tema – típus: string – Felhasználó által választott háttérszín
- LogoEleresiUtja – típus: string – Felhasználó profilképe

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 401 – Unauthorized (Hitelesítés sikertelen)
- 200 – Ok (Sikerült a kérés)

```
[HttpPost]
[Route("api/Profil/Authenticate")]
[ResponseType(typeof(ProfilResponseModel))]
1 reference
public IHttpActionResult Post([FromBody] Authenticate value)
{
    var res = ctx.Profilok.Where(x => x.Email == value.Email).FirstOrDefault();
    if (res!=null)
    {
        var validate = PasswdManager.VerifyPasswordHash(value.Jelszo, res.JelszoUjra, res.Jelszo);
        if (validate) return Ok(res);
        else return Unauthorized();
    }
    return NotFound();
}
```

25. ábra: Profilhoz tartozó Authentication

PatchJelszo:

Végpontja: https://localhost:44316/api/Profil/ProfilJelszoUpdateModel?id=0&email=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- email – típusa: string – Felhasználó email címe
- UjJelszo – típusa: string – Felhasználó új jelszava

Kimeneti értékek:

- Szöveges üzenet

Használt státuszkód

- 401 - Unauthorized (Hitelesítés sikertelen)
- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 200 – Ok (Sikerült a kérés)

```
[HttpPatch]
[Route("api/Profil/ProfilJelszoUpdateModel")]
[ResponseType(typeof(ProfilResponseModel))]
1 reference
public IHttpActionResult PatchJelszo(int id, string email, [FromBody] ProfilJelszoUpdateModel value)
{
    try
    {
        var result = ctx.Profilok.Where(x => x.Email == email).FirstOrDefault();
        if (result == null) return Content(HttpStatusCode.NotFound,"Nem található felhasználó ezzel az emaillal");
        if (!PasswdManager.VerifyEmail(email, result.Email)) return Unauthorized();

        if (value.UjJelszo != null)
        {
            PasswdManager.CreatePasswordHash(value.UjJelszo, out byte[] hash, out byte[] salt);
            result.Jelszo = salt;
            result.JelszoUjra = hash;
        }
        ctx.SaveChanges();

        return Ok("Sikeres jelszo modositas");
    }
    catch (Exception ex)
    {
        return InternalServerError(ex);
    }
}
```

26. ábra: Profilhoz tartozó PatchJelszo

ProcesszorController:

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Processzor

Kimeneti értékek:

- Nev – típusa: string – Processzor neve
- AlaplapFoglalat – típusa: string – Támogatott alaplapfoglalat típusa
- SzalakSzama – típusa: int – Processzorban található szálak száma
- TamogatottMemoriatipus – típusa: string – Milyen memoria tipust támogat
- ProcesszormagokSzama – típusa: int – Processzorban található magok száma
- ProcesszorFrekvencia – típusa double - Processzorhoz tartozó GHz érték
- BProcesszorFrekvencia – típusa: double - Processzorhoz tartozó boostolt GHz érték
- Gyarto – típusa: string – Processzort gyártó cég neve
- AjanlottTapegyseg – típusa: int - Ajánlott tápegység watt-ban
- IntegraltVideokartya – típusa: bool – Rendelkezik-e integrált videókártyával

- Kepnev – típusa: string – Processzorhoz tartozó kép

Használt státuszkód:

- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(ProcesszorModel))]
1 reference
public IHttpActionResult Get()
{
    IEnumerable<ProcesszorModel> result = null;

    result = ctx.Processzorok.Select(x => new ProcesszorModel
    {
        Nev = x.Nev,
        AlaplapFoglalat = x.AlaplapFoglalat,
        SzalakSzama = x.SzalakSzama,
        TamogatottMemoriatipus = x.TamogatottMemoriatipus,
        ProcesszormagokSzama = x.ProcesszormagokSzama,
        ProcesszorFrekvencia = x.ProcesszorFrekvencia,
        BProcesszorFrekvencia=x.BFrekvencia,
        Gyarto = x.Gyarto,
        AjanlottTapegyseg = x.AjanlottTapegyseg,
        IntegraltVideokartya = x.IntegraltVideokartya,
        Kepnev=x.KepNev

    }).ToList();

    return Ok(result);
}
```

27. ábra: Processzorhoz tartozó paraméter nélküli Get

Paraméteres Get:

Végpontja: https://localhost:44316/api/Processzor/0?name=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Processzor neve

Kimenti értékek:

- Nev – típusa: string – Processzor neve
- AlaplapFoglalat – típusa: string – Támogatott alaplapfoglalat típusa
- SzalakSzama – típusa: int – Processzorban található szálak száma
- TamogatottMemoriatipus – típusa: string – Milyen memoria tipust támogat
- ProcesszormagokSzama – típusa: int – Processzorban található magok száma
- ProcesszorFrekvencia – típusa double - Processzorhoz tartozó GHz érték
- BProcesszorFrekvencia – típusa: double - Processzorhoz tartozó boostolt GHz érték
- Gyarto – típusa: string – Processzort gyártó cég neve
- AjanlottTapegyseg – típusa: int - Ajánlott tápegység watt-ban
- IntegraltVideokartya – típusa: bool – Rendelkezik-e integrált videókártyával
- Kepnev – típusa: string – Processzorhoz tartozó kép

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(ProcesszorModel))]
1 reference
public IHttpActionResult Get(int id, string name)
{
    ProcesszorModel result = null;

    result = ctx.Processzorok.Where(x => x.Nev == name).Select(x => new ProcesszorModel
    {
        Nev = x.Nev,
        AlaplapFoglalat = x.AlaplapFoglalat,
        SzalakSzama = x.SzalakSzama,
        TamogatottMemoriatipus = x.TamogatottMemoriatipus,
        ProcesszormagokSzama = x.ProcesszormagokSzama,
        ProcesszorFrekvencia = x.ProcesszorFrekvencia,
        BProcesszorFrekvencia = x.BFrekvencia,
        Gyarto = x.Gyarto,
        AjanlottTapegyseg = x.AjanlottTapegyseg,
        IntegraltVideokartya = x.IntegraltVideokartya,
        Kepnev = x.KepNev
    }).FirstOrDefault();
    if (result == null) return NotFound();
    return Ok(result);
}
```

28. ábra: Processzorhoz tartozó paraméteres Get

Post:

Végpontja: https://localhost:44316/api/Processzor

Bementi értékek:

