

Paweł Rajba

pawel@ii.uni.wroc.pl

<http://www.itcourses.eu/>

Web API

Agenda

- Wprowadzenie do Web API
- Kiedy WCF, a kiedy Web API
- Hostowanie aplikacji
- Adresowanie
- API controller
- Content negotiation
- Model binding & formatter
- HttpClient

Wprowadzenie do Web API

- Nowy sposób na usługi
- Bliski HTTP, łatwo utworzyć RESTful service
- Nowy typ aplikacji dostępny równoległe do ASP.NET MVC 4
 - ale niezależny od MVC4
- Dostępny nowy klient HTTP
- Dobrze współpracuje z nowymi klientami
 - np. HTML5, mobile
- Dostępny przez NuGET

Wprowadzenie do Web API

- Service to ApiController
- Dostępne są mechanizmy z MVC
 - Routing
 - Model binding
 - Action filters
 - Mapowanie parametrów z URL-a na parametry metody
- Konfiguracja
 - dostępna przez klasę `HttpConfiguration`
 - domyślnie w klasie `App_Start\WebApiConfig`

Kiedy WCF a kiedy Web API

- WebApi to nowe podejście, ale WCF nadal będzie lepszy gdy:
- Mamy .NET w wersji < 4
- Potrzebujemy
 - architektury SOAP
 - innych typów bindingów, np. TCP

Hostowanie aplikacji

- Jako aplikacja ASP.NET w IIS
- Jako self-hosted
 - w console application
 - w Windows Service
- Aplikację piszemy tak samo

Adresowanie

- ASP.NET hosting
 - mapowanie adresu i komend (GET, POST,...) na kontroler
 - konwencja jest taka, że szukana jest metoda GET, POST - można to zmienić przez atrybuty
 - extension method: MapHttpRequest
- Self hosting
 - dowolne mapowanie

API Controller

- Oparty o konwencje
 - metody Get, Post, Put, Delete
- Oparty o atrybuty
 - HttpGet, HttpPost, ...
 - wtedy nazwy metod dowolne

Content Negotiation

- Klient poprzez nagłówek Accept może określić oczekiwany typ wyniku
 - Automatyczne generowanie odpowiedniego typu odpowiedzi
 - Domyślnie działa dla XML, JSON
- Można skonfigurować dla innych typów lub dla rozszerzeń plików (np. .xml czy .json)
 - Poprzez odpowiedni MediaTypeFormatter

Model binding & formatters

- Automatyczne mapowanie komunikatów na typy CLR
 - Można dodatkowo używać atrybutów:
 - przy parametrach:FromBody, FromUri
 - przy akcji: ModelBinder
 - Wsparcie ze strony MediaTypeFormatters
 - Domyślne dla XML, JSON
- Dostęp niskopoziomowy
 - HttpRequestMessage, HttpResponseMessage

HttpClient

- Co było wcześniej?
 - 1.0: HttpWebRequest
 - 2.0: WebClient
- Co daje nowy HttpClient?
 - Oparty o model programistyczny WebApi
 - m.in. HttpRequestMessage/HttpResponseMessage
 - Równoczesny dostęp do wielu URI
 - również z wielu domen
 - Wsparcie dla wywołań asynchronicznych
 - Dostępny przez NuGet

DEMO

- BasicSample
- BasicSampleWithMessages
- BasicSampleWithObjects
- FacebookClient