



---

---

---

---

---

---

---

---

Översikt

- Skriva Controllers och Actions.
- Skriva Action Filters.

.eerec

---

---

---

---

---

---

---

---

Lektion 1: Skriva Controllers och Actions

- Introduktion controller.
- Svara på en begäran.
- Begäran från användare.
- Enkel klass för controller.
- Definera händelser i controller.
- Programkod för controller.
- ActionResult.
- DefaultModelBinder.
- Skicka vidare information till View.
- Controller Factory.

.eerec

---

---

---

---

---

---

---

---

---

---

## Introduktion Controller

- Controller är en .NET Framework klass som ärver från basklassen System.Web.Mvc.Controller.
- Controller svara på något som användaren initierar.
- I controller definierar du händelse för att svara på användarens begäran.
- Actions är metod, inuti controller, som returnerar ett ActionResult objekt.
- ActionResult objektet är oftast en view soom visar svaret på begäran för användaren.

.easesec

---

---

---

---

---

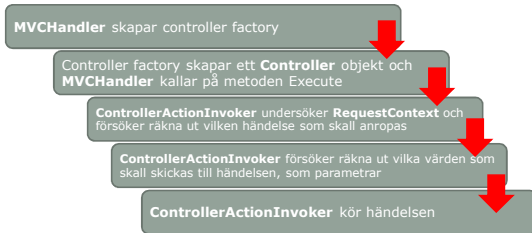
---

---

---

## Svara på en begäran

- När en webbapplikation baserat på MVC tar emot en begärn från en webbläsare händer följande:



.easesec

---

---

---

---

---

---

---

---

## Begärn från användare

- Användare anger antingen en URL i adressfältet, eller klickar på en länk på sidan.
- Följande kommer att hända:
  - <https://easesec.se>, URL för hemsidan.
  - <https://easesec.se/kurser>, URL specificerar ytterligare information: kurser. Som standard hanterar MVCHandler detta som namn för controller.
  - <https://easesec.se/kurser/index>, URL specificerar ett andra värde: index. Som standard hanterar MVCHandler detta som händelse i controller.
  - <https://easesec.se/kurser/display/1>, URL specificerar ett tredje värde: 1. Som standard hanterar ControllerActionInvoker detta som parameter till metod.
  - <https://easesec.se/kurser/display?id=1>, URL specificerar query string: id=1. ModelBinder undersöker händelse display, om den hittar händelse id, kommer parameter att skickas.

.easesec

---

---

---

---

---

---

---

---

## Enkel klass för controller

```
public class PhotoController : Controller
{
    private ContextDB context = new ContextDB();
    public ActionResult Index ()
    {
        return View("Index", context.Photos.ToList());
    }
}
```

.eeec

---

---

---

---

---

---

---

---

## Definera händelser i Controller

- Controllers kapslar in logik för användares interagering i en MVC-baserad applikation.
- Du specificerar logik, genom att skriva programkod för händelser.
- Action är en metod inuti controller klass. Programkod som du skriver, kommer att bestämma hur controller svarar på begäran, klass för model och vilken view som kommer att visa sidan i webbläsaren.

.eeec

---

---

---

---

---

---

---

---

## Programkod för controller

- Att skriva programkod för controller inkluderar:
  - Använda HTTP-verbena GET och POST i händelse.
  - Skapa klass för ActionResult.
  - Skapa eventuella Child Actions.
- Enkel controller action:

```
public ActionResult First ()
{
    Photo firstPhoto = context.Photos.ToList()[0];
    if (firstPhoto != null)
    {
        return View("Details", firstPhoto);
    }
    else
    {
        return HttpNotFound();
    }
}
```

.eeec

---

---

---

---

---

---

---

---

---

---

## ActionResult

- ActionResult används för att returnera något, exempelvis View alternativt ett HTTP fel, genom att använda HttpNotFound.
- Kan även returnera fil, som i sin tur kan användas som källa för bild på en sida.
- Child Actions, är en metod som returnera en delmängd.

