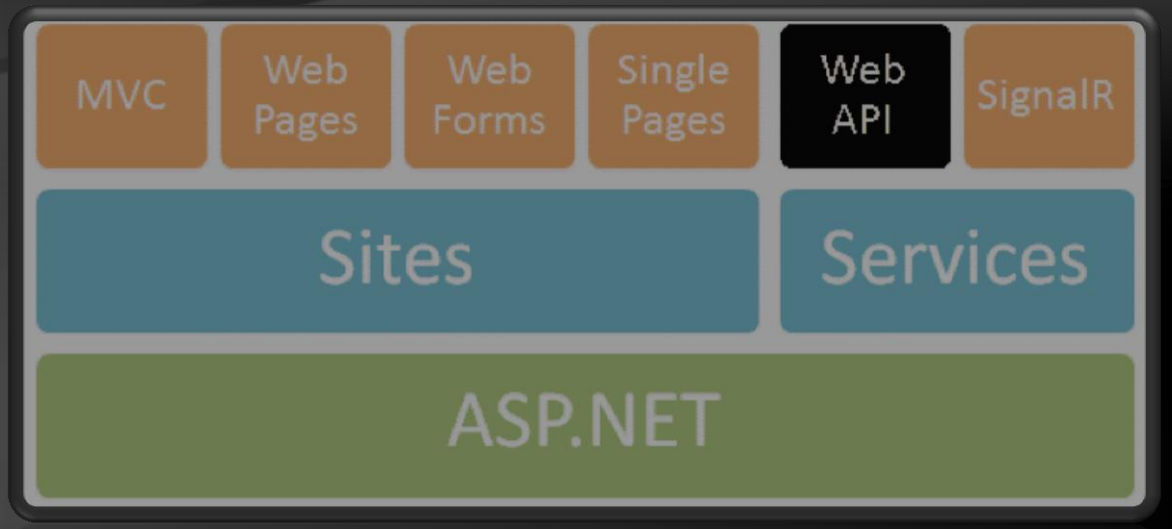
Telerik Academy



ASP.NET Web API

Web Services and Cloud

Telerik Software Academy

http://academy.telerik.com

ASP.NET

.Net Ninja

Telerik Academy

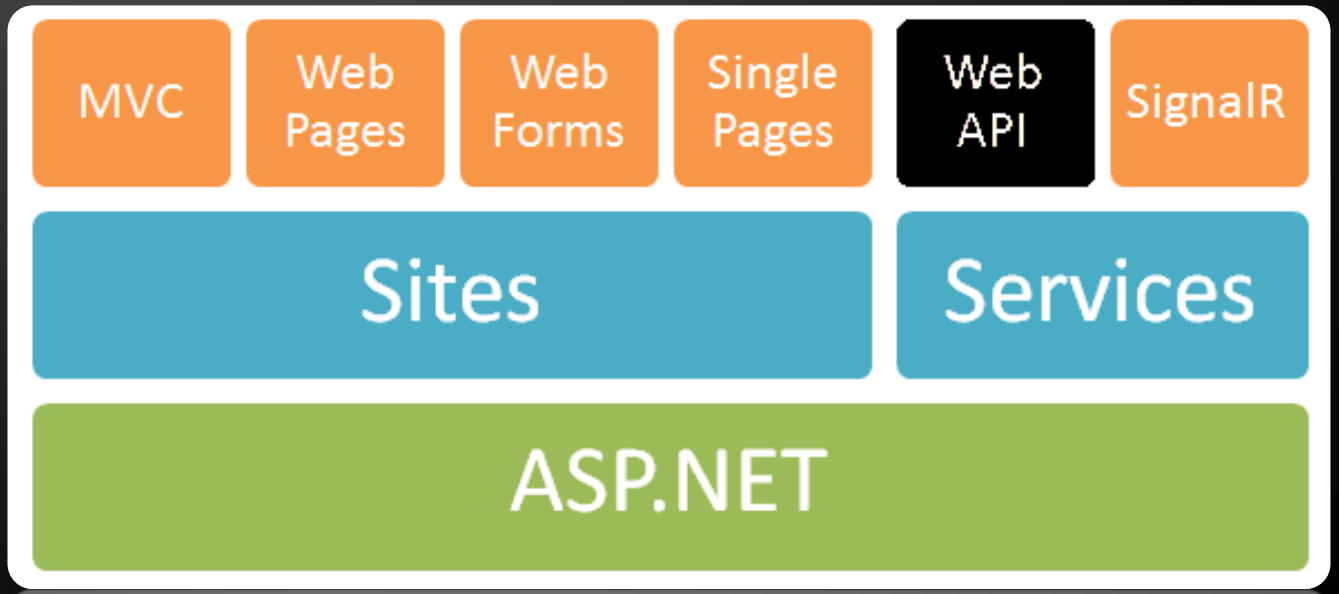
Table of Contents

Telerik Academy

What is ASP.NET Web API?

