Unterschiede zwischen WebForms und ASP.NET MVC

# Gibt es nur in WebForms

* ViewState und pseudo-Statefullnes
* Control Lifecycle mit Init, Load, PreRender…
* Wiederverwendare Controls mit automatischem ID-Scoping etc.
* Modularisierung und „friedliches Nebeneinander“ von Controls auf einer Page

# Sowohl WebForms als auch MVC

* Web.config
* Global.asax
* HttpModules
* Session State
* Membership, Role Providers
* ASCX/ASPX Syntax und Features
* MasterPages
* HttpHandler inkl. AsyncHttpHandler für spezielle Situationen
* Alle .net Features

# Nur ASP.NET MVC

* Url unabhängig von Code-Struktur (wichtig für Search Engine Optimization)
* Urls regelgesteuert aufbauen
* Lesbare, sprechende Urls
* Cshtml Razor Syntax für Views
* Klare Trennung zwischen Ablauf (Controller) und Darstellung (View)
* Saubere Einstiegspunkte für modulare Entwicklung mit Dependency Injection
* Testbarkeit der Controller
* Volle Kontrolle über das HTML
* Fast jeder Aspekt erweiterbar oder ersetzbar (Controller-Lookup, View-Engines etc.)
* Möglichkeit, direkt JSON zurückzugeben

Life Cycle in WebForms und in ASP.NET MVC

# WebForms

|  |  |
| --- | --- |
| **Stage** | **Description** |
| Page request | The page request occurs before the page life cycle begins. When the page is requested by a user, ASP.NET determines whether the page needs to be parsed and compiled (therefore beginning the life of a page), or whether a cached version of the page can be sent in response without running the page. |
| Start | In the start stage, page properties such as [Request](http://msdn.microsoft.com/en-us/library/system.web.ui.page.request.aspx) and [Response](http://msdn.microsoft.com/en-us/library/system.web.ui.page.response.aspx) are set. At this stage, the page also determines whether the request is a postback or a new request and sets the [IsPostBack](http://msdn.microsoft.com/en-us/library/system.web.ui.page.ispostback.aspx) property. The page also sets the [UICulture](http://msdn.microsoft.com/en-us/library/system.web.ui.page.uiculture.aspx) property. |
| Initialization | During page initialization, controls on the page are available and each control's [UniqueID](http://msdn.microsoft.com/en-us/library/system.web.ui.control.uniqueid.aspx) property is set. A master page and themes are also applied to the page if applicable. If the current request is a postback, the postback data has not yet been loaded and control property values have not been restored to the values from view state. |
| Load | During load, if the current request is a postback, control properties are loaded with information recovered from view state and control state. |
| Postback event handling | If the request is a postback, control event handlers are called. After that, the [Validate](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.basevalidator.validate.aspx) method of all validator controls is called, which sets the [IsValid](http://msdn.microsoft.com/en-us/library/system.web.ui.ivalidator.isvalid.aspx) property of individual validator controls and of the page. |
| Rendering | Before rendering, view state is saved for the page and all controls. During the rendering stage, the page calls the [Render](http://msdn.microsoft.com/en-us/library/system.web.ui.control.render.aspx) method for each control, providing a text writer that writes its output to the [OutputStream](http://msdn.microsoft.com/en-us/library/system.web.httpresponse.outputstream.aspx) object of the page's [Response](http://msdn.microsoft.com/en-us/library/system.web.ui.page.response.aspx) property. |
| Unload | The [Unload](http://msdn.microsoft.com/en-us/library/system.web.ui.control.unload.aspx) event is raised after the page has been fully rendered, sent to the client, and is ready to be discarded. At this point, page properties such as [Response](http://msdn.microsoft.com/en-us/library/system.web.ui.page.response.aspx) and [Request](http://msdn.microsoft.com/en-us/library/system.web.ui.page.request.aspx) are unloaded and cleanup is performed. |

# ASP.NET MVC

|  |  |
| --- | --- |
| Stage | Details |
| Receive first request for the application | In the Global.asax file, Route objects are added to the [RouteTable](http://msdn.microsoft.com/en-us/library/system.web.routing.routetable(v=VS.98).aspx) object. |
| Perform routing | The [UrlRoutingModule](http://msdn.microsoft.com/en-us/library/system.web.routing.urlroutingmodule(v=VS.98).aspx) module uses the first matching Route object in the [RouteTable](http://msdn.microsoft.com/en-us/library/system.web.routing.routetable(v=VS.98).aspx) collection to create the [RouteData](http://msdn.microsoft.com/en-us/library/system.web.routing.routedata(v=VS.98).aspx) object, which it then uses to create a [RequestContext](http://msdn.microsoft.com/en-us/library/system.web.routing.requestcontext(v=VS.98).aspx) object. |
| Create MVC request handler | The [MvcRouteHandler](http://msdn.microsoft.com/en-us/library/system.web.mvc.mvcroutehandler(v=VS.98).aspx) object creates an instance of the [MvcHandler](http://msdn.microsoft.com/en-us/library/system.web.mvc.mvchandler(v=VS.98).aspx) class and passes the [RequestContext](http://msdn.microsoft.com/en-us/library/system.web.routing.requestcontext(v=VS.98).aspx) instance to the handler. |
| Create controller | The [MvcHandler](http://msdn.microsoft.com/en-us/library/system.web.mvc.mvchandler(v=VS.98).aspx) object uses the [RequestContext](http://msdn.microsoft.com/en-us/library/system.web.routing.requestcontext(v=VS.98).aspx) instance to identify the [IControllerFactory](http://msdn.microsoft.com/en-us/library/system.web.mvc.icontrollerfactory(v=VS.98).aspx) object (typically an instance of the [DefaultControllerFactory](http://msdn.microsoft.com/en-us/library/system.web.mvc.defaultcontrollerfactory(v=VS.98).aspx) class) to create the controller instance with. |
| Execute controller | The [MvcHandler](http://msdn.microsoft.com/en-us/library/system.web.mvc.mvchandler(v=VS.98).aspx) instance calls the controller's [Execute](http://msdn.microsoft.com/en-us/library/system.web.mvc.controllerbase.execute(v=VS.98).aspx) method. |
| Invoke action | For controllers that inherit from the [ControllerBase](http://msdn.microsoft.com/en-us/library/system.web.mvc.controllerbase(v=VS.98).aspx) class, the [ControllerActionInvoker](http://msdn.microsoft.com/en-us/library/system.web.mvc.controlleractioninvoker(v=VS.98).aspx) object that is associated with the controller determines which action method of the controller class to call, and then calls that method. |
| Execute result | The action method receives user input, prepares the appropriate response data, and then executes the result by returning a result type. The built-in result types that can be executed include the following: [ViewResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.viewresult(v=VS.98).aspx) (which renders a view and is the most-often used result type), [RedirectToRouteResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.redirecttorouteresult(v=VS.98).aspx), [RedirectResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.redirectresult(v=VS.98).aspx), [ContentResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.contentresult(v=VS.98).aspx), [JsonResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.jsonresult(v=VS.98).aspx), [FileResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.fileresult(v=VS.98).aspx), and [EmptyResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.emptyresult(v=VS.98).aspx). |

Anderes Thema