

Веб-разработка: ASP.NET MVC

Яцканич Андрей Иванович
Программист компании Фогсофт
Опыт разработки: 3 года

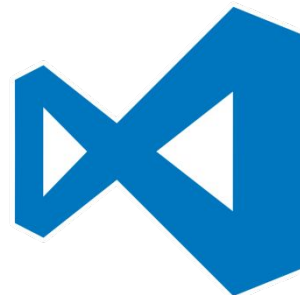
ASP.NET + IIS

Введение в разработку с
использованием ASP.NET MVC 4

Инструменты



Visual Studio®



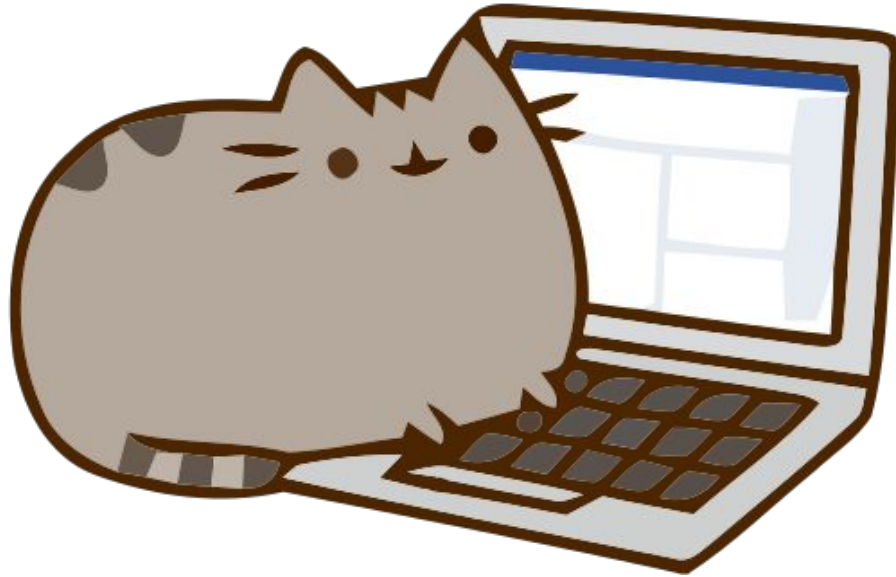
С чем познакомимся



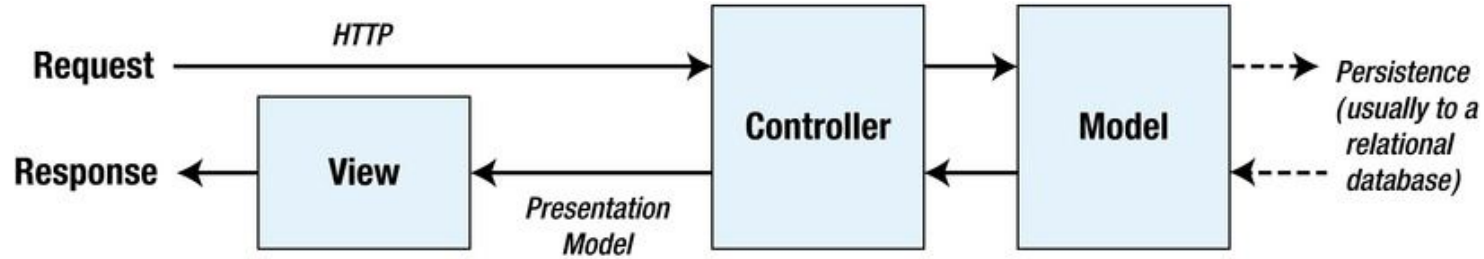
ReSharper



ASP.NET MVC Framework



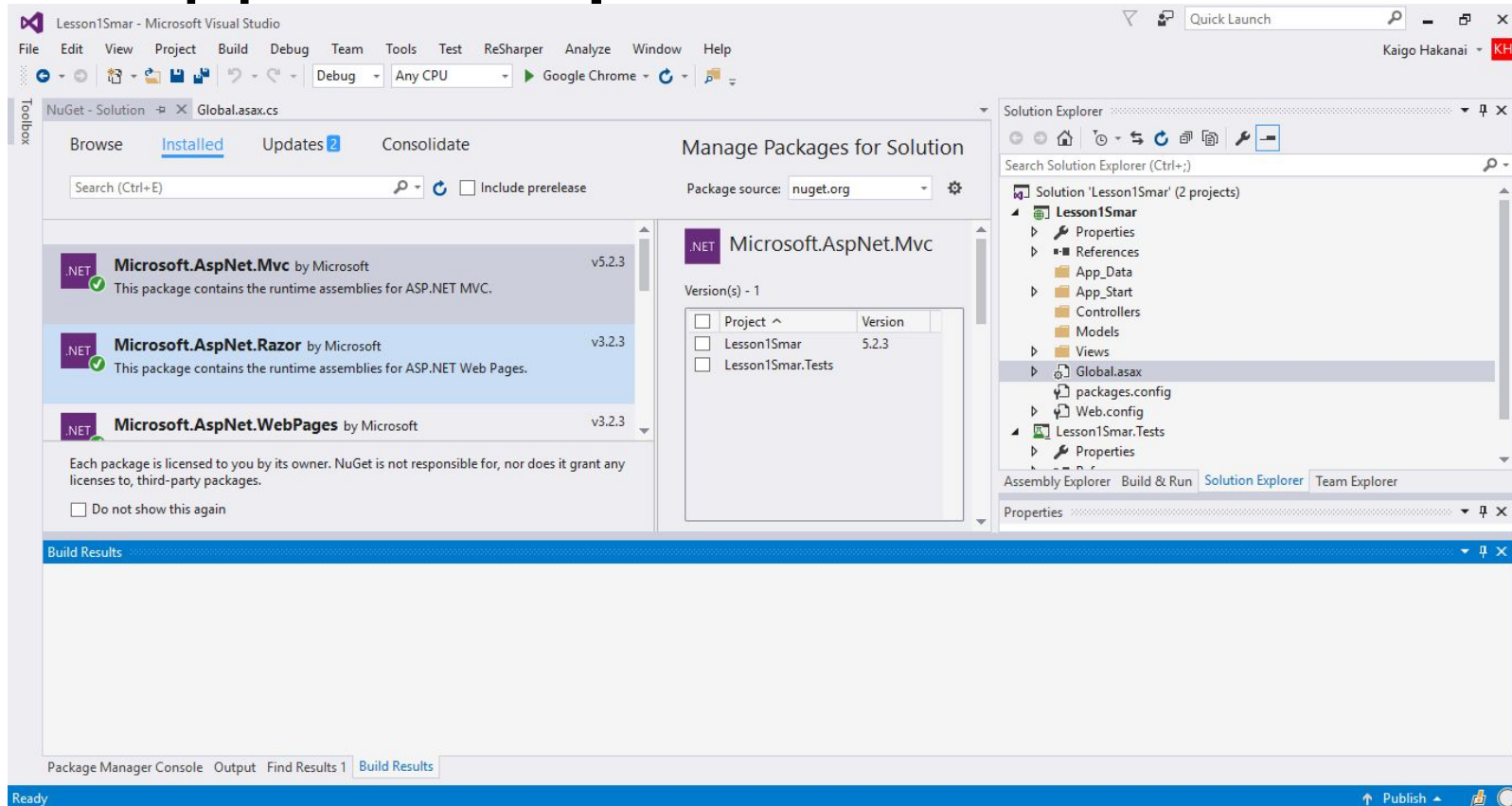
ASP.NET MVC Pattern



