

ASP.NET MVC Best Practices

Tomáš Jecha (MVP)

✉ tomas@jecha.net

t jechtom

e www.jecha.net



Školení vývojářů
www.dotNETcollege.cz

Články a diskuzní fóra
www.dotNETportal.cz

Témata

- Logování chyb
- Web Essentials
- Zbavení se tlustých controllerů
- Grid v MVC
- Dependency Injection
- Ukázka z existujícího projektu

LOGOVÁNÍ VÝJIMEK

Jakým způsobem logovat?

- Obecné logování:
 - Trace (Trace, TraceSource)
 - Log4net, Nlog, Enterprise Library
- Logování pro webové aplikace:
 - ELMAH (Error Logging Modules and Handlers)
 - Řada konektorů, rozhraní pro procházení
- Vlastní logování



ELMAH

Logování za pár minut

Výjimky ve webových aplikacích

- Chyby během zpracování požadavku
 - Před aktivací controlleru (neexistující controller, chyba routování)
 - V kontextu controlleru
- Chyby bez kontextu požadavku
- Chyby terminující aplikaci



ELMAH

AppDomain unhandled errors

Kdy se Exception nezaloguje?

- ASP.NET MVC – [ExceptionHandler]
 - Pokud se použije, ošetří vyjímku a zobrazí Error.cshtml
 - Lze upravit: `context.ExceptionHandled=false`
- ASP.NET WebAPI – odchyťává vyjímky
 - Lze vytvořit globální WebAPI Error Filter



HandleError + ELMAH

Nejčastější chyby při logování

- Žádné logování
- Spoléháme se na `HttpContext.Current`
- Výjimka při sestavování detailů
- Zobrazení výjimek uživateli

WEB ESSENTIALS

Web Essentials

- Plugin zdarma, pomáhá s webovým vývojem
- Compilers: LESS, CoffeeScript, SweetJs, LiveScript...
- BrowserLink
- Bundling (+inline)
- HTML vylepšení, JS vylepšení
- ...



demo

LESS CSS

Browser Link

ARCHITEKTURA MVC

Co je to MVC?

- Architektura prezentační vrstvy
- ASP.NET MVC:
 - View – šablona generující z Modelu uživatelské prostředí (HTML)
 - Model – doménový model
 - Controller – obsluha příchozích požadavků a jejich mapování na Model a View

Controller

- Handler příchozích požadavků
- Kontext požadavku
- Mapuje svět HTTP na Model
- Neměl by řešit business logiku nebo DB
- Opakující se problémy je dobré automatizovat
(např. parsování komplexních typů)

View specific model

- Doménový model nemusí být přesně data UI
- ASP.NET MVC funguje nejlépe s model 1:1 UI
- Doporučuji vytvořit model pro každý formulář
- Logika mapování na doménový model uvnitř nebo v pomocných službách

demo

„Thustý controller“
a View Specific Model

Scénář: Registrace uživatelů

Komponenty v MVC?

- Vývojáři WebForms hledají náhradu autonomních komponent
- V ASP.NET MVC lze:
 - Znovupoužít modely, používat předky / rozhraní
 - Znovupoužít šablony pro zobrazení
 - Automatizovat opakující kód v controllerech

demo

Jednoduchý grid v ASP.NET MVC



Placené komponenty

DEPENDENCY INJECTION

Dependency Injection

- Rozvolnění závislostí mezi komponentami
 - Vytváření, konfiguraci a vkládání závislostí řídí kontejner; komponenty jen definují závislosti
- Nejčastější techniky:
 - Constructor injection
 - Property Injection
- Realizuje DI Container:
 - Castle Windsor, Unity, StructureMap, Ninject...

DI v ASP.NET MVC

- Nejčastěji se zasahuje:
 - ASP.NET MVC – IControllerFactory
 - ASP.NET WebAPI – IHttpControllerActivator
- Řízení životního cyklu „per web request“
 - Například kontext připojení do databáze



Castle Windsor v MVC

WEB MS-FEST.CZ



Layout modely

Lokalizace

Modely

Security

Tomáš Jecha (MVP)

✉ tomas@jecha.net

t [jechtom](https://twitter.com/jechtom)

e www.jecha.net



Školení vývojářů
www.dotNETcollege.cz

Články a diskuzní fóra
www.dotNETportal.cz