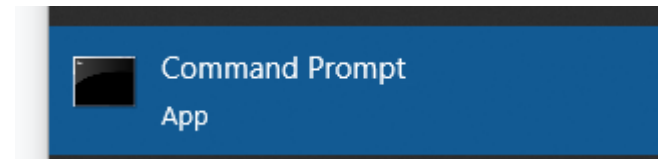


Vježba 6 - Integracijsko testiranje

Uvod

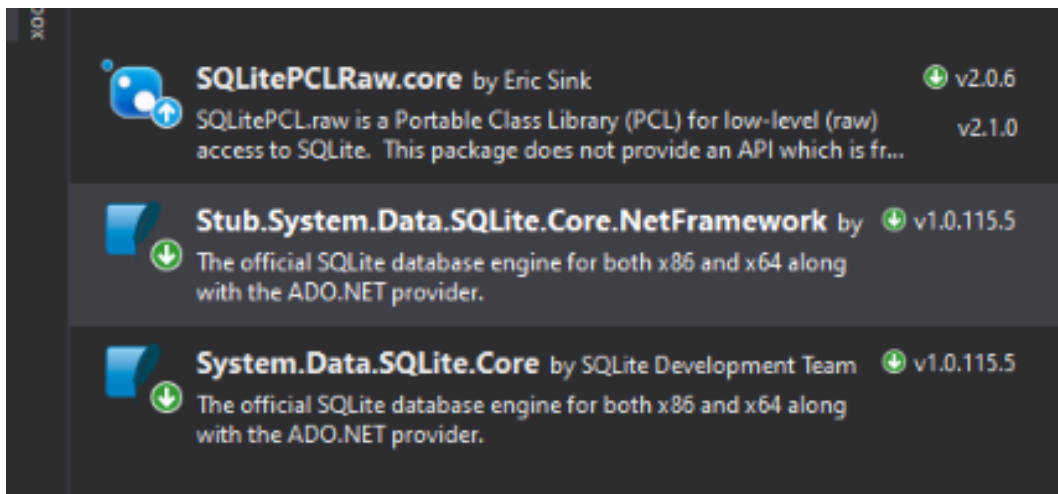
Dohvat koda za vježbu 6 iz git repozitorija



```
D:
cd TPP\Study
git fetch --all
git checkout app_study_vj6
git pull origin app_study_vj6
```

Ručni instalirati pakete u projekt Study.IntegrationTests:

- Desni klik na ime projekta -> Manage NuGet Packages



Integracijski test - osnovno, na stvarnim klasama

ZADATAK 1:

Testirati klasu `StudentRepository`, integrirati i testirati s klasom `SqliteDataAccess`

- Testirati metodu `Add`
 - Dodati studenta u bazu podataka
 - Provjeriti je li student dodan
 - Provjeriti postoji li student čije su ime i razina jednaki dodanom
 - `TearDown`: Izbrisati dodanog studenta

ZADATAK 2:

Testirati klasu `StudentRepository`, integrirati i testirati s klasom `JsonDataAccess`

- Testirati metodu `Remove`
 - Izbrisati studenta iz json datoteke pozivom `Remove`
 - Provjeriti je li student izbrisan
 - Provjeriti postoji li student s identifikatorom koji je jednak poslanom

- TearDown: Ponovo dodati izbrisanog studenta

ZADATAK 3:

Na klasi `Student` testirati metodu `Enroll()`

- Dodati studenta u bazu podataka pomoću repozitorija (koristiti `SqliteDataAccess`)
- Dodati course u bazu podataka pomoću repozitorija (koristiti `SqliteDataAccess`)
- Pozvati `Enroll`
- Provjeriti `CourseId` na studentu
- Poništiti promjene na bazi podataka

ZADATAK 4:

Na klasi `Student` testirati metodu `GetNumberOfCourseMates()`

- Pomoću repozitorija dodati 3 studenta u bazu podataka
- Pozvati `Enroll` na svim studentima
- Pozvati `GetNumberOfCourseMates()` na jednom studentu
- Provjeriti je li broj jednak 2
- Poništiti promjene na bazi podataka