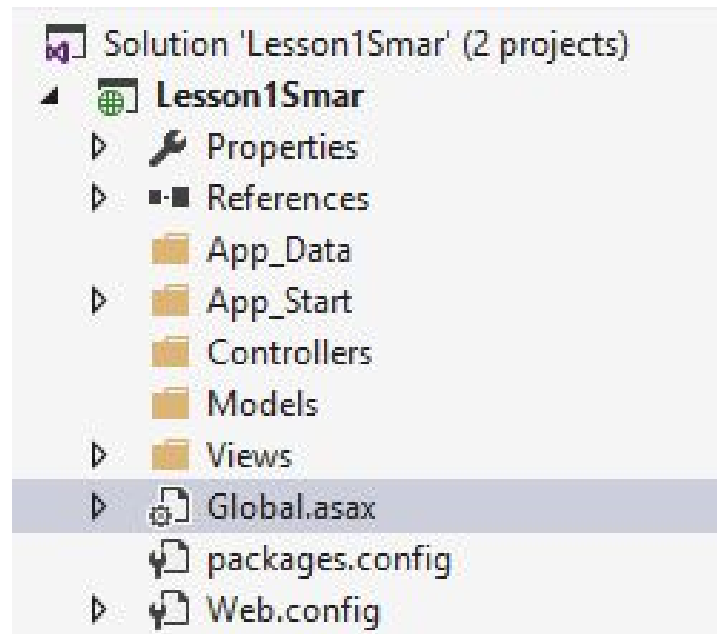
Демонстрация



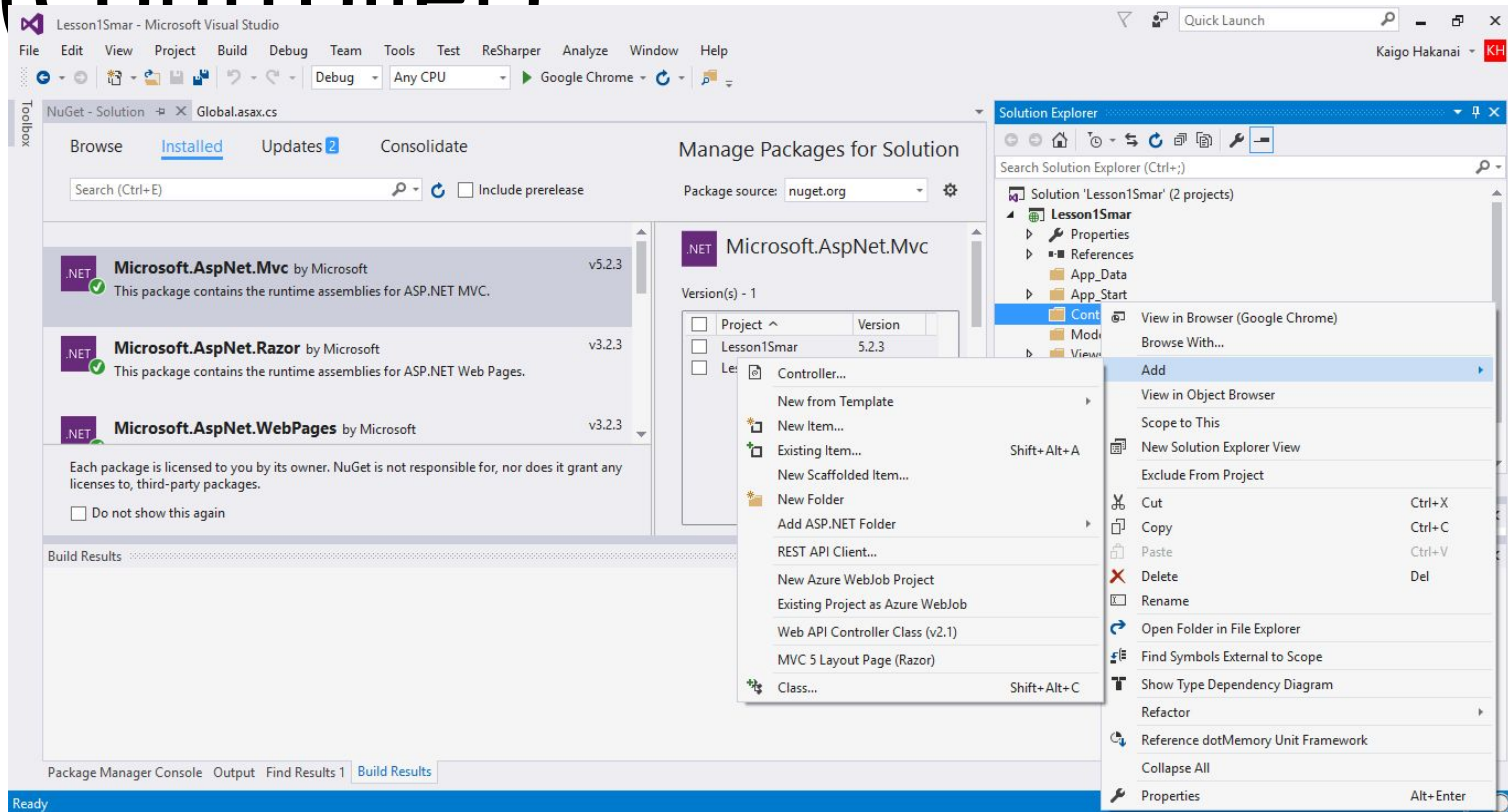
Создание проекта



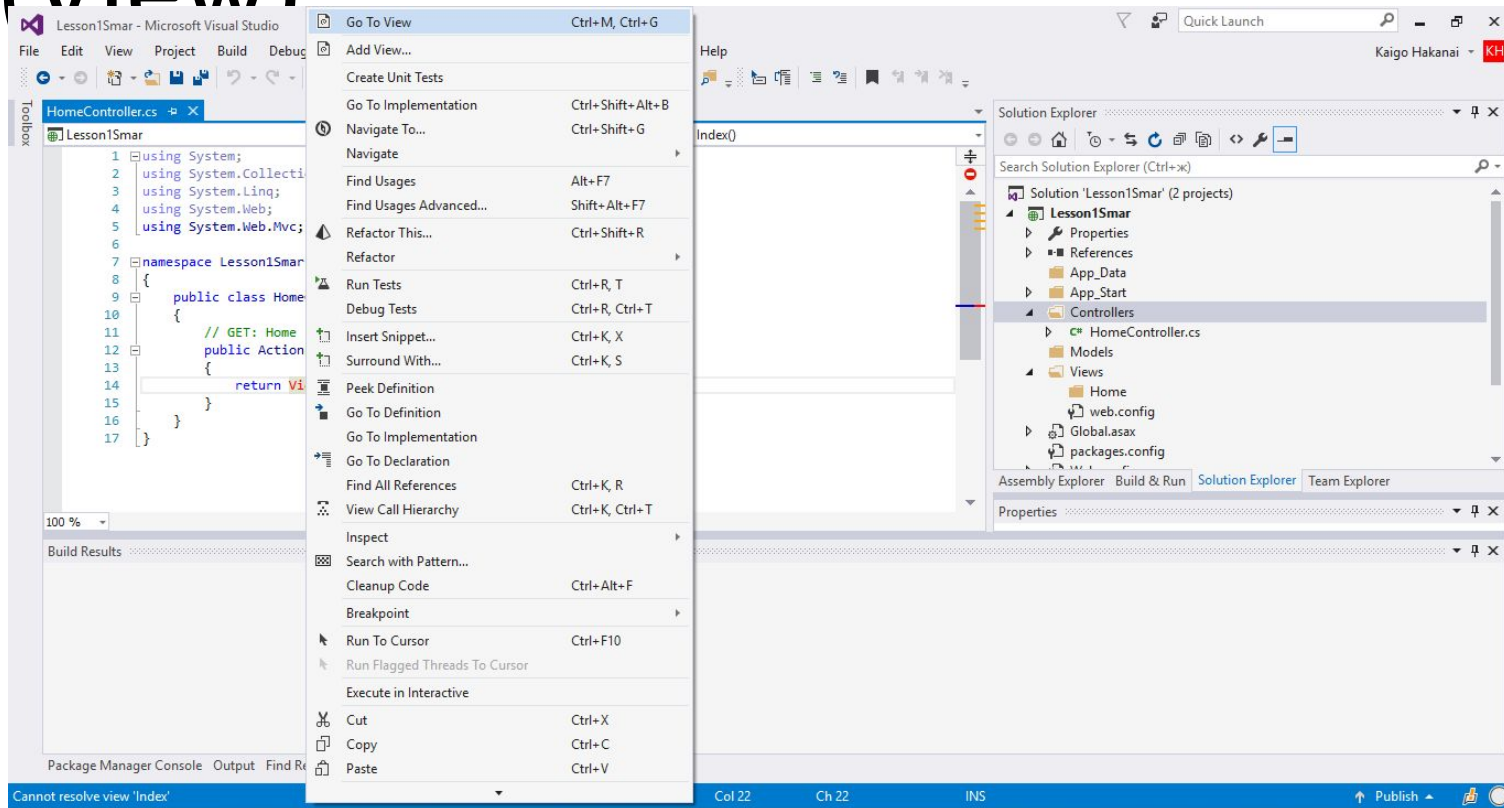
Структура MVC проекта



Добавить контроллер (Controller)



