بسم اللّه الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

آموزش پرس و جو و ذخیره Async (همسان سازی)

مدرس: مهندس افشین رفوآ

آدرس آموزشگاه. تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 561 - واحد 7
88146323 - 88446780 - 88146330

آموزش پرس و جو و ذخیره Async (همسان سازی)

شما می توانید از مزیت اجرای غیر هم زمان .net 4.5 با entity framework استفاده کنید. EF 6 می تواند با استفاده از DbContext به طور غیر هم زمان یک query و دستور را اجرا کند.

ابتدا ببینیم که چگونه پرس و جوی غیر همزمان را اجرا کنیم و سپس به طور غیر همزمان متد context.SaveChanges را فرخوانی می کنیم.

Query غیر هم زمان

```
private static async Task GetStudent()
            {
                Student student = null;
                using (var context = new SchoolDBEntities())
                {
                    Console.WriteLine("Start GetStudent...");
                    student = await (context.Students.Where(s =>
s.StudentID == 1).FirstOrDefaultAsync());
                    Console.WriteLine("Finished GetStudent...");

                }
                return student;
        }
```

همانطور که در کد بالا مشاهده کردید، متد GetStudent با async برای asynchronous (غیر هم زمان) مشخص شده است. نوع برگشتی متد asynchrounous باید Task باشد. متد GetStudent شی ای از نوع Student برمی گرداند بنابراین نوع برگشتی باید Task باشد.

همچنین پرس و جو با await مشخص شده است یعنی یک فراخوانی thread برای انجام دادن چیزی است مگر اینکه پرس و جو را اجرا می کند و داده را برمی گرداند. ما در اینجا از متد گسترشی FirstOrDefaultAsync از System.Data.Entity استفاده می کنیم. همچنین می توانید از دیگر متد های گسترشی به طور مناسب استفاده کنید مانند SingleOrDefaultAsync، ToListAsyn و غیره.

ذخیره کردن غیر همزمان (Asynchronous Save)

می توانید متد context.SaveChanges به طور غیر همزمان، همان طور که پرس وجوی async را فراخوانی می کنید.

```
  private static async Task SaveStudent(Student editedStudent)
            {
                using (var context = new SchoolDBEntities())
                {
                    context.Entry(editedStudent).State =
EntityState.Modified;
                    Console.WriteLine("Start SaveStudent...");
                    int x = await (context.SaveChangesAsync());
                    Console.WriteLine("Finished
SaveStudent...");
                }
            }
```

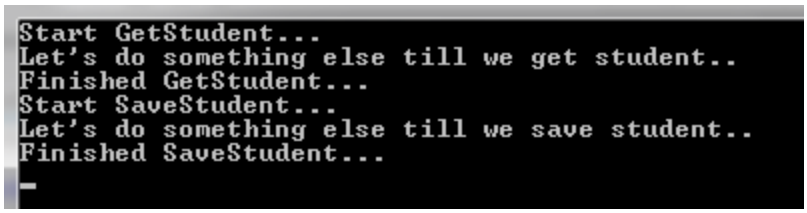
نتیجه پرس و جوی async

شما زمانی می توانید نتیجه بگیرید که asynchronous از متد Wait به شرح زیر استفاده می کند.

```
   public static void AsyncQueryAndSave()
            {
                var student = GetStudent();
                Console.WriteLine("Let's do something else till
we get student..");
                student.Wait();
                var studentSave = SaveStudent(student.Result);
                Console.WriteLine("Let's do something else till
we get student.." );
                studentSave.Wait();
            }
```

همان طور که در کد بالا می بینید، ما متد غیر همزمان GetStudent را به همان طریق معمول فراخوانی می کنیم و نتیجه را در متغییر student ذخیره می کنیم. سپس ما student.wait() را فراخوانی می کنیم که این بدین معنی است که فراخوانی thread تا زمانی که متد asynchronous کامل شود باید صبر کند، بنابراین می توانیم فرآیند دیگری انجام دهیم تا زمانی که از متدasynchronous نتیجه بگیریم.

کد نشان داده شده در بالا خروجی زیر را خواهد داشت.



می توانید پروژه نمونه را برای Async query & save دانلود کنید.