



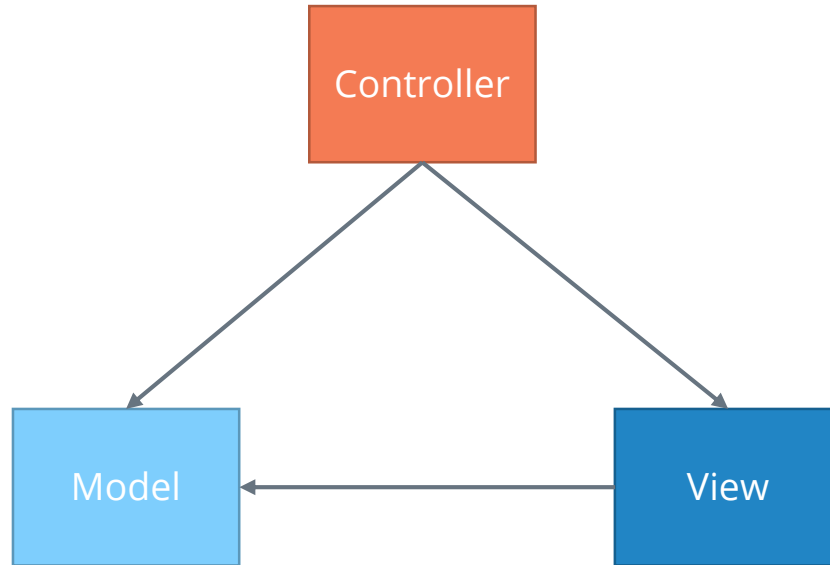
Headspring

Introduction to Controllers

| Agenda

- Controller fundamentals
- Actions
- Model binding
- Action results

| What is a controller?



| Controller Responsibilities

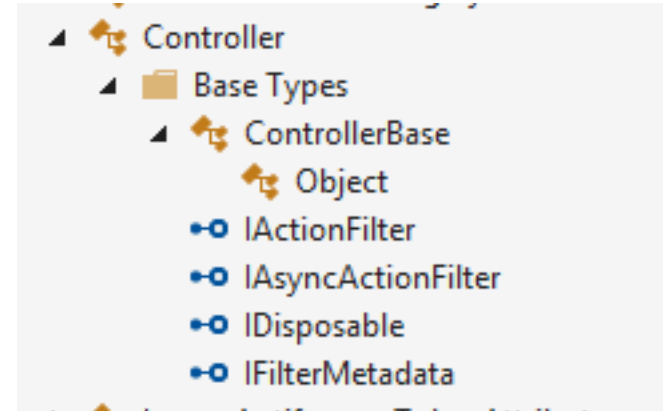
- Handle request
- Build model
- Return result

What a controller is not

```
//  
// POST: /Manage/ChangePassword  
[HttpPost]  
[ValidateAntiForgeryToken]  
public async Task<IActionResult> ChangePassword(ChangePasswordViewModel model)  
{  
    if (!ModelState.IsValid)  
    {  
        return View(model);  
    }  
    var user = await GetCurrentUserAsync();  
    if (user != null)  
    {  
        var result = await _userManager.ChangePasswordAsync(user, model.OldPassword, model.NewPassword);  
        if (result.Succeeded)  
        {  
            await _signInManager.SignInAsync(user, isPersistent: false);  
            _logger.LogInformation(3, "User changed their password successfully.");  
            return RedirectToAction(nameof(Index), new { Message = ManageMessageId.ChangePasswordSuccess });  
        }  
        AddErrors(result);  
        return View(model);  
    }  
    return RedirectToAction(nameof(Index), new { Message = ManageMessageId.Error });  
}
```

What is a controller?

- Inherits Controller
- Includes common filters
- Action result helpers
- Used for both MVC and Web API controllers



| What is an Action?

- The method must be public.
- The method cannot be a static method.
- The method cannot be an extension method.
- The method cannot be a constructor, getter, or setter.
- The method cannot have open generic types.
- The method is not a method of the Controller base class.
- The method cannot contain ref or out parameters

| Typical Action

```
//  
// GET: /Store/  
public async Task<IActionResult> Index()  
{  
    var genres = await DbContext.Genres.ToListAsync();  
  
    return View(genres);  
}
```


Complex Action

```
//  
// GET: /Store/Browse?genre=Disco  
public async Task<IActionResult> Browse(string genre)  
{  
    var genreModel = await DbContext.Genres  
        .Include(g => g.Albums)  
        .Where(g => g.Name == genre)  
        .FirstOrDefaultAsync();  
  
    if (genreModel == null)  
    {  
        return NotFound();  
    }  
  
    return View(genreModel);  
}
```

| Return Types

- IActionResult – synchronous
- Task<IActionResult> - asynchronous
- Asynchronous can return tasks or await

Model Binding

| What is Model Binding?

```
//  
// POST: /Account/Login  
[HttpPost]  
[AllowAnonymous]  
[ValidateAntiForgeryToken]  
public async Task<IActionResult> Login(  
    LoginViewModel model, string returnUrl = null)  
{
```

| What is Model Binding?