Добавить представление (View)



Cassini

Основные отличия:

- Хостинг лишь одного ASP.NET приложения на порт
- Нет поддержки HTTPS
- Не поддерживает аутентификацию
- Отвечает на запросы лишь с локальной машины



Логирование информации

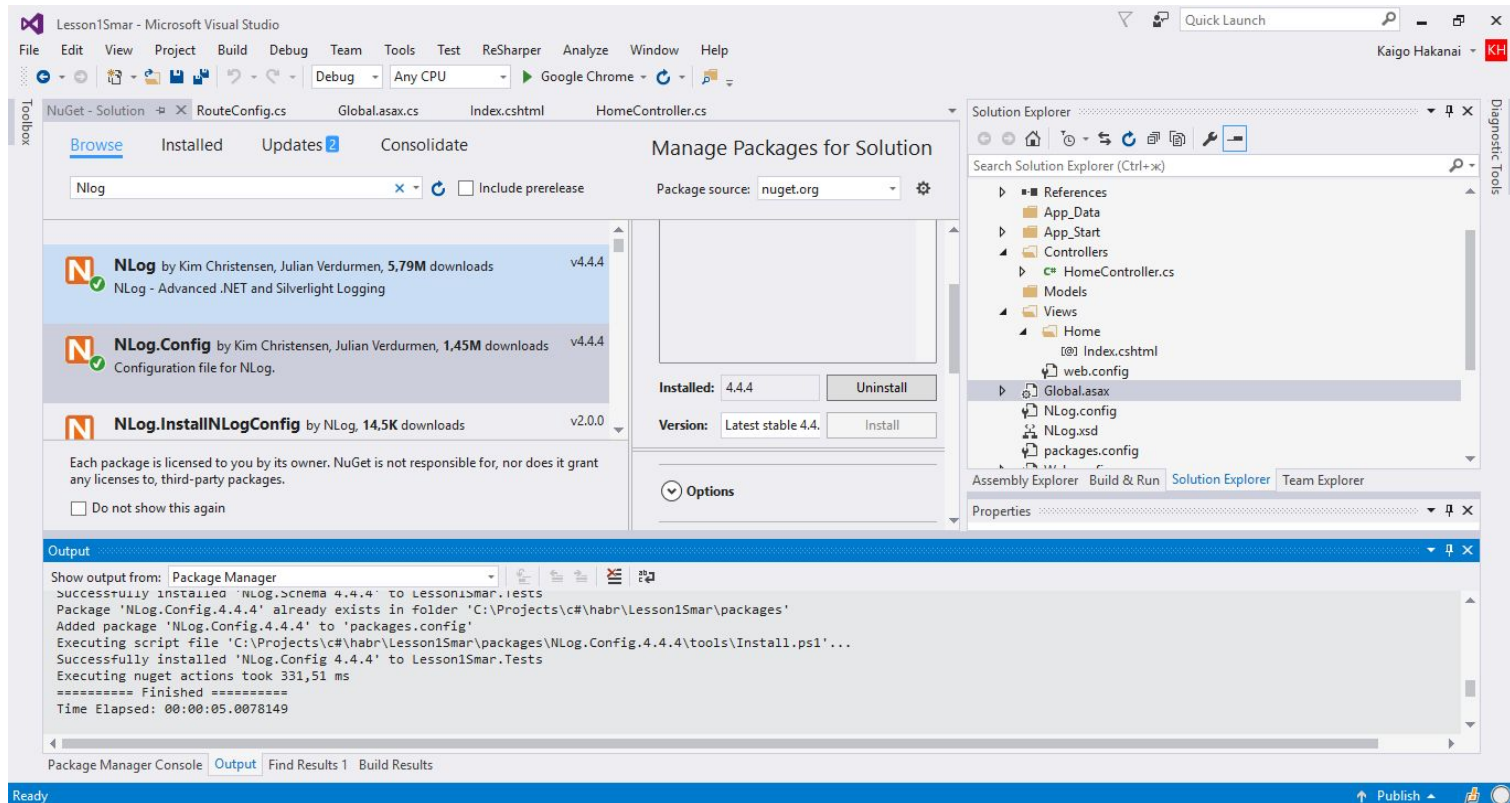


Log4Net

NuGet магия

- Get-Package
- Get-Package –ListAvailable
- Get-Package -ListAvailable -Filter Nlog
- Install-Package Nlog

Но можно попроще....



