

# ASP.NET MVC 3 Deep Dive

---

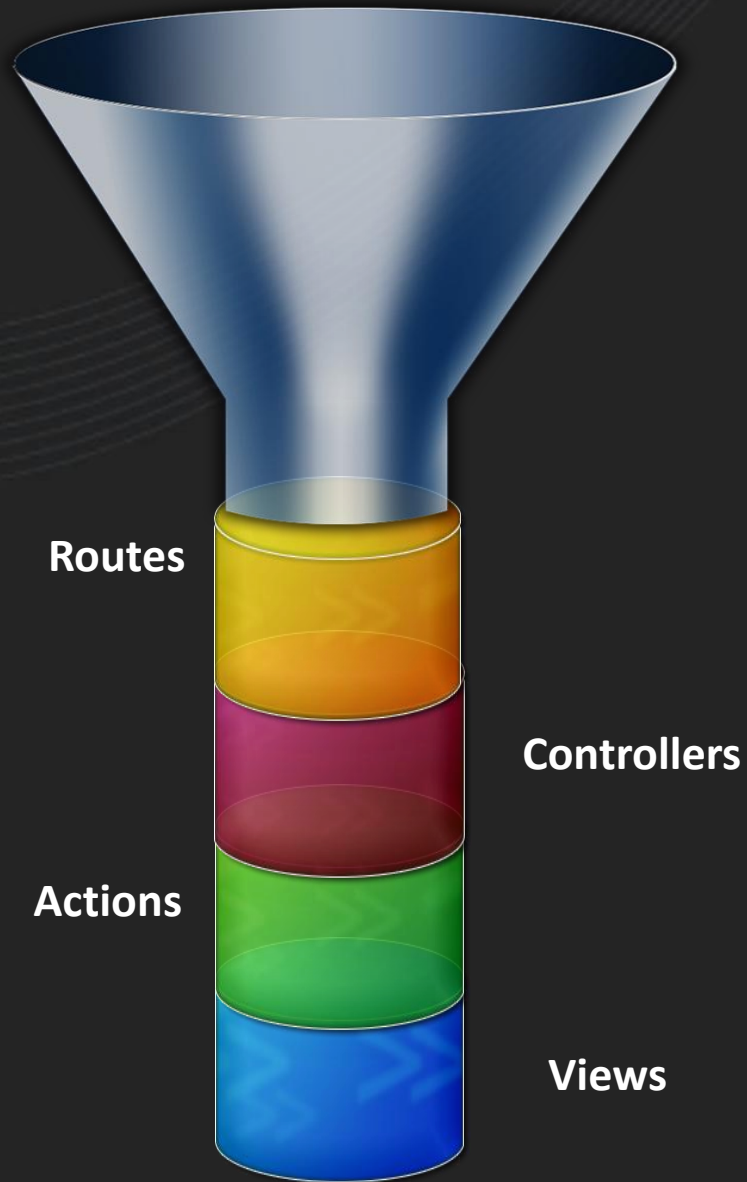
Hadi Hariri - JetBrains

@hhariri / <http://hadihariri.com>



# Agenda

---





Model



Controllers



Actions



Routing

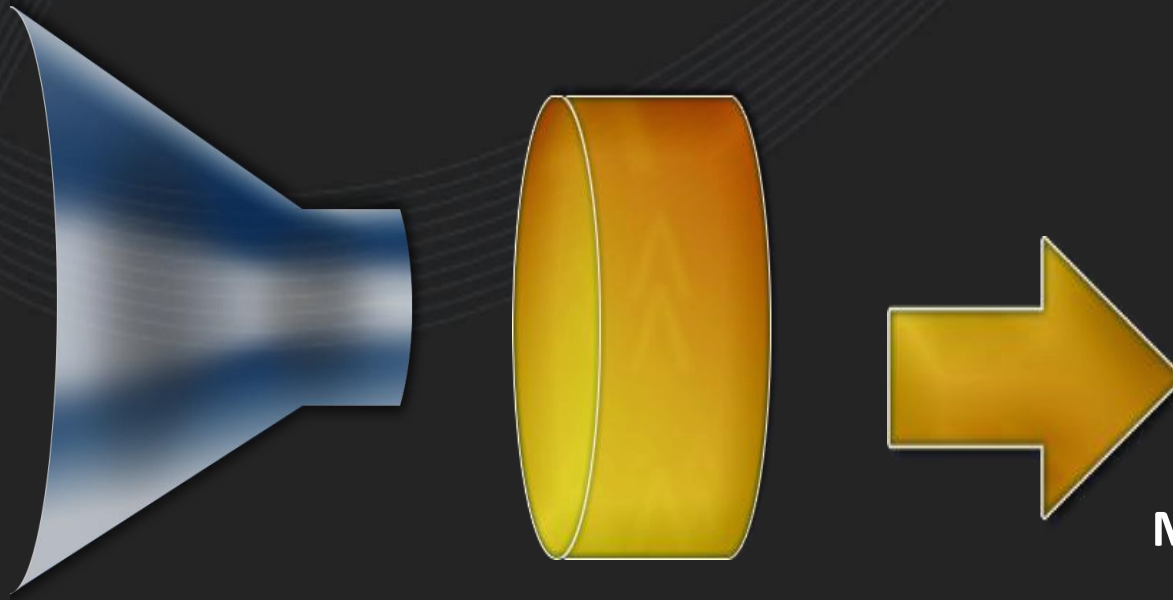


View

# Routes

---

**RouteData**



**MvcHandler**  
**IHttpHandler**

**System.Web.Routing**

```
var route = new  
Route("{controller}/{action}/{param}",  
    new MvcRouteHandler());
```

```
class Route: RouteBase {  
    string Url  
    RouteValueDictionary Defaults  
    RouteValueDictionary Constraints  
    RouteValueDictionary DataTokens  
    IRouteHandler RouteHandler  
}
```

# RouteCollection Extensions

---

**MapRoute(name, pattern, defaults. . .)**



# Routing

---

- Route Table: RouteTable
  - Collection of Routes
- Defined in Global.asax
- Ordered
  - More restrictive first
- Constraints
- Incoming and Outgoing
  - Html.ActionLink
  - Url.Action



# DEMO

---

Constraint and Verb Routing





# Controllers

---



**MvcRouteHandler**  
**IRouteHandler**



**ControllerFactory**  
**IControllerFactory**

```
public interface IControllerFactory
{
    IController CreateController(RequestContext requestContext, string controllerName);
    SessionStateBehavior GetControllerSessionBehavior(RequestContext requestContext, string controllerName);
    void ReleaseController(IController controller);
}
```



Delegate Creation to **IControllerActivator**

```
private class DefaultControllerActivator : IControllerActivator {  
    Func<IDependencyResolver> _resolverThunk;  
  
    public DefaultControllerActivator()  
        : this(null) {  
    }  
  
    public DefaultControllerActivator(IDependencyResolver resolver) {  
        if (resolver == null) {  
            _resolverThunk = () => DependencyResolver.Current;  
        }  
        else {  
            _resolverThunk = () => resolver;  
        }  
    }  
}
```

## Private Nested Class

```
public interface IController
{
    void Execute(RequestContext requestContext);
}
```



**ControllerBase: IController**



**Controller: ControllerBase, IDisposable, IAuthorizationFilter...**

How do we replace the `ControllerFactory`?



We don't...



We implement an `IDependencyResolver`



# IDependencyResolver

---

- Service Locator
- Default Implementation: DependencyResolver
- Override two methods:
  - GetService: Return single instance of service or null
  - GetServices: Return multiple instances of service or empty array
- Set Dependency Resolver when overriding





