10 Ways to Build Web Services in .NET Nancy

Chad McCallum
@ChadEmm

Module Outline

- Introduction to Nancy
- Installing Nancy from NuGet
- Creating our first NancyModule and endpoint
- An endpoint with a route parameter
- An endpoint with a request body
- Specifying a root route
- Returning a HttpStatusCode from an endpoint
- Connecting to a web service using RestSharp
- Sending data using RestSharp
- Checking the response status code
- Review

Nancy

- A lightweight, low-ceremony framework for building HTTP services based on .NET
- Goal is to stay out of the way as much as possible and provide a super-duper-happy-path to all interactions

- Sensible defaults and conventions
- Runs in multiple hosting environments (ASP.NET, OWIN, WCF, Self-Hosting, etc)

Nancy

- **Super-Duper-Happy-Path**
  - “It Just Works“ – focuses on convention to automatically register modules, view engines, dependencies, without configuration
  - “Easily Customizable” – configuration hooks to allow users to easily configure or exchange components
  - “Low Ceremony” – minimal amount of code required to use Nancy
  - “Low Friction” – framework APIs are obvious, required configuration is minimal, but still powerful and extensible

Review

- Introduction to Nancy
- Installing Nancy from NuGet
- Creating our first NancyModule and endpoint
- An endpoint with a route parameter
- An endpoint with a request body
- Specifying a root route
- Returning a HttpStatusCode from an endpoint
- Connecting to a web service using RestSharp
- Sending data using RestSharp
- Checking the response status code

Nancy - NancyModule

- **Base class for code that handles requests**
- **Define routes in constructor**
  - Get["/route/{var}"] = arguments => Method(arguments.var);
  - Post, Put, Delete
  - Arguments is a dynamic object representing the request
- **Can define a "root" route in default constructor**
  - public PunService : base("/root")

Nancy – (De)Serialization

- **Return an object, Nancy will take care of content negotiation**
  - Client can request specific content format using extension (.json) or Accept header
- **NancyModule.Bind<T>() automatically deserializes request body to specified type**
  - Extension method in Nancy.ModelBinding namespace

RestSharp

- **A simple REST and HTTP API Client for .NET**
- **RestClient defines the domain and executes the requests**
  - new RestClient("http://localhost")
- **RestRequest defines the route and parameters of the request**
  - AddUrlSegment injects variables into a route template
  - AddObject adds a .NET object to serialize in the request
- **RestResponse wraps the result of the request and any deserialized data**
  - client.Execute<T>(request) will return a RestResponse with a .Data property of the type specified