* Introduction
* What is OWIN
  + Open Web Interface for .NET(OWIN)
  + Standard interface between .NET web servers and web applications
  + Specification that defines how you can abstract away web server in a efficient way
  + Build web applications without caring about how it will be hosted
  + .NET + Web = IIS
  + Abstract away web server using a single delegate and dictionary
  + Ex)
    - using AppFunc = Func<IDictionary<string, object>, Task>;
    - var f = new AppFunc(environment = > { return Task.FromResult(null); });
  + IDictioanry<string,object> contains all information passed to server in http request
    - Also any functionality that web server offers to web application
* The Parts of OWIN
  + Owin defines parts/actors that are involved in handling the incoming request and return response
  + Host
    - Some process that host all other parts
    - Starting everything up
  + Server
    - Responsible for accepting incoming request and sending back responses
  + Middleware pipeline
    - Piece of code that request passes on the way to and from the application
    - Can be used to inspect and modify incoming request and outgoing response
  + Application
    - Responsible for generating response
  + Use web framework to abstract away middleware and application
* The Flow
* Basic Keys in the Environment …
  + A lot of keys, prefix with owin.
  + Ex) owin.RequestPath
* Project Katana
  + Microsoft implementation of OWIN specification
  + Start new ASP.NET web application
  + Choose MVC template
  + Change authentication(individual user accounts)
    - Will include owin packages
* Summary
  + OWIN is specification for webserver abstraction
  + Open Web Interface for .NET(OWIN)
  + Katana is Microsoft implementation of OWIN
* Introduction
  + Building a simple OWIN Pipeline
* Creating an OWIN Based Appli…
  + Create an empty project
  + Install package ‘Microsoft.Owin.Host.SystemWeb’
    - Microsoft.Owin and Owin will also get installed
    - Microsoft.Owin contains katana
  + Add class Startup
    - Create entry point
    - Ex)
    - public class Startup
    - {
    - public static void Configuration(IAppBuilder app)
    - {
    - app.Use(async (ctx, next) => { await ctx.Response.WriteAsync(“Hello World”});
    - }
    - }
  + IAppBuilder is used to add middleware to OWIN pipeline
  + app.Use((IOwinContext, Func<Task>)
* Adding Multiple Middlewares t…
  + Need to make sure rest of pipeline executes
  + Ex)
    - app.Use(async (ctx, next) => {
    - Debug.WriteLine(ctx.Request.Path);
    - await next();
    - });
  + Next references to the next middle ware in the pipeline
* Summary
  + NuGet the required host
  + Add public class Startup
  + Add Configuration(IAppBuilder app)
  + Add middlewares
* Introduction
* Creating the Middleware Class