# DEMO

---

Injecting IoC's



# Simplifying IoC

---

- NuGet and WebActivator
- Startup Tasks for Packages



# DEMO

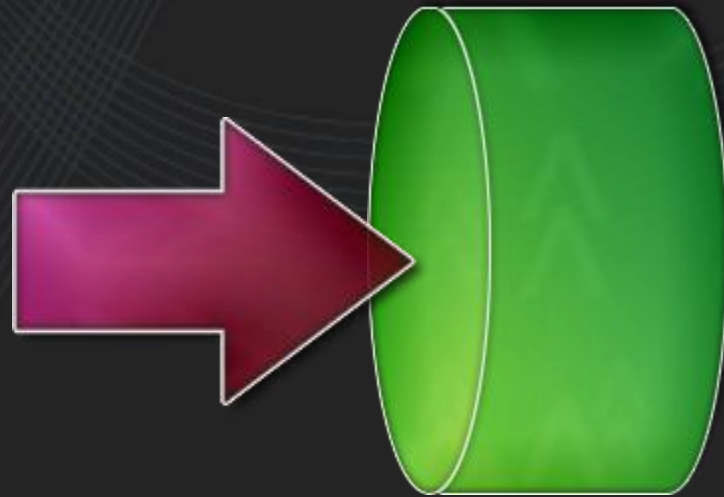
---

Simplifying IoC's



# Actions

---



**Find Method**  
**Bind Parameters**



**Execute Action**  
**& Filters**

**ControllerActionInvoker**  
**IActionInvoker**

# IActionInvoker

---

- `ControllerActionInvoker`
- `AsyncControllerActionInvoker: IAsyncActionInvoker`
- Executes Filters then Action, then Filters
- Override when providing custom Action Invocation



# Action Filters

---

```
[HttpPost|HttpGet|HttpPut|HttpDelete]
[ActionName("ActionName")]
[Authorize(Order, Roles, Users)]
[OutputCache(Duration, VarByName. . . )]
[HandleError(Order, View, Master, ExceptionType)]
public ActionResult SomeMethod(params. . . )
{
}
```



# Controlling Actions

`IActionFilter: Intercept Action`

`IAuthorizationFilter: Specify Permission`

`IExceptionFilter: Intercept Exceptions`

`IResultFilter: Intercept Result`

Mostly use in form of Attributes



# Using Filters

---

```
public abstract class Controller IFilter,  
    IAuthorizationFilter, IDisposable,  
    IExceptionHandler, IResultFilter {
```

*OnActionExecuting*

*OnActionExecuted*

*OnResultExecuting*

*OnResultExecuted*

*OnException*

*OnAuthorization*





# Global Filters and Providers

---

- Global Filters
  - Registered for all Actions and Controllers
- Filter Providers
  - Create Filters and apply them based on requests
  - Support for DI