- Nev – típusa: string – Processzor neve
- AlaplapFoglalat – típusa: string – Támogatott alaplapfoglalat típusa
- SzalakSzama – típusa: int – Processzorban található szálak száma
- TamogatottMemoriatipus – típusa: string – Milyen memoria tipust támogat
- ProcesszormagokSzama – típusa: int – Processzorban található magok száma
- ProcesszorFrekvencia – típusa double - Processzorhoz tartozó GHz érték
- BProcesszorFrekvencia – típusa: double - Processzorhoz tartozó boostolt GHz érték
- Gyarto – típusa: string – Processzort gyártó cég neve
- AjanlottTapegyseg – típusa: int - Ajánlott tápegység watt-ban
- IntegraltVideokartya – típusa: bool – Rendelkezik-e integrált videókártyával
- Kepnev – típusa: string – Processzorhoz tartozó kép

Kimeneti érték:

- Nev – típusa: string – Processzor neve
- AlaplapFoglalat – típusa: string – Támogatott alaplapfoglalat típusa
- SzalakSzama – típusa: int – Processzorban található szálak száma
- TamogatottMemoriatipus – típusa: string – Milyen memoria tipust támogat
- ProcesszormagokSzama – típusa: int – Processzorban található magok száma
- ProcesszorFrekvencia – típusa double - Processzorhoz tartozó GHz érték
- BProcesszorFrekvencia – típusa: double - Processzorhoz tartozó boostolt GHz érték
- Gyarto – típusa: string – Processzort gyártó cég neve
- AjanlottTapegyseg – típusa: int - Ajánlott tápegység watt-ban
- IntegraltVideokartya – típusa: bool – Rendelkezik-e integrált videókártyával
- Kepnev – típusa: string – Processzorhoz tartozó kép
- Hiba esetén szöveges üzenet

Használt státuszkódok:

- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 201 – Created (Sikeres feltöltés)

```
[ResponseType(typeof(ProcesszorModel))]
1 reference
public IHttpActionResult Post([FromBody] ProcesszorModel value)
{
    try
    {
        var result = ctx.Processzorok.Add(new Processzor
        {
            Nev = value.Nev,
            AlaplapFoglalat= value.AlaplapFoglalat,
            SzalakSzama = value.SzalakSzama,
            TamogatottMemoriatipus = value.TamogatottMemoriatipus,
            ProcesszormagokSzama=value.ProcesszormagokSzama,
            ProcesszorFrekvencia=value.ProcesszorFrekvencia,
            BFrekvencia=value.BProcesszorFrekvencia,
            Gyarto=value.Gyarto,
            AjanlottTapegyseg=value.AjanlottTapegyseg,
            IntegraltVideokartya=value.IntegraltVideokartya,
            KepNev=value.Kepnev

        });
        ctx.SaveChanges();

        return Content(HttpStatusCode.Created, result);
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "Ezzel a névvel már létezik processzor");
        return InternalServerError(ex);
    }
```

29. ábra: Processzorhoz tartozó Post

Patch

Végpontja: https://localhost:44316/api/Processzor/0?name=

Bementi értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Processzor neve
- A modositani kivánt paraméterek

Kimeneti érték:

- Szöveges üzenet

Használt státuszkódok:

- 404 – NotFound(nem található a bemeneti értékeknek megfelelő adat)
- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(ProcesszorModel))]
1 reference
public IHttpActionResult Patch(int id, string name, [FromBody] ProcesszorModel value)
{
    try
    {
        var result = ctx.Processzorok.Where(x => x.Nev == name).FirstOrDefault();
        if (result == null) return NotFound();
        if(value.Nev!=null) result.Nev = value.Nev;
        if (value.AlaplapFoglalat!=null) result.AlaplapFoglalat = value.AlaplapFoglalat;
        if (value.SzalakSzama!=0) result.SzalakSzama = value.SzalakSzama;
        if (value.TamogatottMemoriatipus!=null) result.TamogatottMemoriatipus = value.TamogatottMemoriatipus;
        if (value.ProcesszormagokSzama!=0) result.ProcesszormagokSzama = value.ProcesszormagokSzama;
        if (value.ProcesszorFrekvencia!=0) result.ProcesszorFrekvencia = value.ProcesszorFrekvencia;
        if (value.BProcesszorFrekvencia!=0) result.BFrekvencia = value.BProcesszorFrekvencia;
        if(value.Gyarto!=null) result.Gyarto = value.Gyarto;
        if (value.AjanlottTapegyseg!=0) result.AjanlottTapegyseg = value.AjanlottTapegyseg;
        if (value.IntegraltVideokartya!=null) result.IntegraltVideokartya = value.IntegraltVideokartya;
        if (value.Kepnev != null) result.KepNev = value.Kepnev;

        ctx.SaveChanges();
        return Ok("Sikeres Update");
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "Ezzel a névvel már létezik processzor");
        return InternalServerError(ex);
    }
}
```

30. ábra: Processzorhoz tartozó Patch

Delete:

Végpontja: https://localhost:44316/api/Processzor/0?name=

Bementi értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Processzor neve

Kimeneti érték:

- Szöveges üzenet

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(ProcesszorModel))]
1 reference
public IHttpActionResult Delete(int id, string name)
{
    var ProcId = ctx.Processzorok.Where(x => x.Nev == name).Select(x => x.Id).FirstOrDefault();
    var set = ctx.Setupok.Where(x => x.ProcId == ProcId).ToList();

    foreach (var item in set)
    {
        item.ProcId = null;
    }

    var result = ctx.Processzorok.Where(x => x.Nev == name).FirstOrDefault();
    if (result != null)
    {
        ctx.Processzorok.Remove(result);
        ctx.SaveChanges();
        return Ok("Törlés sikeresen véghezment");
    }
    ctx.SaveChanges();
    return NotFound();
}
```

31. ábra: Processzorhoz tartozó Delete

OprendszerController

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Oprendszer

Kimeneti értékek:

- Nev – típusa: string – Operációs rendszer neve
- BuildSzam – típusa: string – Operációs rendszer buildszáma
- Verzio – típusa: string – Operációs rendszer Verziója
- KepNev – típusa: string – Operációs rendszer logója

Használt státuszkód:

- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(OprendszerModel))]
1 reference
public IHttpActionResult Get()
{
    IEnumerable<OprendszerModel> result = null;

    result = ctx.Oprendszerek.Select(x => new OprendszerModel
    {
        Nev = x.Nev,
        BuildSzam = x.BuildSzam,
        Verzio = x.Verzio,
        KepNev=x.KepNev
    }).ToList();

    return Ok(result);
}
```

32. ábra: Oprendszer paraméter nélküli get

Paraméteres Get:

Végpontja: https://localhost:44316/api/Oprendszer/0?name=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Operációs rendszer neve

Kimeneti értékek:

- Nev – típusa: string – Operációs rendszer neve
- BuildSzam – típusa: string – Operációs rendszer buildszáma
- Verzio – típusa: string – Operációs rendszer Verziója
- KepNev – típusa: string – Operációs rendszer logója

Használt státuszkódok:

- 404 – NotFound(nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(OprendszerModel))]
1 reference
public IHttpActionResult Get(int id, string name)
{
    OprendszerModel result = null;

    result = ctx.Oprendszerek.Where(x => x.Nev == name).Select(x => new OprendszerModel
    {
        Nev = x.Nev,
        BuildSzam = x.BuildSzam,
        Verzio = x.Verzio,
        KepNev=x.KepNev
    }).FirstOrDefault();
    if (result == null) return NotFound();
    return Ok(result);
}
```

33. ábra: Oprendszerhez tartozó paraméteres get

Post:

Végpontja: https://localhost:44316/api/Oprendszer

Bementi értékek:

- Nev – típusa: string – Operációs rendszer neve
- BuildSzam – típusa: string – Operációs rendszer buildszáma
- Verzio – típusa: string – Operációs rendszer Verziója
- KepNev – típusa: string – Operációs rendszer logója

Kimeneti értékek:

- Nev – típusa: string – Operációs rendszer neve
- BuildSzam – típusa: string – Operációs rendszer buildszáma
- Verzio – típusa: string – Operációs rendszer Verziója
- KepNev – típusa: string – Operációs rendszer logója
- Hiba esetén szöveges üzenet

Használt státuszkódok:

- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 201 – Created (Sikerült a feltöltés)

```
[ResponseType(typeof(OprendszerModel))]
1 reference
public IHttpActionResult Post([FromBody] OprendszerModel value)
{
    try
    {
        var result = ctx.Oprendszerek.Add(new Operaciosrendszer
        {
            Nev = value.Nev,
            BuildSzam=value.BuildSzam,
            Verzio=value.Verzio,
            KepNev=value.KepNev
        });
        ctx.SaveChanges();

        return Content(HttpStatusCode.Created, result);
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "Ezzel a névvel és buildszámmal már létezik Operációs rendszer.");
        return InternalServerError(ex);
    }

}
```

34. ábra: Oprendszerhez tartozó Post

Patch:

Végpontja: https://localhost:44316/api/Oprendszer/0?name=&buildszam=

Bementi értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Operációs rendszer neve
- buildszam - típusa: string – Operációs rendszer buildszáma
- A modositani kivánt paraméterek

Kimeneti érték:

- Szöveges üzenet

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(OprendszerModel))]
1 reference
public IHttpActionResult Patch(int id, string name,string buildszam, [FromBody] OprendszerModel value)
{
    try
    {
        var result = ctx.Oprendszerek.Where(x => x.Nev == name && x.BuildSzam==buildszam).FirstOrDefault();
        if (result == null) return NotFound();
        if (value.Nev != null) result.Nev = value.Nev;
        if (value.BuildSzam != null) result.BuildSzam = value.BuildSzam;
        if (value.Verzio != null) result.Verzio = value.Verzio;
        if (value.KepNev != null) result.KepNev = value.KepNev;

        ctx.SaveChanges();
        return Ok("Update sikeres");
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "Ezzel a névvel és buildszámmal már létezik Operációs rendszer.");
        return InternalServerError(ex);
    }
}
```

35. ábra: Oprendszerhez tartozó Patch

Delete:

Végpontja: https://localhost:44316/api/Oprendszer/0?name=&buildszam=

Bementi értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Operációs rendszer neve
- buildszam - típusa: string – Operációs rendszer buildszáma

Kimeneti érték:

- Szöveges üzenet

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(OprendszerModel))]
1 reference
public IHttpActionResult Delete(int id, string name, string buildszam)
{
    var OpId = ctx.Oprendszerek.Where(x => x.Nev == name && x.BuildSzam==buildszam).Select(x => x.Id).FirstOrDefault();
    var set = ctx.Setupok.Where(x => x.OpId == OpId).ToList();

    foreach (var item in set)
    {
        item.OpId = null;
    }

    var result = ctx.Oprendszerek.Where(x => x.Nev == name).FirstOrDefault();
    if (result != null)
    {
        ctx.Oprendszerek.Remove(result);
        ctx.SaveChanges();
        return Ok("Törlés sikeresen véghezment");
    }
    ctx.SaveChanges();
    return NotFound();
}
```

36. ábra: Oprendszerhez tartozó Delete

KategoriaController

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Kategoria

Kimeneti érték:

- Nev – típusa: string – Kategória neve

Használt státuszkód:

- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(KategoriaModel))]
1 reference
public IHttpActionResult Get()
{
    IEnumerable<KategoriaModel> result = null;
    result = ctx.Kategoriak.Select(x => new KategoriaModel
    {
        Nev = x.Nev
    }).ToList();
    return Ok(result);
}
```

37. ábra: Kategoriához tartozó paraméter nélküli Get

Paraméteres Get:

Végpontja: https://localhost:44316/api/Kategoria/0?name=

Bementi értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Kategória neve

Kimeneti érték:

- Nev – típusa: string – Kategória neve

Használt státuszkód:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(KategoriaModel))]
1 reference
public IHttpActionResult Get(int id, string name)
{
    KategoriaModel result = null;
    result = ctx.Kategoriak.Where(x => x.Nev == name).Select(x => new KategoriaModel
    {
        Nev = x.Nev
    }).FirstOrDefault();
    if (result == null) return NotFound();
    return Ok(result);
}
```

38. ábra: Kategóriához tartozó paraméteres Get

Post:

Végpontja: https://localhost:44316/api/Kategoria

Bemeneti érték:

- Nev – típusa: string – Kategória neve

Kimeneti értékek:

- Nev – típusa: string – Kategória neve
- Hiba esetén szöveges üzenet

Használt státuszkódok:

- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 201 – Created (Sikerült a feltöltés)

```
[ResponseType(typeof(KategoriaModel))]
1 reference
public IHttpActionResult Post([FromBody] KategoriaModel value)
{
    try
    {
        var result = ctx.Kategoriak.Add(new Kategoria
        {
            Nev=value.Nev
        });
        ctx.SaveChanges();

        return Content(HttpStatusCode.Created, result);
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "Ezzel a névvel már létezik Kategoria rendszer.");
        return InternalServerError(ex);
    }

}
```

