

ASP.NET Web API

Tomáš Herceg


Microsoft MVP (ASP.NET)

www.dotnetcollege.cz

Yet Another WCF?

- .NET Web Services (SOAP, WSDL)
 - Fungují vždy a všude, nedá se nic zkazit
- Windows Communication Foundation
 - Pokročilé možnosti konfigurace (až moc možností)
 - Lze přinutit komunikovat i pomocí JSONu nebo XML
 - webHttpBinding
 - Příliš obecné, málokdo umí, configuration hell !#GRR!
- ASP.NET Web API
 - Jednoduchá a jednoúčelová technologie
 - OWIN

Co to umí?

- Technologie pro REST služby
 - URL routing
 - Podpora JSON, OData a dalších formátů
 - Generování dokumentace (Help Pages)
 - Rozšiřitelnost
 - Pokročilé možnosti hostování
- 

Základy

- Controller – třída dědící z ApiController
- Action – metoda uvnitř controlleru
 - HTTP metoda
 - Pomocí názvu – GetProducts (GET), DeleteProduct (DELETE)
 - Nebo atributem – [HttpGet], [HttpPost]
 - Parametry metody
 - Primitivní dat. typy – int, bool, string, Guid atd.
 - Defaultně z URL
 - Komplexní typy – z těla požadavku
 - Možno řídit atributy FromUri a FromBody
 - Lze dát vždy jen jednomu z parametrů
 - Návrátová hodnota
 - Void => 204 No Content
 - Objekt => 200
 - Redirect

Konzumace Web API

- Buď klasický WebClient
- Nebo třída HttpClient
 - NuGet balíček ASP.NET Web Api Client Libs
 - `client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));`
 - `GetAsync`, `PostAsync`
 - `response.Content.ReadAsAsync`

Užitečné tipy

- Vyhazování výjimek

- `throw new HttpResponseException(HttpStatusCode.NotFound);`
- Využívat HTTP kódy
 - 400 Bad Request
 - 402 Payment Required
 - 403 Forbidden
 - 404 Not Found
 - 409 Conflict
 - 418 I'm a teapot ☺

- Redirect

- Např. po přidání záznamu redirectnout na jeho detail
- `var uri = Url.Link("DefaultApi", new { id = item.Id });`
`response.Headers.Location = new Uri(uri);`
`return response;`

Routing

- Výchozí konfigurace
 - `routes.MapHttpRoute(`
 `name: "API Default",`
 `routeTemplate: "api/{controller}/{id}",`
 `defaults: new { id = RouteParameter.Optional });`
 - `ProductsController => api/Products`
- Parametr `{action}` – název metody v Controlleru
 - Možnost customizace pomocí `[ActionName("...")]`
 - Atribut `[NonAction]`
 - Atributy `[HttpGet]`, `[HttpPost]`, ..., `[AcceptVerbs]`

Různé routy pro různé controllery

```
routes.MapHttpRequestRoute(  
    name: "ApiRoot",  
    routeTemplate: "api/root{id}",  
    defaults: new { controller = "products",  
                  id = RouteParameter.Optional }  
);  
routes.MapHttpRequestRoute(  
    name: "DefaultApi",  
    routeTemplate: "api/{controller}/{id}",  
    defaults: new { id = RouteParameter.Optional }  
);
```


ASP.NET Web API 2.0

- Attribute Routes
 - Routy přímo nad action metodami
 - Není nutná složitá globální konfigurace
 - Ale lze kombinovat
 - `[Route("customers/{customerId}/orders")]`
 - Volitelné parametry `RouteName` a `RouteOrder`
- Při startu volat
 - `httpConfiguration.MapHttpAttributeRoutes()`
- Atribut `[RoutePrefix("...")]` nad controllerem

Route Constraints

- [Route("users/{id:int:min(1)}")]
 - **alpha**
 - **bool**
 - **datetime**
 - **decimal**
 - **double**
 - **float**
 - **guid**
 - **int**
 - **length(5, 20)**
 - **long**
 - **max(10)**
 - **maxlength(10)**
 - **min(1)**
 - **minlength(10)**
 - **range**
 - **regex**
- Vlastní constrainty – **IHttpRequestConstraint**
- Volitelné parametry – {id?}

