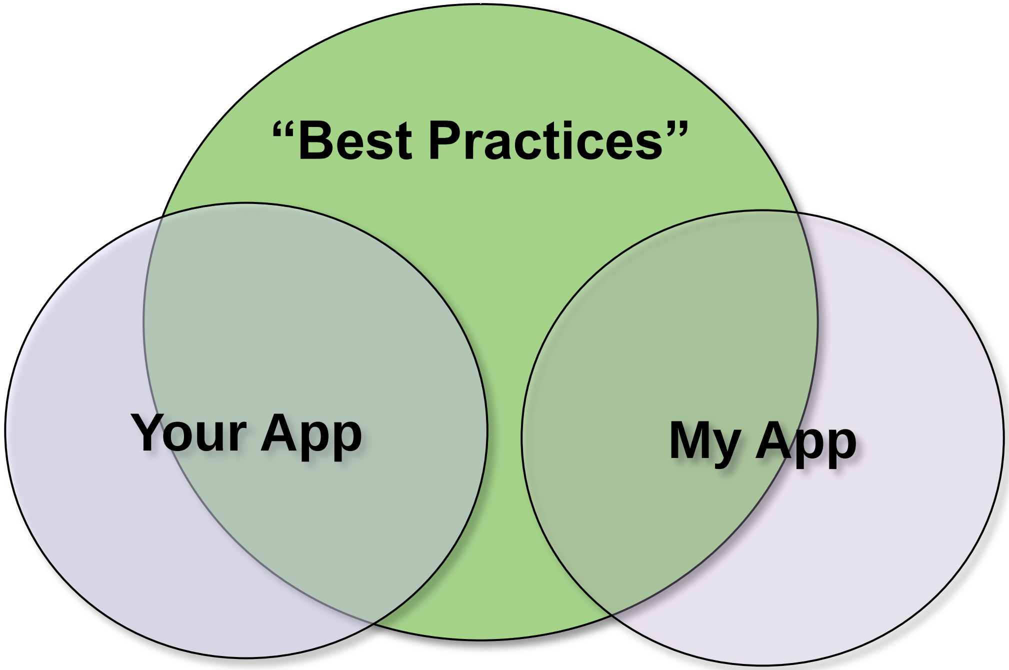
Best Practices for ASP.NET MVC

Guiding Principles

Overview

- Views
- Controllers
- Security
- General tips

Context



View Code

- **Avoid spaghetti code**
  - Use partial views to reduce complexity
  - Use HTML helpers to encapsulate logic

```
<%
    if (Request.IsAuthenticated) {
%>
        Welcome <b><%= Html.Encode(Page.User.Identity.Name) %></b>!
        [ <%= Html.ActionLink("Log Off", "LogOff", "Account") %> ]
<%
    }
    else {
%>
        [ <%= Html.ActionLink("Log On", "LogOn", "Account") %> ]
<%
    }
%>
```

