# Installation

1. If you don’t already have Visual Studio, MonoDevelop, or Xamarin Studio installed, install one of the following:
   1. F# Tools for Visual Studio Express 2012 for Web (and walk through of Type Providers) <http://blogs.msdn.com/b/fsharpteam/archive/2012/09/12/announcing-the-release-of-f-tools-for-visual-studio-express-2012-for-web.aspx>
   2. Xamarin Studio - <http://xamarin.com/studio>
   3. MonoDevelop - <http://monodevelop.com/>
2. Install Templates:
   1. If you installed Visual Studio in step 1, get the project templates by following the instructions at <http://bloggemdano.blogspot.com/2011/12/building-aspnet-mvc-4-solution-with-f.html>
   2. If you installed Xamarin Studio or MonoDevelop in step 1, get the project template by following the instructions at <http://bloggemdano.blogspot.com/2012/12/using-new-aspnet-mvc-4-template-in.html>
3. Create your first F# ASP.NET MVC 4 project from one of the templates and verify that it compiles/runs.

# Exercise

1. Create a Razor F#/C# ASP.NET MVC solution from the Empty Project template that is included with the F#/C# MVC 4 template that was installed in step 2 of the Installation section of this document.
2. Add a new folder named Guitars under the Views folder (in the C# project if you are using VS).
3. Add a new View named Index.cshtml within the new Guitars folder. Unselect the “User a layout or master page” setting before clicking Add.
4. Replace the content of the Index.cshtml file with the content found at <https://raw.github.com/dmohl/fs-web-cloud-mobile/master/Ch%201/FsMvc4_OO_Example/FsCsMvc44/FsCsMvc44Web/Views/Guitars/Index.cshtml>
5. Add a new .fs file (in the F# project if using VS) named Guitar.fs and add the following F# code to it.

namespace FsWeb.Models

open System.ComponentModel.DataAnnotations

type Guitar() =

[<Required>] member val Name = "" with get, set

1. Add a new .fs file (to the F# project if using VS) named GuitarsController.fs and add the following F# code to it (Note: You may need to change the location of the file that will be used to store the guitar names. You may also need to grant appropriate permissions to allow the file to be created and updated by the web app).

namespace FsWeb.Controllers

open System

open System.Web.Mvc

open FsWeb.Models

[<HandleError>]

type GuitarsController() =

inherit Controller()

let guitars = match System.IO.File.Exists @"c:\temp\Guitars.txt" with

| true ->

System.IO.File.ReadAllText(@"c:\temp\Guitars.txt").Split(',')

|> Array.map (fun x -> Guitar(Name = x))

| \_ -> [||]

member this.Index () = guitars |> this.View

1. Run the solution and navigate to ../Guitars. You should see a simple screen with a search box and a header that states “Guitars”.
2. Stop the web application from running and add a new View to the Guitars folder (in the C# project if using VS) named Create.cshtml. Replace the content of this new file with the content from <https://raw.github.com/dmohl/fs-web-cloud-mobile/master/Ch%201/FsMvc4_OO_Example/FsCsMvc44/FsCsMvc44Web/Views/Guitars/Create.cshtml>
3. Now update the GuitarsController.fs file to have the following F# code. The highlighted code is new:

namespace FsWeb.Controllers

open System

open System.Web.Mvc

open FsWeb.Models

[<HandleError>]

type GuitarsController() =

inherit Controller()

let guitars = match System.IO.File.Exists @"c:\temp\Guitars.txt" with

| true ->

System.IO.File.ReadAllText(@"c:\temp\Guitars.txt").Split(',')

|> Array.map (fun x -> Guitar(Name = x))

| \_ -> [||]

member this.Index () = guitars |> this.View

[<HttpGet>]

member this.Create () = this.View()

[<HttpPost>]

member this.Create (guitar : Guitar) : ActionResult =

let isNameValid = not (String.IsNullOrEmpty(guitar.Name))

match base.ModelState.IsValid, isNameValid with

| false, false | true, false | false, true ->

upcast this.View guitar

| \_ ->

let result = guitars |> Array.fold(fun acc x -> acc + x.Name + ",") ""

System.IO.File.WriteAllText(@"c:\temp\Guitars.txt", result + guitar.Name)

upcast base.RedirectToAction("Index")

1. You can now run the solution again. Navigate to /Guitars/Create page and add a new guitar to try out your creation. The full Visual Studio example (for both C# and F#) can be found in the ..\MVC Lab\Guitar Example\ folder.

# Extra Credit Exercises

1. Create a simple web page that displays a list of contacts. Step by step instructions for MonoDevelop are available at <http://bloggemdano.blogspot.com/2012/12/using-new-aspnet-mvc-4-template-in.html>
2. Asynchronously read the contents of 3 different web pages and determine how many characters are on each web page. Initiate this check from a button on the web page and display the character count as a result.

# F# Cheat Sheets

DZone Reference Card: <http://cdn.dzone.com/sites/all/files/refcardz/rc081-010d-fsharp.pdf>

The F# Handout: <http://www.hamletdarcy.com/files/2008/TheFSharpHandout.pdf>