.eeec

---

---

---

---

---

---

---

---

## DefaultModelBinder

- När användare efterfrågar webbsida, specificerar de oftast ytterligare information, exempelvis ett produktnamn.
- Extra information refereras till parametrar.
- DefaultModelBinder kan lokalisera parametrar, kalla på metod i action och skicka vidare parameter.
- Du kan skapa skräddarsydd ModelBinder, viktigt att du arbetar med rätt parameter.

.eeec

---

---

---

---

---

---

---

---

## Skicka vidare information till View

- View() helper, är en metod som kan användas för att skicka vidare controller action till view.
- ViewBag är ett dynamiskt objekt som är en del av basklass för controller:
  - Dynamisk, alla typer av information kan läggas här.
  - View kan hämta information som finns i ViewBag.

.eeec

---

---

---

---

---

---

---

---

---

---

## Controller Factory

- Controller Factory är en MVC-komponent som instansierar klassen controller som du har skapat.
- Finns inbyggd, men egna kan användas.
- Om egna skall användas, finns följande krav:
  - Scope för klass måste vara public.
  - Klass får inte vara märkt som abstrakt.
  - Klass får inte använda parameter av typen generic.
  - Klass måste ha ett namn som slutar på Controller.
  - Klass måste implementera IController.
  - Registreras i filen Global.asax.

.eeec

---

---

---

---

---

---

---

---

## Övning Skapa controller



.eeec

---

---

---

---

---

---

---

---

## Lektion 2: Skriva Action Filters

- Vad är Filters?
- Custom Action Filters.
- Använda Custom Action Filters.

.eeec

---

---

---

---

---

---

---

---

---

---

## Vad är filters?

- Filters är MVC-klasser som är relevanta för många delar i din webbapplikation.
- Det finns fyra typer av filter:
  - Authorization filters, som körs innan något annat filter och innan programkod i metod som är definierad i händelse.
  - Action filters körs innan och efter programkod för metod i händelse.
  - Result filters körs innan eller efter det att resultat returneras från metod i händelse.
  - Exception filters körs bara om metod i händelse eller ett annat filter kastar ett undantag.

.eeec

---

---

---

---

---

---

---

---

## Vad är filters? (forts.)

Filtertyp	Interface	Standardklass	Beskrivning
Authorization	IAuthorizationFilter	AuthorizeAttribute	Körs innan något annat filter och innan programkod i metod som är definierad i händelse.
Action	IActionFilter	ActionFilterAttribute	Körs innan och efter programkod för metod i händelse.
Result	IResultFilter	ActionFilterAttribute	Körs innan eller efter det att resultat returneras från metod i händelse.
Exceptions	IExceptionFilter	HandlerErrorAttribute	Körs bara om metod i händelse eller ett annat filter kastar ett undantag.

.eeec

---

---

---

---

---

---

---

---

## Custom Action Filters

- Skräddarsydda filter kan skapas för både action och result filters.
- Du implementerar interface IActionFilter eller IResultFilter.
- Du kan skapa ett filter som kör programkod både innan och efter händelse körs och innan eller efter ett resultat returneras.

.eeec

---

---

---

---

---

---

---

---

---

---

## Custom Action Filters -exempel

```
public class SimpleActionFilter : ActionFilterAttribute
{
    public override void OnActionExecuting
    (ActionExecutingContext filterContext)
    {
        Debug.WriteLine("This Event Fired: OnActionExecuting");
    }
    public override void OnActionExecuted
    (ActionExecutedContext filterContext)
    {
        Debug.WriteLine("This Event Fired: OnActionExecuted");
    }
}
```

.cs/ce

---

---

---

---

---

---

---

---

## Använda Custom Action Filters

- När du har skapat ditt skräddarsydda filter, kan detta appliceras på vilken metod som helst eller klass i din webbapplikation.
- Exempel:

```
public class PhotoController : Controller
{
    ContextDB contextDB = new ContextDB();
    [SimpleActionFilter]
    public ActionResult Index()
    {
        return View("Index", contextDB.Photos.ToList());
    }
}
```

.cs/ce

---

---

---

---

---

---

---

---

## Repetitionsfrågor

.cs/ce

---

---

---

---

---

---

---

---

---

---