Pristup Big Bang

ZADATAK 5:

Testirati metodu `Index` na klasi `HomeController`

- Instancirati klasu `HomeController` s repozitorijima:
 - `StudentRepository`
 - `CourseRepository`
 - `CityRepository`
- Repozitorije instancirati s klasom `SqliteDataAccess`
- Pozvati `Index()`
- Provjeriti sadrži li `ViewBag` liste:
 - `Students`
 - `Courses`
 - `Cities`

Pristup Top Down

ZADATAK 6:

Izraditi novi model `Country`

- Implementira sučelje: `IModel`
- Svojstva:
 - `int Id`
 - `string Name`
- Izraditi interface `ICountryRepository` koji proširuje `IRepository<Country>`
- Modificirati `HomeController`
 - Dodati svojstvo tipa `ICountryRepository`
 - postaviti vrijednost pomoću prvog konstruktora
 - Modificirati `Index()`
 - Pozvati metodu `GetAll` repozitorija `CountryRepository` i spremiti vrijednost u `ViewBag.Countries`
- Testirati metodu `Index()`
 - NAPOMENA: Parametri konstruktora `HomeController` su u ovom zadatku izmijenjeni, zbog čega puca test iz prethodnog zadatka.

Taj test sada treba izmijeniti na način da se kreira mock klasa `ICountryRepository` koju ćemo dodati kao parametar u konstruktoru `HomeController`

```
_homeController = new HomeController(  
    _studentRepository,  
    _courseRepository,  
    _cityRepository,  
    _countryRepositoryMock.Object  
);
```

- Izraditi stub za `CountryRepository` pomoću biblioteke `Moq`
 - Programirati stub tako da metoda `GetAll` vraća listu od 3 proizvoljne države
- Pozvati `Index()`
- Provjeriti da `ViewBag` sadrži liste:
 - `Students`
 - `Courses`
 - `Cities`
 - `Countries`

ZADATAK 7:

Implementirati `CountryRepository`

- Izraditi klasu `CountryRepository` koja implementira interface `ICountryRepository` i nasljeđuje klasu `Repository`
- Izraditi tablicu `Country` u bazi podataka
 - Otvoriti "D:\TPP\Study\DB.Browser.for.SQLite-3.12.2-win64\DB Browser for SQLite\DB Browser for SQLite.exe"
 - Kliknuti "Open Database" i pronaći: "D:\TPP\Study\Study\StudyDB.db"

- Kreirati tablicu Country

Table: Country

Advanced

Fields Constraints

Add Remove Move to top Move up Move down Move to bottom

Name	Type	NN	PK	AI	U	Default	Check
Id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Name	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```

1 CREATE TABLE "Country" (
2     "Id"    INTEGER NOT NULL UNIQUE,
3     "Name"  TEXT,
4     PRIMARY KEY("Id" AUTOINCREMENT)
5 );
  
```

OK Cancel

- Dodati barem jednu državu u bazu podataka
- Spremiti izmjene klikom na "Write Changes"
- Ponoviti testove za metodu `Index()`
 - Zamijeniti stub s implementiranim repozitorijem

Pristup Bottom up

ZADATAK 8:

Testirati integrirani rad modula `StudentRepository` i `StudentSorter`

- Izraditi driver klasu `StudentDriver`
 - Napisati metode
 - `List<Student> GetStudentsOrderedByName()`
 - Dohvatiti studente iz repozitorija `StudentRepository`
 - Sortirati pomoću metode `ByName` klase `StudentSorter`
 - `List<Student> GetStudentsOrderedByLevel()`
 - Dohvatiti studente iz repozitorija `StudentRepository`
 - Sortirati pomoću metode `ByLevel` klase `StudentSorter`
- Testirati integraciju testiranjem metoda `GetStudentsOrderedByName()` i `GetStudentsOrderedByLevel()`
 - Provjeriti da metode vraćaju liste u ispravnom redoslijedu

Slanje vježbi na Moodle