39. ábra: Kategóriához tartozó Post

Patch

Végpontja: https://localhost:44316/api/Kategoria/0?name=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Kategória neve

Kimeneti érték:

- Szöveges üzenet

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(KategoriaModel))]
1 reference
public IHttpActionResult Patch(int id, string name, [FromBody] KategoriaModel value)
{
    try
    {
        var result = ctx.Kategoriak.Where(x => x.Nev == name).FirstOrDefault();
        if (result == null) return NotFound();
        if (value.Nev != null) result.Nev = value.Nev;

        ctx.SaveChanges();
        return Ok("Update sikeres");
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "Ezzel a névvel már létezik Kategoria rendszer.");
        return InternalServerError(ex);
    }
}
```

40. ábra:Kategóriához tartozó Patch

Delete:

Végpontja: https://localhost:44316/api/Kategoria/0?name=

Bemeneti értékek?

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Kategória neve

Kimeneti érték:

- Szöveges üzenet

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(KategoriaModel))]
1 reference
public IHttpActionResult Delete(int id, string name)
{
    var KatdId = ctx.Kategoriak.Where(x => x.Nev == name).Select(x => x.Id).FirstOrDefault();
    var set = ctx.Applikaciok.Where(x => x.KatId == KatdId).ToList();

    foreach (var item in set)
    {
        item.KatId = null;
    }

    var result = ctx.Kategoriak.Where(x => x.Nev == name).FirstOrDefault();
    if (result != null)
    {
        ctx.Kategoriak.Remove(result);
        ctx.SaveChanges();
        return Ok("Törlés sikeresen véghezment");
    }
    ctx.SaveChanges();
    return NotFound();
}
```

41. ábra: Kategóriához tartozó Delete

CsatlakozoController:

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Csatlakozo

Kimenti Érték:

- Nev – típusa: string – Csatlakozó neve

Használt státuszkódok:

- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(CsatlakozoModel))]
1 reference
public IHttpActionResult Get()
{
    IEnumerable<CsatlakozoModel> result = null;

    result = ctx.Csatlakozok.Select(x => new CsatlakozoModel
    {
        Nev = x.Nev
    }).ToList();

    return Ok(result);
}
```

42. ábra: Csatlakozóhoz tartozó paraméter nélküli Get

Paraméteres Get:

Végpontja: https://localhost:44316/api/Csatlakozo/0?name=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Csatlakozó neve

Kimeneti érték:

- Nev – típusa: string – Csatlakozó neve

Használt státuszkódok:

- NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(CsatlakozoModel))]
1 reference
public IHttpActionResult Get(int id, string name)
{
    CsatlakozoModel result = null;

    result = ctx.Csatlakozok.Where(x => x.Nev == name).Select(x => new CsatlakozoModel
    {
        Nev = x.Nev
    }).FirstOrDefault();
    if (result == null) return NotFound();
    return Ok(result);
}
```

43. ábra: Csatlakozóhoz tartozó paraméteres Get

Post:

Végpontja: https://localhost:44316/api/Csatlakozo

Bemeneti értékek:

- Nev – típusa: string – Csatlakozó neve

Kimeneti értékek:

- Nev – típusa: string – Csatlakozó neve
- Hiba esetén szöveges üzenet

Használt státuszkódok:

- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 201 – Created (Sikerült a feltöltés)

```
[ResponseType(typeof(CsatlakozoModel))]
1 reference
public IHttpActionResult Post([FromBody] CsatlakozoModel value)
{
    try
    {
        var result = ctx.Csatlakozok.Add(new Csatlakozo
        {
            Nev = value.Nev
        });
        ctx.SaveChanges();


        return Content(HttpStatusCode.Created, result);
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "Ez a csatlakozo már létezik.");
        return InternalServerError(ex);
    }
}
```

44. ábra: Csatlakozóhoz tartozó Post

Put és a Delete nincs Implementálva:

```
public IHttpActionResult Put(int id, [FromBody] string value)
{
    return StatusCode(HttpStatusCode.NotImplemented);
}

// DELETE api/<controller>/5
0 references
public IHttpActionResult Delete(int id)
{
    return StatusCode(HttpStatusCode.NotImplemented);
}
```

45. ábra: Csatlakozóhoz a Put és Delete

ApplikacioController:

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Applikacio

Kimeneti értékek

- Nev – típusa: string – Alkalmazás neve
- KategoriaNev – típusa: string – Kategória neve
- KepeleresiUtja – típusa: string – Alkalmazáshoz tartozó kép
- Tarhely – típusa: int – Alkalmazás tárhely igénye

Használt státuszkód:

- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(ApplikacioModel))]
1 reference
public IHttpActionResult Get()
{
    IEnumerable<ApplikacioModel> result = null;
    result = ctx.Applikaciok.Include(x => x.Kategoria).Select(x => new ApplikacioModel
    {
        Nev = x.Nev,
        KategoriaNev = x.Kategoria.Nev,
        KepeleresiUtja = x.Kepeleresiutja,
        Tarhely=x.Tarhely

    }).ToList();
    return Ok(result);
}
```

46. ábra: Applikációhoz tartozó paraméter nélküli Get

Paraméteres Get:

Végpontja: https://localhost:44316/api/Applikacio/0?name=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Alkalmazás neve

Kimeneti értékek:

- Nev – típusa: string – Alkalmazás neve
- KategoriaNev – típusa: string – Kategória neve
- KepeleresiUtja – típusa: string – Alkalmazáshoz tartozó kép
- Tarhely – típusa: int – Alkalmazás tárhely igénye

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(ApplikacioModel))]
1 reference
public IHttpActionResult Get(int id, string name)
{
    ApplikacioModel result = null;
    result = ctx.Applikaciok.Include(x => x.Kategoria).Where(x => x.Nev == name).Select(x => new ApplikacioModel
    {
        Nev = x.Nev,
        KategoriaNev = x.Kategoria.Nev,
        KepeleresiUtja = x.Kepeleresiutja,
        Tarhely = x.Tarhely
    }).FirstOrDefault();
    if (result == null) return NotFound();
    return Ok( result);
}
```

47. ábra: Applikációhoz tartozó paraméteres Get

Post:

Végpontja: https://localhost:44316/api/Applikacio

Bemeneti értékek:

- Nev – típusa: string – Alkalmazás neve
- KategoriaNev – típusa: string – Kategória neve
- KepeleresiUtja – típusa: string – Alkalmazáshoz tartozó kép
- Tarhely – típusa: int – Alkalmazás tárhely igénye

Kimenti értékek:

- Szöveges üzenet

Használt státuszkódok:

- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 201 – Created (Sikerült a feltöltés)
- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)

```
[ResponseType(typeof(ApplikacioModel))]
1 reference
public IHttpActionResult Post([FromBody] ApplikacioModel value)
{
    var katId = ctx.Kategoriak.Where(x => x.Nev == value.KategoriaNev).Select(x => x.Id).FirstOrDefault();
    if (katId == 0) return Content(HttpStatusCode.NotFound, "Nem szerepel ez a kategoria");
    try
    {
        var result = ctx.Applikaciok.Add(new Applikacio
        {
            Nev=value.Nev,
            Tarhely=value.Tarhely,
            KatId=katId,
            Kepeleresiutja=value.KepeleresiUtja
        });
        ctx.SaveChanges();

        return Content(HttpStatusCode.Created,"");
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "már szerepel ez az applikáció");
        return BadRequest("Applikacio feltoltese sikertelen");
    }
}
```

48. ábra: Applikációhoz tartozó Post

Patch

Végpontja: https://localhost:44316/api/Applikacio/0?name=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Alkalmazás neve
- A modositani kivánt paraméterek

Kimeneti értékek:

- Kategória:
  - Id – típusa: int – Kategória egyedi azonosítója
  - Nev – típusa: string – Kategória neve
- Applikacio
  - Id – tipusa: int – Alkalmazás egyedi azonosítója
  - Nev – típusa: string – Alkalmazás neve
  - KategoriaNev – típusa: string – Kategória neve
  - KepeleresiUtja – típusa: string – Alkalmazáshoz tartozó kép
  - Tarhely – típusa: int – Alkalmazás tárhely igénye
- Hiba esetén szöveges üzenet

Használt státuszkódok:

- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 200 – Ok (Sikerült a kérés)
- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)

```
[ResponseType(typeof(ApplikacioModel))]
1 reference
public IHttpActionResult Patch(int id,string name, [FromBody] ApplikacioModel value)
{
    try
    {
        var result = ctx.Applikaciok.Where(x => x.Nev == name).FirstOrDefault();
        var katresult = ctx.Kategoriak.Where(x => x.Nev == value.KategoriaNev).Select(x=>x.Id).FirstOrDefault();
        if (katresult != 0) result.KatId = katresult;
        if (result == null) return Content(HttpStatusCode.NotFound, "Nincs ilyen Alkalmazás"); ;
        if (value.Nev != null) result.Nev = value.Nev;
        if (value.Tarhely != 0) result.Tarhely=value.Tarhely;
        if (value.KepeleresiUtja != null) result.Kepeleresiutja = value.KepeleresiUtja;
        if (value.KategoriaNev != null) result.KatId = katresult;

        ctx.SaveChanges();
        return Ok(result);
        //return Ok(result);
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "üttközés van a videokartyanal");
        return InternalServerError(ex);
    }
}
```

49. ábra: Applikációhoz tartozó Patch

Delete:

Végpontja: https://localhost:44316/api/Applikacio/0?name=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Alkalmazás neve

Kimeneti érték:

- Szöveges üzenet

Használt státuszkódok:

- 404 - NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 - Ok (Sikerült a kérés)

```
[ResponseType(typeof(ApplikacioModel))]
1 reference
public IHttpActionResult Delete(int id, string name)
{
    var applikaciok = ctx.Applikaciok.Where(x => x.Nev == name).FirstOrDefault();
    if (applikaciok == null) return NotFound();
    int ApplikacioId = applikaciok.Id;
    var set = ctx.Setupok.Where(x => x.ApplikacioId == ApplikacioId).ToList();
    ctx.Setupok.RemoveRange(set);
    var prof = ctx.Applikacio_Profilok.Where(x => x.AppId == ApplikacioId).ToList();
    ctx.Applikacio_Profilok.RemoveRange(prof);

    ctx.Applikaciok.Remove(applikaciok);
    ctx.SaveChanges();
    return Ok("Törlés sikeres");
}
```

50. ábra:Applikációhoz tartozó Delete

Applikacio_ProfilController

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Applikacio_Profil

Kimeneti értékek:

- UserName – típusa: string – Felhasználó neve
- ApplikacioNeve – típusa: string – Applikáció neve

Használt státuszkód:

- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(AlaplapModel))]
1 reference
public IHttpActionResult Get()
{
    IEnumerable<ApplikacioProfilModel> result = null;
    result = ctx.Applikacio_Profilok.Include(x => x.Applikacio).Include(x => x.Profil).Select(x => new ApplikacioProfilModel
    {
        UserName = x.Profil.Felhasznalonev,
        ApplikacioNeve = x.Applikacio.Nev
    }).ToList();
    return Ok(result);
}
```

51. ábra:App_Profilhoz tartozó paraméter nélküli Get

Paraméteres Get:

Végpontja: https://localhost:44316/api/Applikacio_Profil/0?name=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Alkalmazás neve

Kimeneti értékek:

- UserName – típusa: string – Felhasználó neve
- ApplikacioNeve – típusa: string – Applikáció neve

Használt státuszkódok:

- 404 - NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 - Ok (Sikerült a kérés)