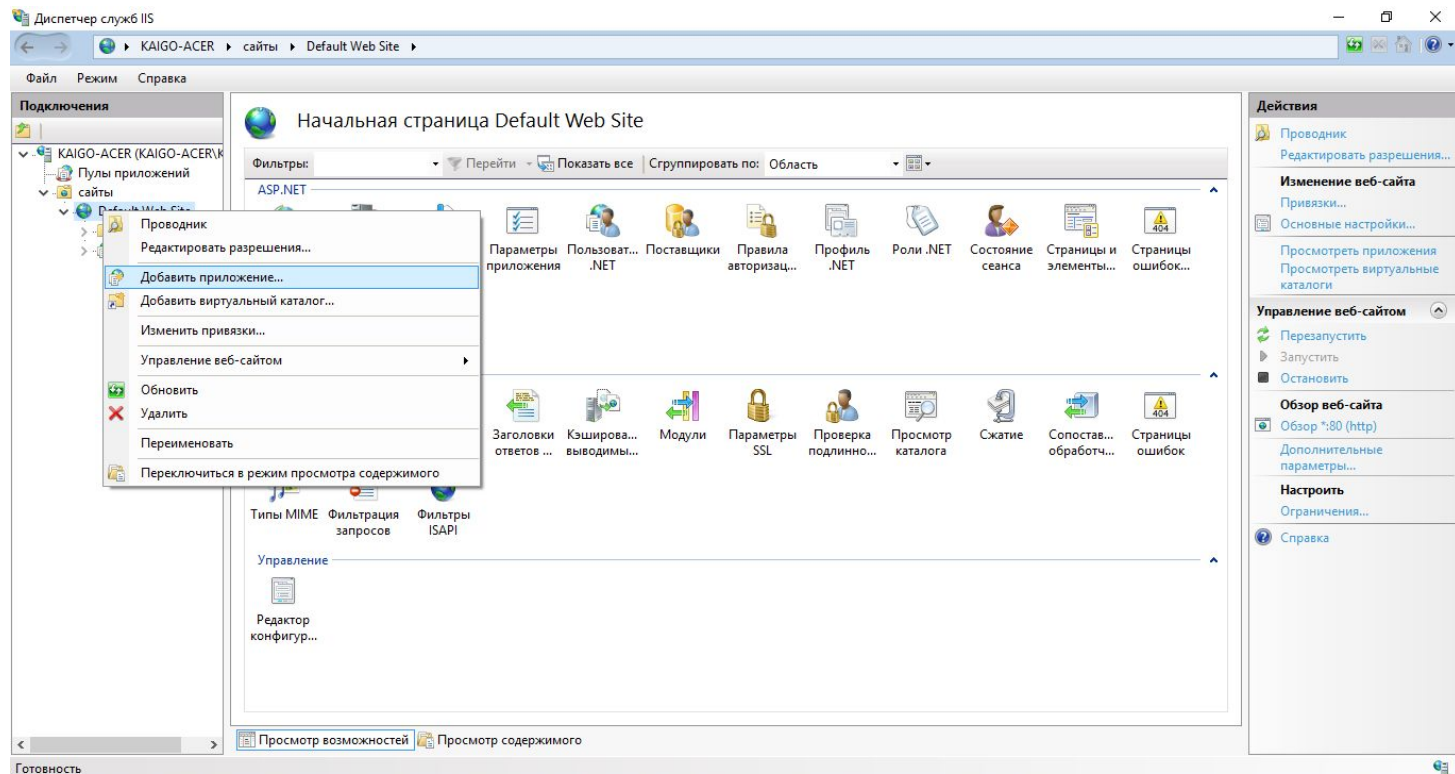
События ASP.NET MVC

```
// Это позволяет создавать объект доступные для всех экземпляров
// класса HttpApplication
protected void Application_Start()
{
    Log.Info("Log from Application Start");
    AreaRegistration.RegisterAllAreas();
    RouteConfig.RegisterRoutes(RouteTable.Routes);
}

// Наступает, когда приложение получает запрос.
// Первый раз это событие наступает для запроса страницы
// Когда пользователь вводит URL
protected void Application_BeginRequest()
{
    Log.Info("Log from begin request");
}

// Последнее событие, которое наступает для запроса к приложению
protected void Application_EndRequest()
{
    Log.Info("Log from end request");
}
```


Хостинг в IIS



Dependency Injection

Unity



Castle Windsor



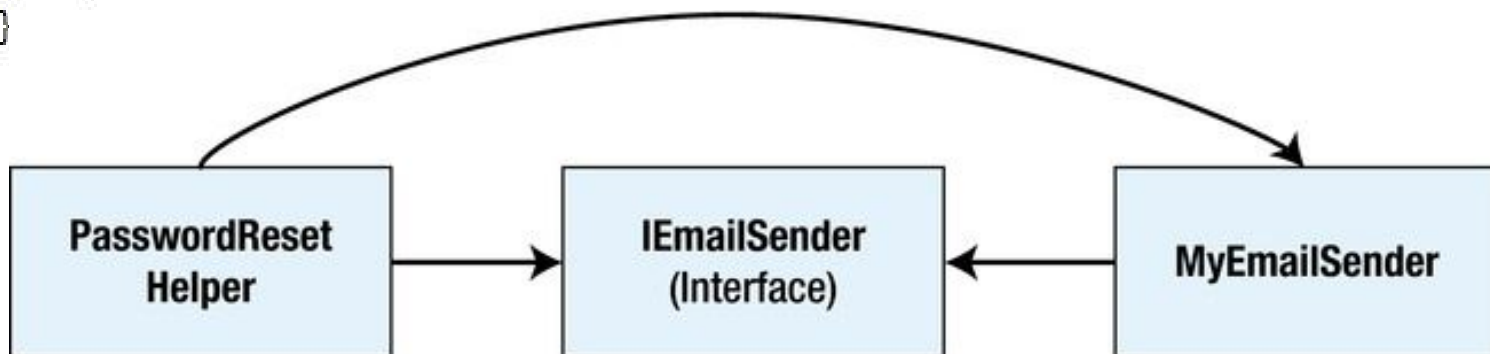
Dependency Injection: идея



Проблема



```
public class PasswordResetHelper
{
    public void ResetPassword()
    {
        IEmailSender mySender = new MyEmailSender();
        //...вызов методов интерфейса для конфигурации информации по имейлу
        mySender.SendEmail();
    }
}
```



Решение

```
public class PasswordResetHelper
{
    private readonly IEmailSender _emailSender;

    public PasswordResetHelper(IEmailSender emailSenderParam)
    {
        _emailSender = emailSenderParam;
    }

    public void ResetPassword()
    {
        // ...вызов методов интерфейса для конфигурации информации по имейлу...
        _emailSender.SendEmail();
    }
}
```



constructor injection

setter injection

Вопрос

Как создавать экземпляр нашего интерфейса, не порождая новых зависимостей?

