csarp-back-03-01-01-student-delete-insert-study

# Diák törlése és új diák felétele a backend oldalon

A kiindulási repónk az a repó ahol a diák adatok módosítását végeztük. De ezt a kódot refaktoráltuk Dto osztályok bevezetésével és a Kreate.Shared projekt létrehozásával: <https://github.com/csarp-wpf-mvvm/07-00-01-wpf-mvvm-update-stady>

Ha a törlés kódjával megvagyunk és elfogadhatónak tartjuk a kódot, akkor a munkát folytathatjuk új diák adatának felvételével.

## Új diák adatának felvétel

Az eljárás nagyon hasonló lesz az update művelethez. Arra figyelünk, ha frontendtől kapott adatnak van valódi id-je, akkor frissíteni kell az adatot, ha nem akkor új adatként felvenni.

A Student osztályba tehetünk kiegészítő metódusokat, property-ket, mert nem Dto osztály, tehát nem használjuk adatközvetítésre. Először a Student osztály kiegészítjük egy HasId property-vel. A HasId property jelzi, hogy van-e valós id-je a diáknak:

public bool HasId => Id != Guid.Empty;

Ezután az StudentRepo osztályba implementálunk egy InsertAsyncmetódust:

public async Task<ControllerResponse> InsertStudentAsync(Student student)

{

if (student.HasId)

{

return await UpdateStudentAsync(student);

}

else

{

return await InsertNewItemAsync(student);

}

}

Ezután megírjuk az InsertNewItemAsync **private** metódust amely sokban hasonlít a már megírt Update metódusra.

Ezután a Controllerbe megírjuk a Post kérés, kevéske refaktorálással, amivel egyszerűrítjük az insert és update metódust is.

Teszteléshez használható json kód (Guid.Empy így az Id):

{

"id": "00000000-0000-0000-0000-000000000000",

"firstName": "Marcel",

"lastName": "Matek",

"birthsDay": "2021-04-04T00:00:00",

"schoolYear": 10,

"schoolClass": 2,

"educationLevel": "érettségi"

}