```
[ResponseType(typeof(AlaplapModel))]
1 reference
public IHttpActionResult Get(int id, string name)
{
    ApplikacioProfilModel result = null;
    result = ctx.Applikacio_Profilok.Include(x => x.Applikacio).Include(x => x.Profil).Where(x => x.Applikacio.Nev == name)
        .Select(x => new ApplikacioProfilModel
    {
        UserName = x.Profil.Felhasznalonev,
        ApplikacioNeve = x.Applikacio.Nev
    }).FirstOrDefault();
    if (result == null) return NotFound();
    return Ok(result);
}
```

52. ábra:App_Profilhoz tartozó paraméteres Get

Post, Put és Delete nincs implementálva:

53. ábra: App_Profilhoz tartozó Post, Put, Delete

AlaplapController:

Paraméter nélküli Get:

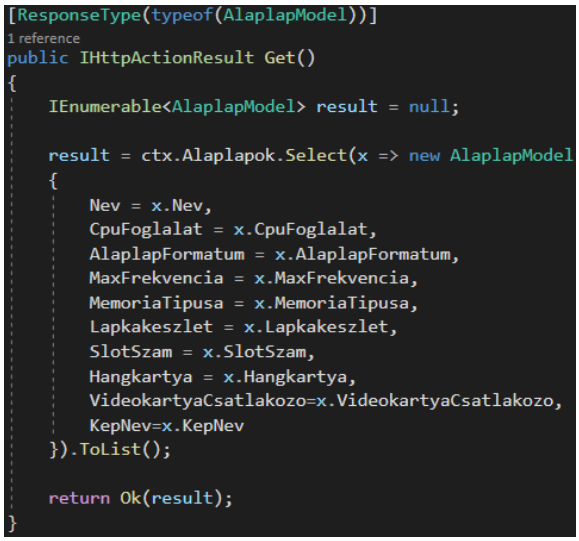
Végpontja: https://localhost:44316/api/Alaplap

Kimeneti értékek:

- Nev – típusa: string – Alaplap nev
- CpuFoglalat – típusa: string – Processzor foglalat
- AlaplapFormatum – típusa: string – Alaplap formátuma
- MaxFrekvencia – típusa: double – Ram max frekvenciája, amit kezel az alaplap
- MemoriaTipusa – típusa: string – Ram típusa, amivel kompatibilis az alaplap
- Lapkakeszlet – típusa: string – Alaplap lapkakészlete
- SlotSzam – típusa: int – Alaplapon található slotok száma
- Hangkartya – típusa: bool – Van-e integrált hangkártya az alaplapban
- VideokartyaCsatlakozo – típus: string – Videókártya foglalat típusa
- KepNev – típusa: string – Alaplap fényképe

Használt státuszkód:

- 200 – Ok (Sikerült a kérés)



54. ábra: Alaplaphoz tartozó paraméter nélküli Get

Paraméteres Get

Végpontja: https://localhost:44316/api/Alaplap/0?name=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Alaplap neve

Kimeneti értékek:

- Nev – típusa: string – Alaplap nev
- CpuFoglalat – típusa: string – Processzor foglalat
- AlaplapFormatum – típusa: string – Alaplap formátuma
- MaxFrekvencia – típusa: double – Ram max frekvenciája, amit kezel az alaplap
- MemoriaTipusa – típusa: string – Ram típusa, amivel kompatibilis az alaplap
- Lapkakeszlet – típusa: string – Alaplap lapkakészlete
- SlotSzam – típusa: int – Alaplapon található slotok száma
- Hangkartya – típusa: bool – Van-e integrált hangkártya az alaplapban
- VideokartyaCsatlakozo – típus: string – Videókártya foglalat típusa
- KepNev – típusa: string – Alaplap fényképe

Használt státuszkódok:

- 404 - NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 - Ok (Sikerült a kérés)

```
[ResponseType(typeof(AlaplapModel))]
1 reference
public IHttpActionResult Get(int id, string name)
{
    AlaplapModel result = null;
    result = ctx.Alaplapok.Where(x => x.Nev == name).Select(x => new AlaplapModel
    {
        Nev = x.Nev,
        CpuFoglalat = x.CpuFoglalat,
        AlaplapFormatum = x.AlaplapFormatum,
        MaxFrekvencia = x.MaxFrekvencia,
        MemoriaTipusa = x.MemoriaTipusa,
        Lapkakeszlet = x.Lapkakeszlet,
        SlotSzam = x.SlotSzam,
        Hangkartya = x.Hangkartya,
        VideokartyaCsatlakozo = x.VideokartyaCsatlakozo,
        KepNev=x.KepNev
    }).FirstOrDefault();
    if (result == null) return NotFound();
    return Ok(result);
}
```

55. ábra: Alaplaphoz tartozó paraméteres Get

Post

Végpontja: https://localhost:44316/api/Alaplap

Bemeneti értékek:

- Nev – típusa: string – Alaplap nev
- CpuFoglalat – típusa: string – Processzor foglalat
- AlaplapFormatum – típusa: string – Alaplap formátuma
- MaxFrekvencia – típusa: double – Ram max frekvenciája, amit kezel az alaplap
- MemoriaTipusa – típusa: string – Ram típusa, amivel kompatibilis az alaplap
- Lapkakeszlet – típusa: string – Alaplap lapkakészlete
- SlotSzam – típusa: int – Alaplapon található slotok száma
- Hangkartya – típusa: bool – Van-e integrált hangkártya az alaplapban
- VideokartyaCsatlakozo – típus: string – Videókártya foglalat típusa
- Csatlakozok – típusa: szöveges lista – Alaplaphoz tartozó csatlakozók neveit tárolja
- KepNev – típusa: string – Alaplap fényképe

Kimeneti értékek:

- Nev – típusa: string – Alaplap nev
- CpuFoglalat – típusa: string – Processzor foglalat
- AlaplapFormatum – típusa: string – Alaplap formátuma
- MaxFrekvencia – típusa: double – Ram max frekvenciája, amit kezel az alaplap
- MemoriaTipusa – típusa: string – Ram típusa, amivel kompatibilis az alaplap
- Lapkakeszlet – típusa: string – Alaplap lapkakészlete
- SlotSzam – típusa: int – Alaplapon található slotok száma
- Hangkartya – típusa: bool – Van-e integrált hangkártya az alaplapban
- VideokartyaCsatlakozo – típus: string – Videókártya foglalat típusa

- KepNev – típusa: string – Alaplap fényképe
- Hiba esetén szöveges üzenet

Használt státuszkódok:

- 409 – Conflict (Ütközés az adatbázisban)
- 400 – BadRequest (Valamilyen oknál fogva nem sikerül a feltöltés)
- 201 – Created (Sikeres feltöltés)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)

```
[ResponseType(typeof(AlaplapUploadModel))]
1 reference
public IHttpActionResult Post([FromBody] AlaplapUploadModel value)
{
    List<int> storageport = new List<int>();
    try
    {
        var result = ctx.Alaplapok.Add(new Alaplap
        {
            Nev = value.Nev,
            CpuFoglalat = value.CpuFoglalat,
            AlaplapFormatum = value.AlaplapFormatum,
            MaxFrekvencia = value.MaxFrekvencia,
            MemoriaTipusa = value.MemoriaTipusa,
            Lapkakeszlet = value.Lapkakeszlet,
            SlotSzam = value.SlotSzam,
            Hangkartya = value.Hangkartya,
            VideokartyaCsatlakozo = value.VideokartyaCsatlakozo,
            KepNev=value.Kepnev
        });
        ctx.SaveChanges();

        try
        {

            foreach (var item in value.Csatlakozok)
            {
                storageport.Add(ctx.Csatlakozok.Where(x => x.Nev == item).Select(x => x.Id).FirstOrDefault());
            }

            var storagemboard = ctx.Alaplapok.Where(x => x.Nev == value.Nev).FirstOrDefault();
            foreach (var item in storageport)
            {
                var resultconnect = ctx.Alaplap_Csatlakozok.Add(new Alaplap_Csatlakozo
                {
                    AlaplapId = storagemboard.Id,
                    CsatlakozoId = item,
                });
                ctx.SaveChanges();
            }