Создание классов для проекта

```
public class Issue
{
    public int IssueId { get; set; }

    public string Title { get; set; }

    public string Description { get; set; }

    public decimal Price { get; set; }

    public DateTime? BeginDate { get; set; }

    public DateTime? EndDate { get; set; }

    public IssueStatuses Status { get; set; }
}

public enum IssueStatuses
{
    New,
    Open,
    InProgress,
    CodeReview,
    ReOpened,
    Resolved,
    Closed
}
```

```
public class IssuePriceCalculator
{
    public decimal PriceIssues(IEnumerable<Issue> issues)
    {
        return issues.Sum(i => i.Price);
    }
}

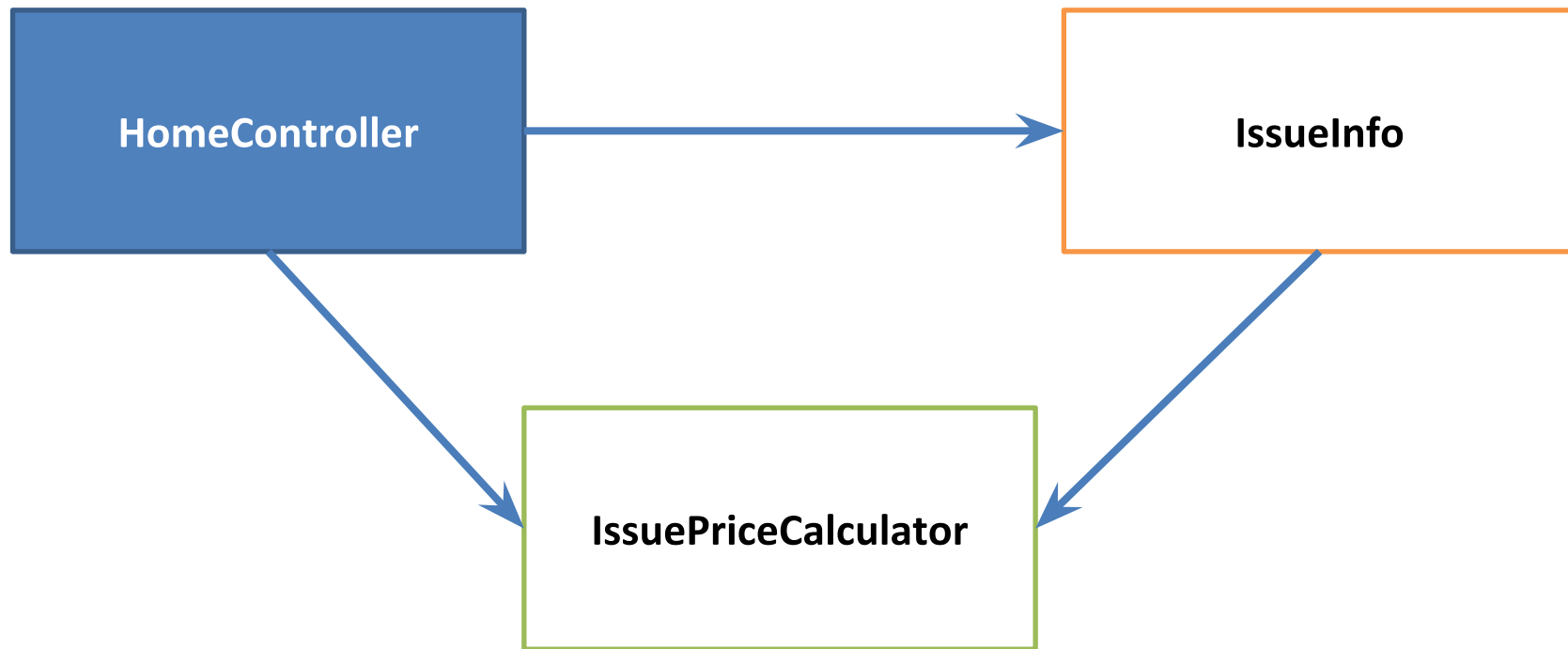
public class IssueInfo
{
    private readonly IssuePriceCalculator _calc;

    public IssueInfo(IssuePriceCalculator calcParam)
    {
        _calc = calcParam;
    }

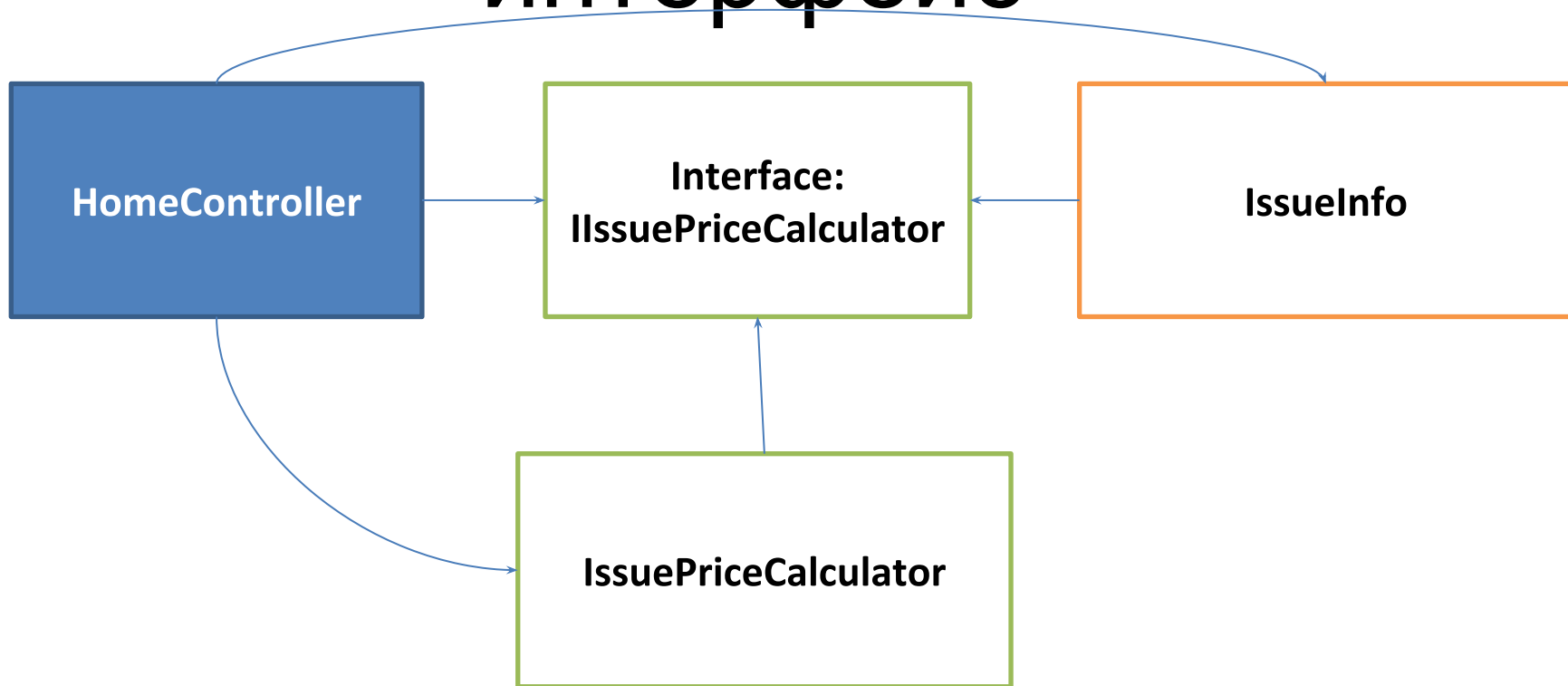
    public IEnumerable<Issue> Issues { get; set; }

    public decimal CalculateIssueTotal()
    {
        return _calc.PriceIssues(Issues);
    }
}
```