Use Strongly Typed Views

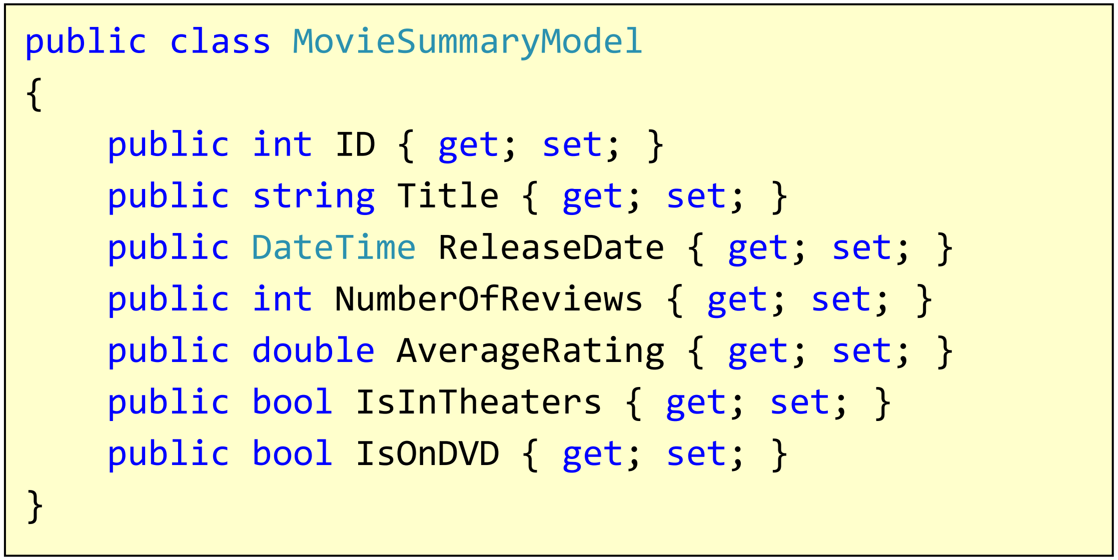
- **Easier to author with Intellisense**
- **Easier to refactor and find errors**
- **Build your views to track down problems**
  - Aspnet_compiler
  - Web Deployment Projects
  - Set MvcBuildViews to true (slower build times)

```
<%@ Page Title="" Language="C#"
         MasterPageFile="~/Views/Shared/Site.Master"
         Inherits="System.Web.Mvc.ViewPage<IEnumerable<MovieSummary>>"
%>
```

```
<MvcBuildViews>true</MvcBuildViews>
```

View Models

- **A view model is dedicated to one or more views**
  - Takes pressure off your business objects and entities
  - Easier to perform calculated values outside of view
  - Easy to unit test



```
public class MovieSummaryModel
{
    public int ID { get; set; }
    public string Title { get; set; }
    public DateTime ReleaseDate { get; set; }
    public int NumberOfReviews { get; set; }
    public double AverageRating { get; set; }
    public bool IsInTheaters { get; set; }
    public bool IsOnDVD { get; set; }
}
```

Unobtrusive JavaScript

- **Remove all signs of JavaScript from a view**
- **Advantages**
  - View is strictly presentation
  - Allows you to focus on script code
- **Disadvantages**
  - Additional file(s) to download
  - Can combine scripts (http://aspmvccombine.codeplex.com/)

```
$(function() {
    $("#loginImage").click(login);
}
```