        }
        catch (Exception)
        {

            return Content(HttpStatusCode.BadRequest, new { error = "Alaplap és csatlakozo közti kapcsolat létrehozása sikertelen!" });
        }

        return Content(HttpStatusCode.Created, result);
    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.") return Content(HttpStatusCode.Conflict, "Már szerepel ezzel a névvel alaplap");
        return InternalServerError(ex);
    }
}
```

Patch

Végpontja: https://localhost:44316/api/Alaplap/0?name=

Bementi értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Alaplap neve
- A modositani kivánt paraméterek

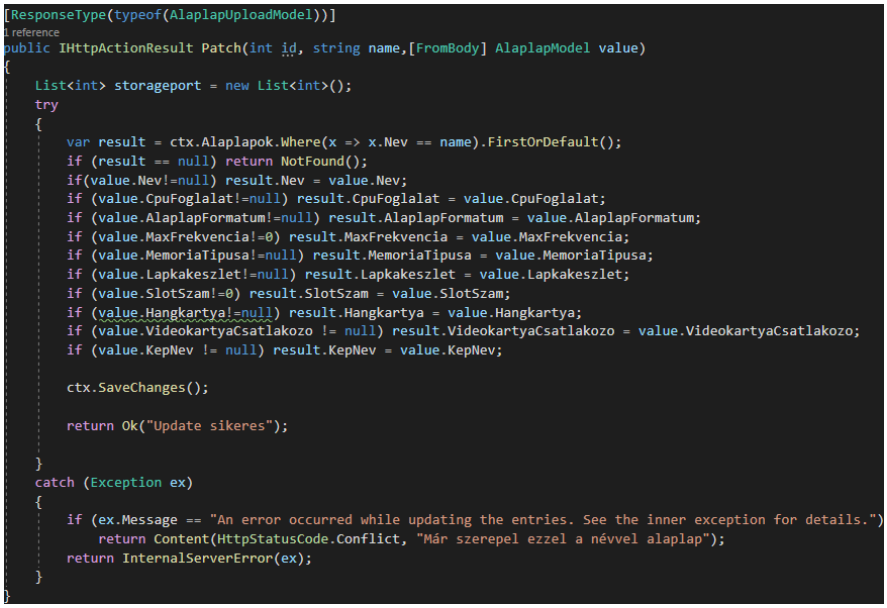
Kimeneti értékek:

- Szöveges üzenet

Használt státuszkódok:

- 409 – Conflict (Ütközés az adatbázisban)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 200 – Ok (Sikerült a kérés)

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)



```
[ResponseType(typeof(AlaplapUploadModel))]
1 reference
public IHttpActionResult Patch(int id, string name,[FromBody] AlaplapModel value)
{
    List<int> storageport = new List<int>();
    try
    {
        var result = ctx.Alaplapok.Where(x => x.Nev == name).FirstOrDefault();
        if (result == null) return NotFound();
        if(value.Nev!=null) result.Nev = value.Nev;
        if (value.CpuFoglalat!=null) result.CpuFoglalat = value.CpuFoglalat;
        if (value.AlaplapFormatum!=null) result.AlaplapFormatum = value.AlaplapFormatum;
        if (value.MaxFrekvencia!=0) result.MaxFrekvencia = value.MaxFrekvencia;
        if (value.MemoriaTipusa!=null) result.MemoriaTipusa = value.MemoriaTipusa;
        if (value.Lapkakeszlet!=null) result.Lapkakeszlet = value.Lapkakeszlet;
        if (value.SlotSzam!=0) result.SlotSzam = value.SlotSzam;
        if (value.Hangkartya!=null) result.Hangkartya = value.Hangkartya;
        if (value.VideokartyaCsatlakozo != null) result.VideokartyaCsatlakozo = value.VideokartyaCsatlakozo;
        if (value.KepNev != null) result.KepNev = value.KepNev;

        ctx.SaveChanges();

        return Ok("Update sikeres");

    }
    catch (Exception ex)
    {
        if (ex.Message == "An error occurred while updating the entries. See the inner exception for details.")
            return Content(HttpStatusCode.Conflict, "Már szerepel ezzel a névvel alaplap");
        return InternalServerError(ex);
    }
}
```

56. ábra: Alaplaphoz tartozó Patch

Delete:

Végpontja: https://localhost:44316/api/Alaplap/0?name=

Bementi értékek:

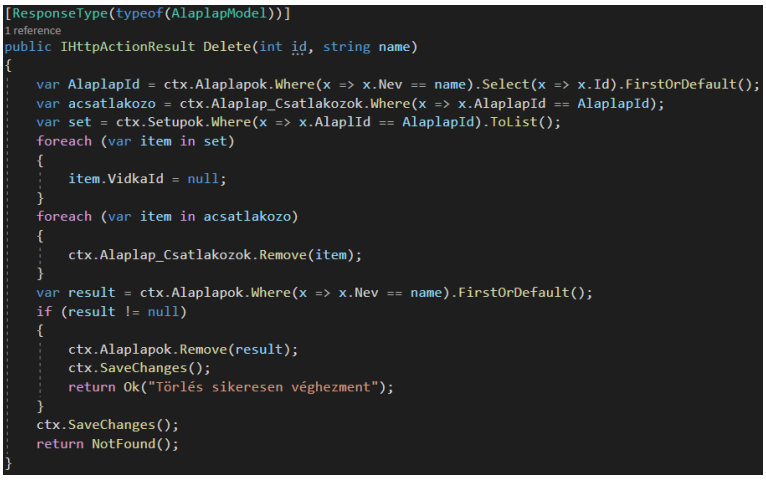
- id – típusa: int – mindig statikus nulla
- name – típusa: string – Alaplap neve

Kimeneti érték:

- Szöveges üzenet

Használt státuszkódok:

- 200 – Ok (Sikerült a kérés)
- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)



