* Introduction
  + Understanding MVC pipeline
  + ASP.NET Life Cycle
    - Begin request
    - Resolve request cache
    - Map request handler
    - Acquire request state
    - Request handler execute(controllers & actions)
    - Update request cache
    - Log request
    - End request
  + MVC is supported by ASP.NET platform
* Who is this course for
  + Working knowledge of MVC
  + Concept more than code
  + Existing developers looking for deeper understanding of MVC
  + Developers migrating from another
  + Developers who started their .NET career with MVC
  + Never exposed to the larger ASP.NET platform
* Defining the Request Life Cycle
  + Step/events used to handle request or change in application
  + MVC life cycles
    - Application life cycle
    - Request life cycle
  + Application life cycle
    - Application starts running till it stops
    - Application start and end events in startup file
  + Request life cycle
    - https request is handled by application
* Overview of the MVC Request …
  + Request -> Routing -> Controller Initialization -> Action Execution -> Result Execution -> View Engine -> Result Execution -> Response
  + Routing
    - URL routing module: matching incoming url to routes we define in app
    - MVC Route Handler: retrieves instance of mvc httphandler
    - MVC HttpHandler: initializing and executing controller
  + Controller initialization
    - Controller Factory
    - Activator
    - Dependency Resolution
  + Action Execution
    - Model Binding
    - Action Filters(Action Executing)
    - Action Execution
    - Action Filters(Action Executed)
    - Action Result
  + Result Execution
    - Result filter(result executing)
    - Invoke action result
  + View Engine
    - Find and render view
    - will be skipped if not returning view
  + Result Execution
    - Result filter(result executed)
  + Modules are dot net components that can hook into application life cycle and add functionality
* Webforms and MVC
  + ASP.NET webforms has its own lifecycle
  + Handling a request
    - To a web form, general map to file on disk
    - For MVC, maps to controller action method
  + Both implemented through Http Handler
  + ASP.NET Platform
    - One platform, multiple implementations
    - Http Handler
      * MVC Framework
      * Web forms
      * Custom Implementation
* Summary
* Introduction
* Understanding Application Sta..
  + MVCApplication in Global.asax files
    - Inherits from HttpApplication class
    - Life cycle events get inherited
  + Application start
    - An event fires when first request is received
    - Can be used to run initial configuration code
    - Registers routes
    - Ex
    - protected void Application\_Start()
    - {
    - AreaRegistration.RegisterAllAreas();
    - ….
    - RouteConfig.RegisterRoutes(RouteTable.Routes);
    - …
    - }
  + Registering Routes
    - Adds routes we define to static collection on route table class
    - Collection of routes url routing module will try to match to incoming urls
    - Each routes need an associated route handler class
      * Ex) defaults: new { controller = “Home”, action = “Index”, id = UrlParameter.Optional }
    - Route handler provide asp.net with an http handler that will process incoming request after being matched to a route
    - Routes are registered before any other lifecycle event happens
    - Ex)
    - public static void RegisterRoutes(RouteCollection routes)
    - {
    - routes.MapRoute(
    - name: “Default”,
    - url: “{controller}/{action}/{id}”
    - defaults: new { controller = “Home”, action = “Index”, id = UrlParameter.Optional }
    - );
    - }
  + Application end
    - Event that fires when application ends
    - Not guaranteed to fire when application crashes
    - Not a great option for handling application errors
* Demo - Application Start and E..
  + MapRoute(): assign routes to route table collection
    - Route table collection will be examined by url routing module
    - Behind the scenes mvc creates a new route and mvc handler and adds them to the collection
  + Every route needs an associated route handler class to go with it
    - Purpose is to retrieve the right http handler for request
  + Http handler
    - Is what will execute to generate response
  + MVC is an open source project, you can check the source code online
  + You can do it yourself
  + Ex instead of using routes.MapRoute())
    - Route myRoute = new Route(“{controller}/{action}/{id}”, new RouteValueDictionary{ {“controller”, “Home”}, {“action”, “Index”}, {“id”: “1” }},
    - New MvcRouteHandler());
    - routes.Add(myRoute);
* Configurations with the PreAp..
  + PreApplicationStart
    - Another option for running initial configuration code
    - Applied through an attribute
    - Often used to register modules
* Demo-PreApplicationStartMe..
  + LogModule hooks into log event of request life cycle(near end of lifecycle)
  + Module needs to be registered with application in Global.asax
  + PreApplicationStart attribute has to be defined at the assembly level
  + Ex
    - using …
    - [assembly: PreApplicationStartMethod(typeof(MvcApplication), “Register”)]
    - namespace HealthAssist
    - {
    - public class MvcApplication : System.Web.HttpApplication
    - {
    - …
    - }
  + Then add Register() method to register log module
  + Ex
    - public static void Register()
    - {
    - HttpApplication.RegisterModule(typeof(LogModule));
    - }
    - protected void Application\_Start()
    - {
    - …
    - }
* Exploring the Request Life Cyc..
  + BeginRequest
  + AuthenticateRequest
  + AuthorizeRequest
  + ResolveRequestCache
    - URL Routing Module responds to PostResolveRequestCache, when it selects http handler for request
    - Choose handler
  + MapRequstHandler
    - Officiates handler is about to be chosen
  + AcquireRequestState
  + RequestHandlerExecute
    - MVC handler executes
    - MVC generates response
  + UpdateRequestCache
  + LogRequest
  + EndRequest
* Demo - The Request Life Cyc…
  + Application\_PostRequestHandlerExecute()
    - Where most MVC will execute(action methods, etc)
* Working with Events Across Fr..
  + Application life cycle is framework agnostic
    - Can access events in an asp.net application, mvc or not
* Demo - Events Across Framew..