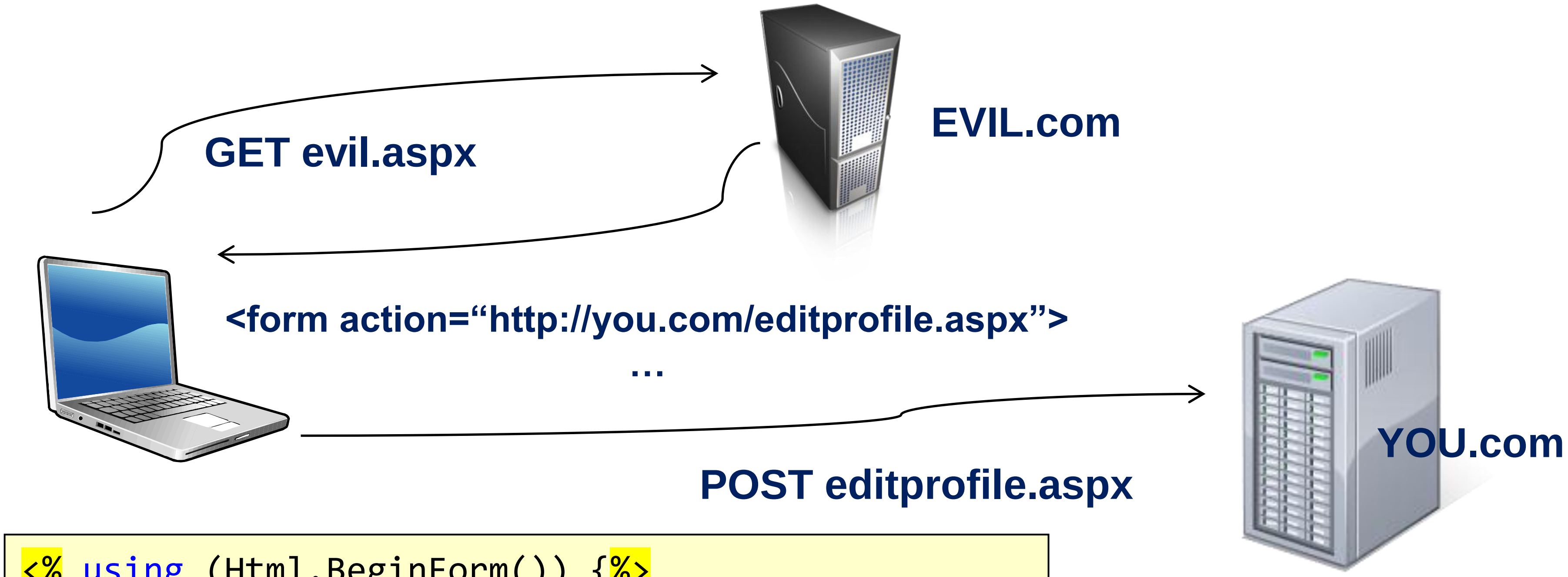
Security: XSS

- **Prevent cross-site scripting attacks**
  - Be very, very, careful turning off input validation
  - Microsoft Anti-Cross Site Scripting Library (http://tinyurl.com/cpu7g4)
- **HTML Encoding**
  - Use HTML.Encode as a default
  - Only skip encoding when you are writing out HTML (like a CMS)

```
[ValidateInput(false)]
[AcceptVerbs(HttpVerbs.Post)]
public ActionResult Create(ToDo newItem)
{
    // ...
}
```

Security: CSRF

- Use anti-forgery tokens on authenticated actions



GET evil.aspx

EVIL.com

<form action=“http://you.com/editprofile.aspx”>

…

POST editprofile.aspx

YOU.com

```
<% using (Html.BeginForm()) {%>
    <% = Html.AntiForgeryToken() %>
```

```
[ValidateAntiForgeryToken]
[AcceptVerbs(HttpVerbs.Post)]
public ActionResult Edit(int id, Profile profile)
   …
```

Security: Model Binding

- **Be wary of binding without constraints**
  - Use Include and Exclude parameters

```
public class GamerProfile
{
    public int ID { get; set; }
    public string Nickname { get; set; }
    public string Address { get; set; }
    public string City { get; set; }
    public string  Street { get; set; }
    public int GamerScore { get; set; }
}
```