ASP.NET Web API

- Framework that makes it easy to build HTTP services for browsers and mobile devices
- Platform for building RESTful applications on the .NET Framework using ASP.NET stack



Telerik Academy

ASP.NET Web API Role

Data storage

Data Layer (EF)

Models

ASP.NET Web API

XML

XML

HTTP GET

JSON

JSON

HTTP PUT, POST, DELETE

Web API Features

- Modern HTTP programming model
  - Access to strongly typed HTTP object model
  - HttpClient API – same programming model
- Content negotiation
  - Client and server work together to determine the right format for data
  - Provide default support for JSON, XML and Form URL-encoded formats
  - We can add own formats and change content negotiation strategy

Web API Features (2)

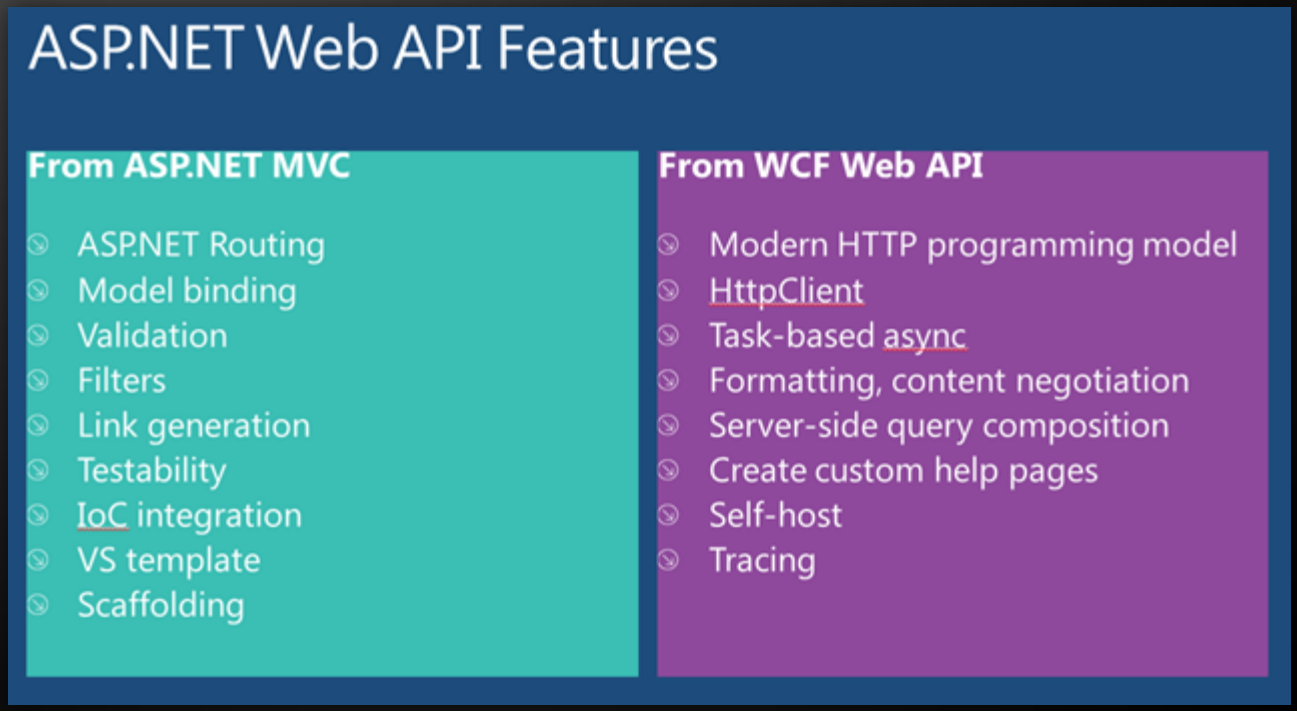
- Query composition
  - Support automatic paging and sorting
  - Support querying via the OData URL conventions when we return IQueryable<T>
- Model binding and validation
  - Combine HTTP data in POCO models
  - Data validation via attributes
  - Supports the same model binding and validation infrastructure as ASP.NET MVC

Web API Features (3)

- Routes (mapping between URIs and code)
  - Full set of routing capabilities supported within ASP.NET (MVC)
- Filters
  - Easily decorates Web API with additional validation (authorization, CORS, etc.)
- Testability
- IoC and dependency injection support
- Flexible hosting (IIS, Azure, self-hosting)

Web API Features (4)

- Visual Studio IDE (+templates and scaffolding)
- Reuse of C# knowledge (+task-based async)
- Custom help pages, tracing, etc.