```
[ResponseType(typeof(AlaplapModel))]
1 reference
public IHttpActionResult Delete(int id, string name)
{
    var AlaplapId = ctx.Alaplapok.Where(x => x.Nev == name).Select(x => x.Id).FirstOrDefault();
    var acsatlakozo = ctx.Alaplap_Csatlakozok.Where(x => x.AlaplapId == AlaplapId);
    var set = ctx.Setupok.Where(x => x.AlaplId == AlaplapId).ToList();
    foreach (var item in set)
    {
        item.VidkaId = null;
    }
    foreach (var item in acsatlakozo)
    {
        ctx.Alaplap_Csatlakozok.Remove(item);
    }
    var result = ctx.Alaplapok.Where(x => x.Nev == name).FirstOrDefault();
    if (result != null)
    {
        ctx.Alaplapok.Remove(result);
        ctx.SaveChanges();
        return Ok("Törlés sikeresen véghezment");
    }
    ctx.SaveChanges();
    return NotFound();
}
```

57. ábra: Alaplaphoz tartozó Delete

Alaplap_CsatlakozoController

Paraméter nélküli Get:

Végpontja: https://localhost:44316/api/Alaplap_Csatlakozok

Kimeneti értékek:

- AlaplapNev – típusa: string – Alaplap neve
- CsatlakozoNev - típusa: string – Csatlakozó neve

Használt státuszkód:

- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(AlaplapModel))]
1 reference | 1/1 passing
public IHttpActionResult Get()
{
    IEnumerable<AlaplapCsatlakozModel> result = null;
    result = ctx.Alaplap_Csatlakozok.Include(x => x.Csatlakozo).Include(x => x.Alaplap).Select(x => new AlaplapCsatlakozModel
    {
        AlaplapNev = x.Alaplap.Nev,
        CsatlakozoNev = x.Csatlakozo.Nev
    }).ToList();
    return Ok(result);
}
```

58. ábra: Alaplap_Csatlakozohoz tartozó paraméter nélküli Ger

Paraméteres Get:

Végpontja: https://localhost:44316/api/Alaplap_Csatlakozok/0?name=

Bementi értékek:

- id – típusa: int – mindig statikus nulla
- name – típusa: string – Alaplap neve

Kimeneti érték:

- AlaplapNev – típusa: string – Alaplap neve
- CsatlakozoNev - típusa: string – Csatlakozó neve

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(AlaplapModel))]
1 reference | 1/1 passing
public IHttpActionResult Get(int id, string name)
{
    IEnumerable<AlaplapCsatlakozModel> result = null;
    result = ctx.Alaplap_Csatlakozok.Include(x => x.Csatlakozo).Include(x => x.Alaplap).Where(x => x.Alaplap.Nev == name)
        .Select(x => new AlaplapCsatlakozModel
    {
        AlaplapNev = x.Alaplap.Nev,
        CsatlakozoNev = x.Csatlakozo.Nev
    }).ToList();
    if (result == null) return NotFound();
    return Ok(result);
}
```

59. ábra:Alaplapi_Csatlakozóhoz tartozó paraméteres Get

Post:

Végpontja: https://localhost:44316/api/Alaplap_Csatlakozok

Bemeneti értékek:

- AlaplapNev – típusa: string – Alaplap neve
- CsatlakozoNev - típusa: szöveges lista – Csatlakozók nevei

Kimeneti értékek:

- Szöveges üzenet

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 201 – Create (Sikeres feltöltés)

```
[ResponseType(typeof(AlaplapModel))]
1 reference | 1/1 passing
public IHttpActionResult Post([FromBody] AlaplapCsatlakozPOSTModel value)
{
    var AlaplapId = ctx.Alaplapok.Where(x => x.Nev == value.AlaplapNev).Select(x => x.Id).FirstOrDefault();
    if (AlaplapId == -1) return NotFound();
    List<int> A = new List<int>();
    try
    {
        foreach (var item in value.Csatlakozok)
        {
            A.Add(ctx.Csatlakozok.Where(x => x.Nev == item).Select(x => x.Id).FirstOrDefault());
        }
        foreach (var item in A)
        {
            ctx.Alaplap_Csatlakozok.Add(new Alaplap_Csatlakozo
            {
                AlaplapId = AlaplapId,
                CsatlakozoId = item
            });
        }
        ctx.SaveChanges();
        return Content(HttpStatusCode.Created,"");
    }
    catch (Exception ex)
    {
        return InternalServerError(ex);}}
```

60. ábra: Alaplap_Csatlakozóhoz tartozó Post

Put nincs implementálva:

```
[ResponseType(typeof(AlaplapModel))]
0 references
public IHttpActionResult Put(int id,string name, [FromBody] string value)
{
    return StatusCode(HttpStatusCode.NotImplemented);
}
```

61. ábra: Alaplap_Csatlakozóhoz tartozó Put

Delete:

Végpontja: https://localhost:44316/api/Alaplap_Csatlakozok/0?AlaplapNeve=&CsatlakozoNev=

Bemeneti értékek:

- id – típusa: int – mindig statikus nulla
- AlaplapNeve– típusa: string – Alaplap neve

- CsatlakozoNev- típusa: string – Csatlakozó neve

Kimeneti érték:

- Szöveges üzenet

Használt státuszkódok:

- 404 – NotFound (nem található a bemeneti értékeknek megfelelő adat)
- 500 – InternalServerError (Valami akadályozza a kérés teljesitését, és nincs lekezelve)
- 200 – Ok (Sikerült a kérés)

```
[ResponseType(typeof(AlaplapModel))]
1 reference | 1/1 passing
public IHttpActionResult Delete(int id, string AlaplapNeve, string CsatlakozoNev)
{
    try
    {
        var AlaplapId = ctx.Alaplapok.Where(x => x.Nev == AlaplapNeve).Select(x => x.Id).FirstOrDefault();

        if (AlaplapId == 0) return Content(HttpStatusCode.NotFound, "Nem található ilyen alaplap");

        var csatlakId = ctx.Csatlakozok.Where(x => x.Nev==CsatlakozoNev).Select(x=>x.Id).FirstOrDefault();

        if(csatlakId==0) return Content(HttpStatusCode.NotFound, "Nem található csatlakozó");

        var kapcsaolat = ctx.Alaplap_Csatlakozok.Where(x => x.AlaplapId == AlaplapId && x.CsatlakozoId==csatlakId).ToList();

        if (kapcsaolat.Count == 0) return Content(HttpStatusCode.NotFound, "Megadott csatlakozók egyike sem kapcsolódik az alaplaphoz.");

        ctx.Alaplap_Csatlakozok.RemoveRange(kapcsaolat); //több rekordot tud törölmi
        ctx.SaveChanges();

        return Ok("A törlés sikeresen megtörtént!");
    }
    catch (Exception ex)
    {
        return InternalServerError(ex);
    }
}
```

62. ábra: Alaplap_Csatlakozóhoz tartozó Delete

Tartalomjegyzék