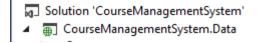
Playbook for 3-Tier Architecture – C# WEB API GENERATION

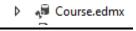
1. Create a Blank C# Solution



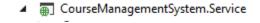
2. Create a C# Class Library (.DLL): Data Layer (within the same solution)



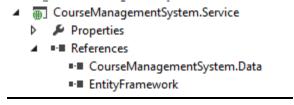
- 3. Add EF 6.x Reference using NUGET Packages
- 4. Create an EDMX file with DB first Approach



5. Create a C# Class Library(.DLL): Service Layer (within the same solution)



6. Add EF 6.x Reference using NUGET Packages and Add DataLayer (.DLL) reference to Service Layer



7. <u>Create Service Repository and Interface to fetch the</u>
<u>records from the data layer</u>

```
■ CourseManagementSystem.Service

Properties

References

C** CourseService.cs

C** ICourseService.cs

Packages.config

Web.config

Web.Debug.config

Web.Release.config
```

8. CourseService.cs

```
public class CourseService:ICourseService
{
    private CourseContext cc= new CourseContext();
    public List<Data.Course> GetCourses()
    {
        return cc.Courses.ToList();
    }

    public Data.Course GetCourse(string code)
    {
        throw new NotImplementedException();
    }
}
```

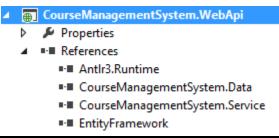
9. ICourseService.cs

```
public interface ICourseService
{
    __references
    List<Course> GetCourses();
    __references
    Course GetCourse(string code);
}
```

10. Create a WEB API (within the same solution)



11. Add EF 6.x Reference using NUGET Packages and Add DataLayer(.DLL) reference and Service Layer(.DLL)



12. Copy the Connection Strings from WEB.CONFIG file from the Data Layer and add to the WEB.CONFIG

System.web

```
<connectionStrings>
     <add name="CourseContext" connectionString="metadata=res://*/Course.csdl
</connectionStrings>
```

13. Generate a APIController corresponding to the context name

```
Properties
  ▶ ■ ■ References
    App_Data
  App_Start
  Areas
  Content
  Controllers
    ▶ C# AccountController.cs
    ▶ C# CoursesController.cs
    ▶ C# HomeController.cs
    ▶ C# ValuesController.cs
  fonts
  Models
    C# AccountBindingModels.cs
    ▶ C# AccountViewModels.cs
    ▶ C# IdentityModels.cs
```

14. Generate a APIController corresponding to the context name and define corresponding controller action method

```
public class CoursesController : ApiController
{
    private CourseContext db = new CourseContext();

    // GET: api/Courses
    oreferences
    public IQueryable<Course> GetCourses()
    {
        return db.Courses;
    }
}
```

15. <u>Build and Publish the web api. Access the Web API resource using http://localhost:port/api/courses/</u>

Copyright 2015-16 Syed Awase Khirni			

C# MVC5 Web API – Ground Up Training Series Copyright 2015-16 Syed Awase Khirni