Help Pages

- NuGet balík `Microsoft.AspNet.WebApi.HelpPage`
- V `Global.asax` zavolat
 - `AreaRegistration.RegisterAllAreas();`
 - `config.SetDocumentationProvider(...);`
- Vlastnosti projektu / Build / XML Doc File
- Na adrese `/Help` pak visí dokumentace
- Možnost vyloučení akcí nebo controllerů
 - `[ApiExplorerSettings(IgnoreApi=true)]`

Exception Handling

- Exception filtry
 - Třída dědící z `ExceptionHandlerAttribute`
- Možno aplikovat na controller nebo na akci
 - Jako atribut
- Nebo globálně
 - `GlobalConfiguration.Configuration.Filters.Add(...)`
- Podpora Model Validation
 - Jako v MVC – `ModelState`
 - Vlastní chyby – `Request.CreateErrorResponse()`


Model State

- Dostupný v každé Web API metodě
- Standardní validační atributy
 - System.ComponentModel.DataAnnotations
 - Required, Range, RegularExpression
- 2 možnosti
 - V každé funkci kontrolovat if (!ModelState.IsValid)...
 - Vlastní ActionFilter
 - Nebude provádět danou funkci a vrátí HTTP kód 400
 - Vrátí validační chyby jako JSON
 - Použití stejné jako u exception filtrů

Response Handlers

- Možnost manipulace se zpracovávaným požadavkem
 - Třída DelegatingHandler
- Registrace pomocí
 - `config.MessageHandlers.Add(...)`
- Využití
 - Kontrola API key
 - Přidávání HTTP hlaviček pro všechny requesty
 - Logging, tracing atd.
 - <http://www.asp.net/web-api/overview/working-with-http/http-message-handlers>

Vlastní zpracování požadavku

- Vhodné např. pro upload souborů
 - `async Task<HttpResponseMessage>`
 - Třída `MultipartFormDataStreamProvider`
 - `Request.Content.ReadAsMultipartAsync`
 - `Request.Content.IsMultipartContent`
- 

Autorizace

- Atribut Authorize
 - Volitelné parametry Users a Roles
- Autentizace spolupracuje s ASP.NET
 - Forms, Windows, Individual Accounts

OData

- OData
 - Hodí se pro relační data
 - Snadné filtrování, řazení, relace (dotaz se specifikuje v query stringu)
 - Např. [http://odatatest.com/Customers?\\$filter=Id+eq+1](http://odatatest.com/Customers?$filter=Id+eq+1)
- Poměrně široká podpora
 - Samplý ukazují integraci s Entity Frameworkem
 - Ale lze i bez něj
 - Funkce, která vrací IQueryable
 - Atribut [Queryable(AllowedQueryOptions=AllowedQueryOptions.All)]
- Bude samostatná přednáška 😊

OWIN

- System.Web
 - Obrovské závislosti na COMu
 - Pomalý startup aplikace, velká paměťová náročnost
 - Celé WebForms jsou v paměti, i když používáte MVC
 - Více než 10 let stará
- OWIN – Open Web Interface for .NET
 - Definice toho, jak má vypadat webový server
 - Jednoduchost
 - Modularita a rozšiřitelnost

OWIN a Katana

- Jak provozovat OWIN aplikaci
 - Self hosting
 - Owin Host for System.Web – možné na IISce
- Katana
 - Implementace frameworků nad OWINem
 - Zatím SignalR a ASP.NET Web API
 - Připravují i MVC
 - WebForms nebudou ☹

Self hosting Web API

- Nainstalovat balíček
Microsoft.AspNet.WebApi.OwinSelfHost
- V OWIN Startup class nastavit
 - `var config = new HttpConfiguration();`
 - `config.Routes.MapHttpRoute...`
 - `appBuilder.UseWebApi(config)`
- `using (WebApp.Start<Startup>(url))`
 - Nastartuje OWIN host s danou Startup třídou