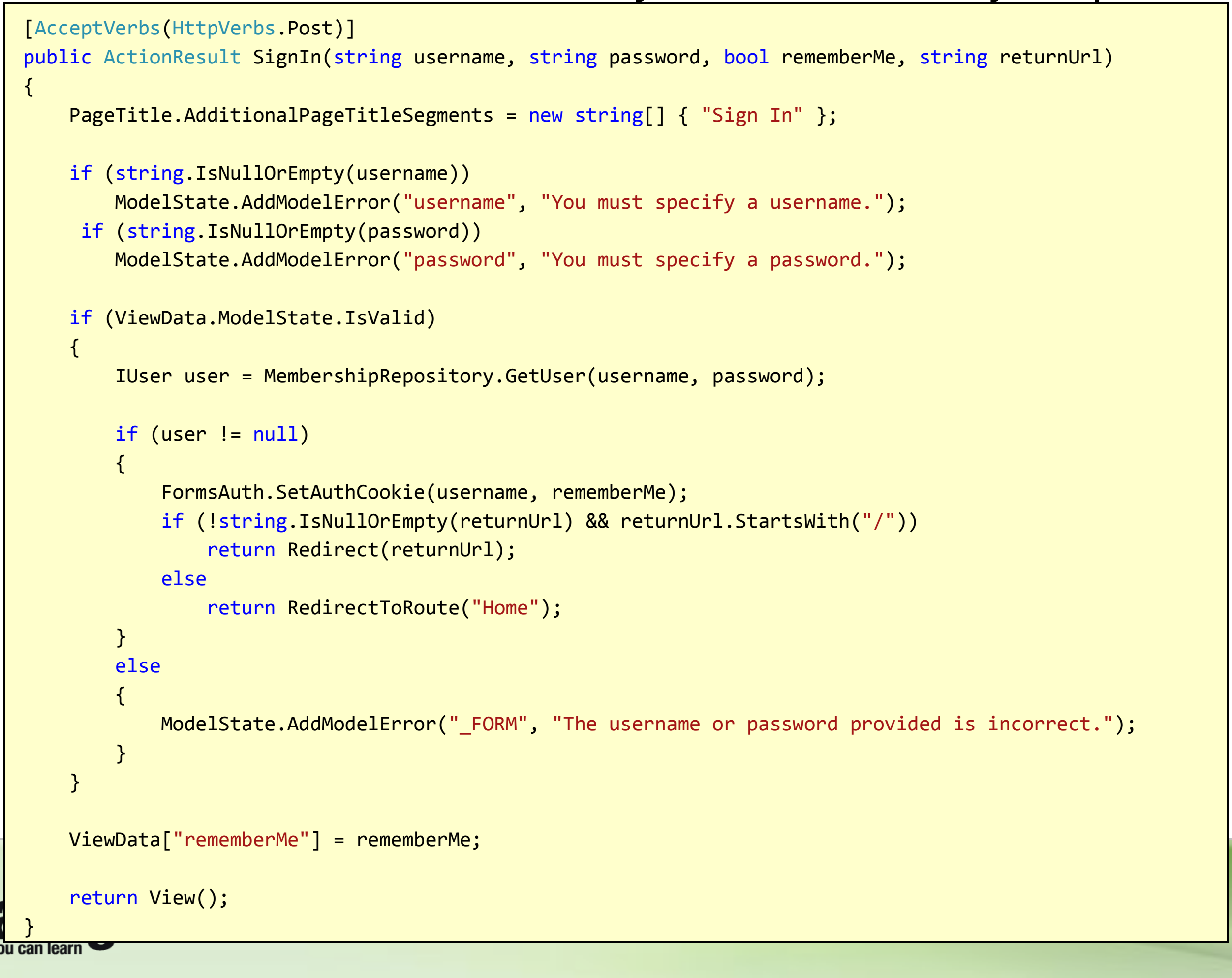
```
[AcceptVerbs(HttpVerbs.Post)]
public ActionResult Edit(GamerProfile profile)
{
    profile.GamerScore = GetCurrentGamerScore();
    UpdateModel(profile);

    // ...

    return View(profile);
}
```

Keep Controllers Focused

- **Controllers are in a position of power**
  - Actions and controllers naturally collect too many responsibilities



```
[AcceptVerbs(HttpVerbs.Post)]
public ActionResult SignIn(string username, string password, bool rememberMe, string returnUrl)
{
    PageTitle.AdditionalPageTitleSegments = new string[] { "Sign In" };

    if (string.IsNullOrEmpty(username))
        ModelState.AddModelError("username", "You must specify a username.");
     if (string.IsNullOrEmpty(password))
        ModelState.AddModelError("password", "You must specify a password.");

    if (ViewData.ModelState.IsValid)
    {
        IUser user = MembershipRepository.GetUser(username, password);

        if (user != null)
        {
            FormsAuth.SetAuthCookie(username, rememberMe);
            if (!string.IsNullOrEmpty(returnUrl) && returnUrl.StartsWith("/"))
                return Redirect(returnUrl);
            else
                return RedirectToRoute("Home");
        }
        else
        {
            ModelState.AddModelError("_FORM", "The username or password provided is incorrect.");
        }
    }

    ViewData["rememberMe"] = rememberMe;

    return View();
}
```

General Tips

- **SRP – Single Responsibility Principle**
  - Smaller classes, smaller methods
- **DRY – Don’t repeat yourself**
  - Refactor duplicate code
- **TDD**
  - Can discover responsibilities, duplication, and dependencies
- **Clean Code – Robert C. Martin**
- **Leverage the work of others**
  - MVC Contrib

Summary

Simplicity