- Transforms request variables into action parameters
- “Where do I get the values from?”
- “How do I deserialize these values into an object?”

“Where do I get the values from?”

```
POST http://localhost:14859/Account/Login HTTP/1.1
Host: localhost:14859
Connection: keep-alive
Content-Length: 279
Cache-Control: max-age=0
Origin: http://localhost:14859
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.157 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://localhost:14859/Account/Login
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: ai_user=iexYl|2016-01-28T15:37:26.993Z; mp_f346a58175feb9c804311b56f206
```

```
Email=jimmy%40headspring.com&Password=correcthorsebatterystaple&RememberMe=true
```

| What sources are supported?

- Querystring
- Form data
- Routes

How do I deserialize these values into an object?

```
public class LoginViewModel
{
    [Required]
    [EmailAddress]
    public string Email { get; set; }

    [Required]
    [DataType(DataType.Password)]
    public string Password { get; set; }

    [Display(Name = "Remember me?")]
    public bool RememberMe { get; set; }
}
```

Don't get cute!

| Fine-grained control

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> AddressAndPayment(
    [FromServices] MusicStoreContext dbContext,
    [FromForm] Order order,
    CancellationToken requestAborted)
{
```

Action Results

| What is an Action Result?

- Represents the result of an action
- Includes instructions on what to do
- Executes how to do it

| The new IActionResult

```
public interface IActionResult
{
    Task ExecuteResultAsync(ActionContext context);
}
```

The ViewResult

```
public class ViewResult : ActionResult
{
    public int? StatusCode { get; set; }

    public string ViewName { get; set; }

    public object Model => ViewData?.Model;

    public ViewDataDictionary ViewData { get; set; }

    public TempDataDictionary TempData { get; set; }

    public IViewEngine ViewEngine { get; set; }

    public string ContentType { get; set; }

    public override async Task ExecuteResultAsync(ActionContext context)...
```

Built-in Action Results (over 30)

- MVC – views, redirects, etc.
- Web API – status code-based
- Files – downloading files
- Authentication – using new AuthenticationManager

| MVC Action Results

- ViewResult
- PartialViewResult
- JsonResult
- ViewComponentResult
- RedirectToActionResult
- RedirectToRouteResult
- RedirectResult
- LocalRedirectResult

MVC Action Example

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Login(LoginViewModel model, string returnUrl = null)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // This doesn't count login failures towards account lockout
    // To enable password failures to trigger account lockout, change to lockoutOnFailure: true
    var result = await SignInManager.PasswordSignInAsync(model.Email, model.Password, model.RememberMe, lo
    if (result.Succeeded)
    {
        _logger.LogInformation("Logged in {userName}.", model.Email);
        return RedirectToLocal(returnUrl);
    }
    if (result.RequiresTwoFactor)
    {
        return RedirectToAction("SendCode", new { ReturnUrl = returnUrl, RememberMe = model.RememberMe });
    }
    if (result.IsLockedOut)
    {
        return View("Lockout");
    }
}
```


C# 6 Support

```
public async Task<IActionResult> AddToCart(int id, CancellationToken requestAborted)
{
    // Retrieve the album from the database
    var addedAlbum = await DbContext.Albums
        .SingleAsync(album => album.AlbumId == id);

    // Add it to the shopping cart
    var cart = ShoppingCart.GetCart(DbContext, HttpContext);

    await cart.AddToCart(addedAlbum);

    await DbContext.SaveChangesAsync(requestAborted);
    _logger.LogInformation("Album {albumId} was added to the cart.", addedAlbum.AlbumId);

    // Go back to the main store page for more shopping
    return RedirectToAction(nameof(Index));
}
```

| Web API Action Results – 20x

- OkResult
 - OkObjectResult
 - CreatedResult
 - CreatedAtActionResult
 - CreatedAtRouteResult
 - NoContentResult
- } 200 OK
- } 201 Created
- } 204 No Content

| Web API Action Results – 4xx

- BadRequestObjectResult
 - BadRequestResult
 - UnauthorizedResult
 - NotFoundObjectResult
 - NotFoundResult
 - UnsupportedMediaTypeResult
- } 400 Bad Request
- } 401 Unauthorized
- } 404 Not Found
- } 415 Not Supported

| Custom Web API Action Results

- ObjectResult
 - Status code
 - Body
 - CreatedResult, OkObjectResult etc.
- StatusCodeResult
 - Status code
 - No body
 - NotFoundResult, BadRequestResult etc.

Custom Web API Action Result Example

```
public class AcceptedActionResult : StatusCodeResult
{
    public string Location { get; set; }

    public AcceptedActionResult(string location)
        : base(StatusCode.Status202Accepted)
    {
        Location = location;
    }

    public override void ExecuteResult(ActionContext context)
    {
        context.HttpContext.Response.Headers[HeaderNames.Location]
            = Location;

        base.ExecuteResult(context);
    }
}
```

| File and Content Results

- ContentResult
 - Raw string content
- FileContentResult
 - Byte array content
- FileStreamResult
 - Stream content
- PhysicalFileResult
 - Physical path
- VirtualFileResult
 - Virtual path