# DEMO

---

Unknown Record, HTTP Auth, ContentTypes



# DEMO

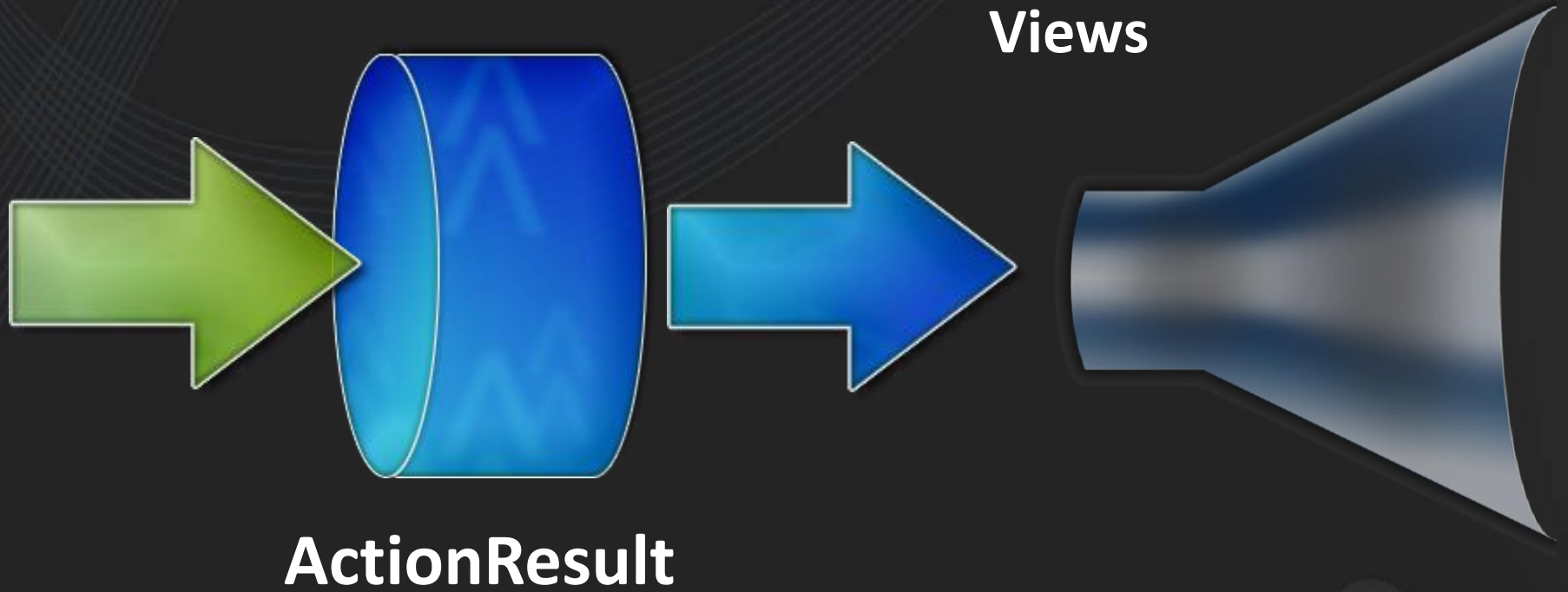
---

Custom Action Invokers: Controllerless Actions



# Returning Data

---



# Result Types

---

- ContentResult
- JsonResult
- PartialViewResult
- RedirectResult
- RedirectToRouteResult
- ViewResult
- EmptyResult



# Built-in Bindings

---

- Binding Method Parameters

- URL
- QueryString
- Post Fields

```
public void Action(int id,  
string value)
```

- Binding to Json (New in MVC 3)

- DefaultModelBinder: IModelBinder

```
public void Action(ComplexType  
complexType)
```

# Validation

---

- `ModelValidatorProvider`
  - Now discoverable via `DependencyResolver`
- `IValidatableObject`



# DEMO

---

Validatable object





# DEMO

---

Fluent Validation: R#Vault



# Model MetaData

---

- ModelMetadata
  - Associate MetaData with Model
  - Default is DataAnnotationsModelMetadata
- ModelMetadataProvider
  - Now hooks into DependencyResolver



# Views

---

- Ability to access `DependencyResolver`
- Can Build, Render and Test!



# Scaffolding



# T4

- Providing Custom Scaffolding
- Customize Globally
  - `%VisualStudioFolder%\Common7\IDE\ItemTemplates\CSharp\Web\MVC\CodeTemplates`
- Customize Project Level
  - Folder CodeTemplates and Subfolders AddController and AddView
- T4 Editors



# DEMO

---

T4



# More about Controllers



# DEMO

---

Async Controllers





# DEMO

---

## Stateless Controllers



# Summary

---

## Extensibility and Flexibility



# Contact me...



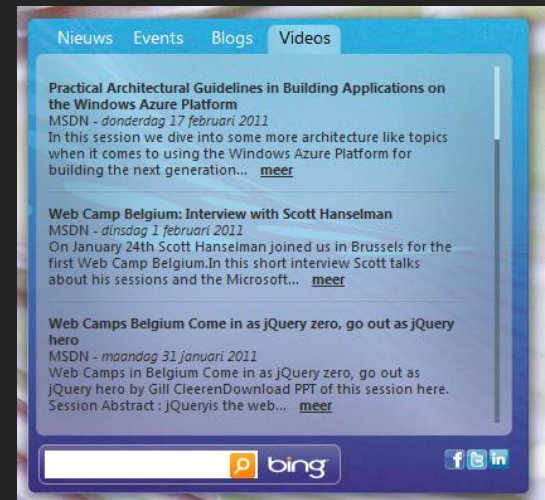
# Stay up to date with MSDN Belux

- Register for our newsletters and stay up to date:  
<http://www.msdn-newsletters.be>
  - Technical updates
  - Event announcements and registration
  - Top downloads
- Follow our blog  
<http://blogs.msdn.com/belux>
- Join us on Facebook  
<http://www.facebook.com/msdnbe>  
<http://www.facebook.com/msdnbelux>
- LinkedIn: <http://linkd.in/msdnbelux/>
- Twitter: [@msdnbelux](https://twitter.com/msdnbelux)

## Download

MSDN/TechNet Desktop Gadget

<http://bit.ly/msdntngadget>



# TechDays 2011 On-Demand

---

- **Watch** this session on-demand via Channel9  
<http://channel9.msdn.com/belux>
- Download to your favorite MP3 or video player
- Get access to slides and recommended resources by the speakers



# THANK YOU

---