Проблема: Сильное сопряжение



Решение: добавляем интерфейс



Ninject

Lesson1Smar - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test ReSharper Analyze Window Help

Debug Any CPU Google Chrome

NuGet - Solution x NLog.config RouteConfig.cs Global.asax.cs Index.cshtml HomeController.cs

Browse Installed Updates 2 Consolidate

Search: ninject Include prerelease Package source: nuget.org

Ninject by Ninject Project Contributors, 4,18M downloads v3.2.2
IoC container for .NET

Ninject.Web.Common by Ninject Project Contributors, 2,12M downloads v3.2.3
Bootstrapper for web projects

Ninject.MVC5 by Remo Gloor, Ian Davis, 658K downloads v3.2.1

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Do not show this again

Manage Packages for Solution

Ninject

Version(s) - 0

<input type="checkbox"/> Project ^	Version
<input type="checkbox"/> Lesson1Smar	
<input type="checkbox"/> Lesson1Smar.Tests	

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'Lesson1Smar' (2 projects)

- Lesson1Smar
 - Properties
 - References
 - App_Data
 - App_Start
 - Controllers
 - Models
 - Views
 - Global.asax
 - NLog.config
 - NLog.xsd
 - packages.config
 - Web.config

Assembly Explorer Build & Run Solution Explorer Team Explorer

Properties

Output

Show output from: Debug

```
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-131350409599212598): Loaded 'C:\Windows\Microsoft.Net\assembly\DAL_MSIL\System.Web.Mobile\v4.0.4.0.0__b03f5f7f11d50a3a\System.Web.Mobile.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-131350409599212598): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Mobile\v4.0.4.0.0__b03f5f7f11d50a3a\System.Web.Mobile.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-131350409599212598): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Mobile\v4.0.4.0.0__b03f5f7f11d50a3a\System.Web.Mobile.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-131350409599212598): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Mobile\v4.0.4.0.0__b03f5f7f11d50a3a\System.Web.Mobile.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-131350409599212598): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Mobile\v4.0.4.0.0__b03f5f7f11d50a3a\System.Web.Mobile.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-131350409599212598): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Mobile\v4.0.4.0.0__b03f5f7f11d50a3a\System.Web.Mobile.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-131350409599212598): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Mobile\v4.0.4.0.0__b03f5f7f11d50a3a\System.Web.Mobile.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-131350409599212598): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Mobile\v4.0.4.0.0__b03f5f7f11d50a3a\System.Web.Mobile.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-131350409599212598): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Mobile\v4.0.4.0.0__b03f5f7f11d50a3a\System.Web.Mobile.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-131350409599212598): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Mobile\v4.0.4.0.0__b03f5f7f11d50a3a\System.Web.Mobile.dll'.
The program '[19036] iisexpress.exe' has exited with code -1 (0xffffffff).
```

Package Manager Console Output Find Results 1 Build Results

Ready Publish

Основные этапы с функционалом Ninject

- `IKernel ninjectKernel = new StandardKernel();`
- `ninjectKernel.Bind<IIssuePriceCalculator>().To<LinqIssuePriceCalculator>();`
- `IIssuePriceCalculator calc = ninjectKernel.Get<IIssuePriceCalculator>();`

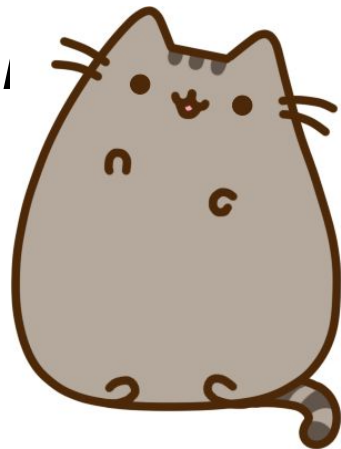
Создание DR (Dependency Resolver)

Что произошло?