ASP.NET Web API 2

- Attribute routing
- OData improvements: $select, $expand, $batch, $value and improved extensibility
- Request batching
- Portable ASP.NET Web API Client
- Improved testability
- CORS (Cross-origin resource sharing)
- Authentication filters
- OWIN support and integration (owin.org)

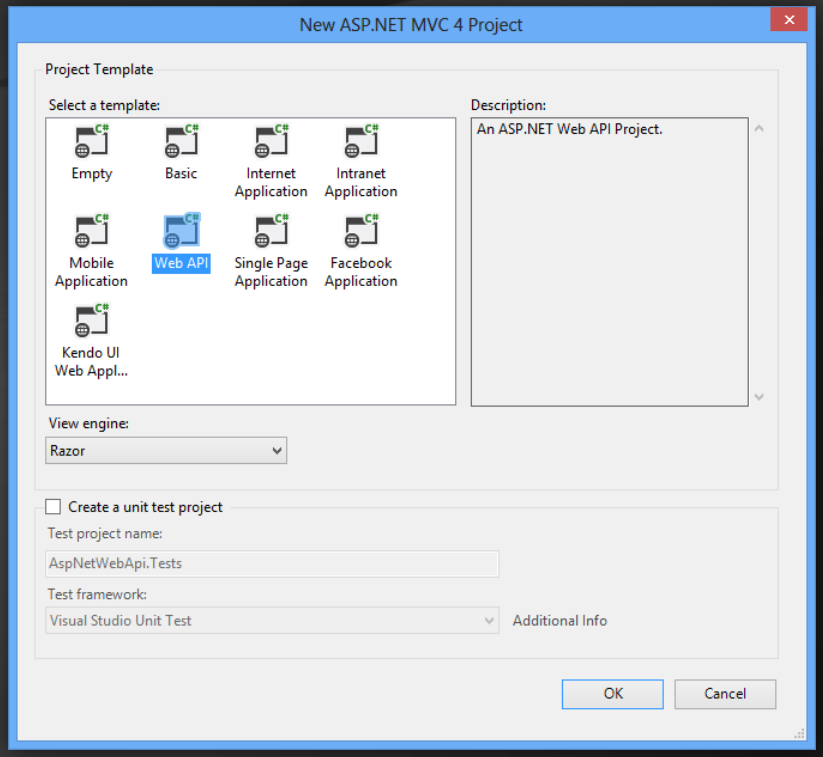
Telerik Academy

WCF vs. ASP.NET Web API

- WCF is also a good framework for building HTTP based services

WCF	ASP.NET Web API
Enables building services that support multiple transport protocols (HTTP, TCP, UDP, and custom transports) and allows switching between them.	HTTP only. First-class programming model for HTTP. More suitable for access from various browsers, mobile devices etc enabling wide reach.
Enables building services that support multiple encodings (Text, MTOM, and Binary) of the same message type and allows switching between them.	Enables building Web APIs that support wide variety of media types including XML, JSON etc.
Supports building services with WS-* standards like Reliable Messaging, Transactions, Message Security.	Uses basic protocol and formats such as HTTP, WebSockets, SSL, JQuery, JSON, and XML. There is no support for higher level protocols such as Reliable Messaging or Transactions.
Supports Request-Reply, One Way, and Duplex message exchange patterns.	HTTP is request/response but additional patterns can be supported through SignalRand WebSockets integration.
WCF SOAP services can be described in WSDL allowing automated tools to generate client proxies even for services with complex schemas.	There is a variety of ways to describe a Web API ranging from auto-generated HTML help page describing snippets to structured metadata for OData integrated APIs.
Ships with the .NET framework.	Ships with .NET framework but is open-source and is also available out-of-band as independent download.