1. MVC фреймворк получил запрос и выяснил, что запрос предназначен для контроллера Home.
2. MVC фреймворк попросил наш пользовательский DR класс создать новый экземпляр класса HomeController, указав класс, используя параметр Type метода GetService.
3. Наш DR попросил Ninject создать новый класс HomeController при помощи передачи объекта Type методу TryGet.
4. Ninject изучил конструктор HomeController и обнаружил, что он требует реализацию `IIssuePriceCalculator`.
5. Ninject создает экземпляр класса **LinqIssuePriceCalculator** и использует его для создания нового экземпляра класса HomeController.
6. Ninject передает вновь созданный экземпляр HomeController пользовательскому DR, который возвращает его MVC фреймворку. MVC фреймворк использует экземпляр контроллера для обработки запроса.

Возможности

- *Цепочка зависимостей*
- *Управление жизненным циклом объекта*
- *Конфигурация значений , пров
конструктора*



Unit-тестирование

```
[TestMethod]
public void TotalPriceInClosedStatus()
{
    Issue[] _issues =
    {
        new Issue {Title = "Issue1", Description = "It is good Issue #1", Price = 20},
        new Issue {Title = "Issue2", Description = "It is good Issue #2", Price = 20},
        new Issue {Title = "Issue3", Description = "It is good Issue #3", Price = 30},
    };

    var calc = getTestObject();
    IssueInfo info = new IssueInfo(calc) { Issues = _issues };
    var result = info.CalculateIssueTotal();
    Assert.AreEqual(525, result);
}
```

Moq

The screenshot shows the Visual Studio IDE with the NuGet Package Manager window open. The 'Browse' tab is selected, and the search filter is set to 'Moq'. The package list on the left shows 'Moq' by Daniel Cazzulino (11.3M downloads, v4.7.8), 'Moq.Contrib' by kzu.net (66K downloads, v0.3.0), and 'Grace Moq' by Ian Johnson (6.73K downloads, v2.4.2). The right pane shows the details for 'Moq', including a table of installed versions for the selected projects.

IssueInfo.cs UnitTest1.cs HomeController.cs IssuePriceCalculator.cs LinqIssuePriceCalculator.cs **NuGet - Solution** Build & Run

Browse Installed Updates **2** Consolidate Manage Packages for Solution

Moq Include prerelease Package source: nuget.org

Moq by Daniel Cazzulino, kzu, **11,3M** downloads v4.7.8
Moq is the most popular and friendly mocking framework for .NET

Moq.Contrib by kzu.net, **66K** downloads v0.3.0
Contributed features and third-party integration for Moq.

Grace Moq by Ian Johnson, **6.73K** downloads v2.4.2

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Do not show this again

Moq

Version(s) - 0

<input checked="" type="checkbox"/>	Project ^	Version
<input checked="" type="checkbox"/>	DemoLesson	
<input checked="" type="checkbox"/>	DemoLesson.Tests	

Build Results

Group by: Project and NameSpace

Доступ к данным



Database-First **Model-First** **Code-First**

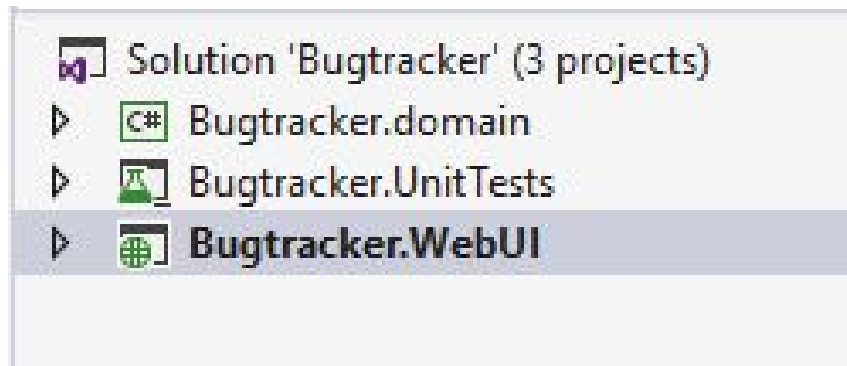
Резюме

1. Логирование это полезно
2. Зависимость между классами надо уменьшать
3. Moq довольно интересный инструмент, с которым познакомимся дальше
4. Работа с EntityFramework довольно проста, но надо учитывать нюансы
5. ASP.NET MVC удобный инструмент

Демонстрация: Bugtracker



Создание Solution



Создание модели

```
public class Issue
{
    public int IssueId { get; set; }

    public string Title { get; set; }

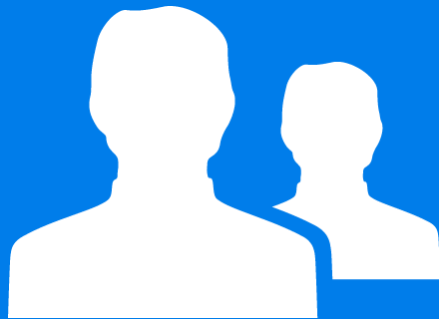
    public string Description { get; set; }

    public DateTime BeginDate { get; set; }

    public DateTime EndDate { get; set; }

    public IssueStatuses Status { get; set; }
}
```

```
public enum IssueStatuses
{
    New,
    Open,
    InProgress,
    CodeReview,
    ReOpened,
    Resolved,
    Closed
}
```



Наши вакансии на
сайте fogsoft.ru
Ждем ваших резюме!