Telerik Academy



Demo: Creating ASP.NET Web API Project

Default ASP.NET Web API project template

Telerik Academy

Web API Controllers

Web API Controllers

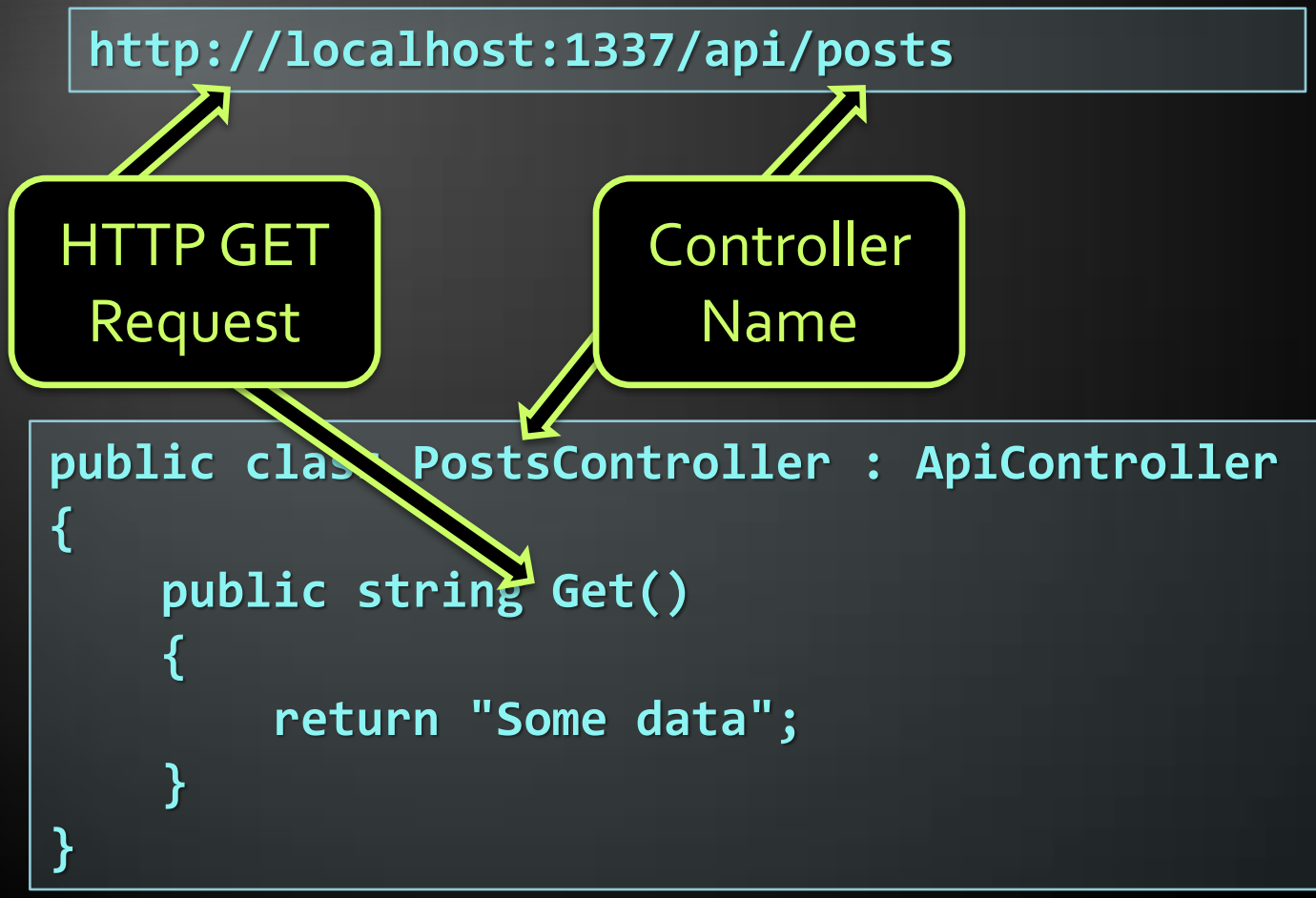
- A *controller* is an object that handles HTTP requests
  - All API controllers derive from ApiController
- By default ASP.NET Web API will map HTTP requests to specific methods called actions

Action	HTTP method	Relative URI	Method
Get a list of all posts	GET	/api/posts	Get()
Get a post by ID	GET	/api/posts/*id*	Get(int id)
Create a new post	POST	/api/posts	Post(PostModel value)
Update a post	PUT	/api/posts/*id*	Put(int id, PostModel value)
Delete a post	DELETE	/api/posts/*id*	Delete(int id)
Get a post by category	GET	/api/posts?category =*category*	Get(string category)

Telerik Academy

Web API Default Behavior

1. Web Request
2. Match a Route
3. API Controller Responds



Routing

- Routing is how ASP.NET Web API matches a URI to a controller and an action
- Web APIs support the full set of routing capabilities from ASP.NET (MVC)
  - Route parameters
  - Constraints (using regular expressions)
  - Extensible with own conventions
  - Attribute routing is available in version 2

Default Route

- Web API also provides smart conventions by default
  - We can create classes that implement Web APIs without having to explicitly write code
  - HTTP Verb is mapped to an action name

```
http://localhost:1337/api/posts
```

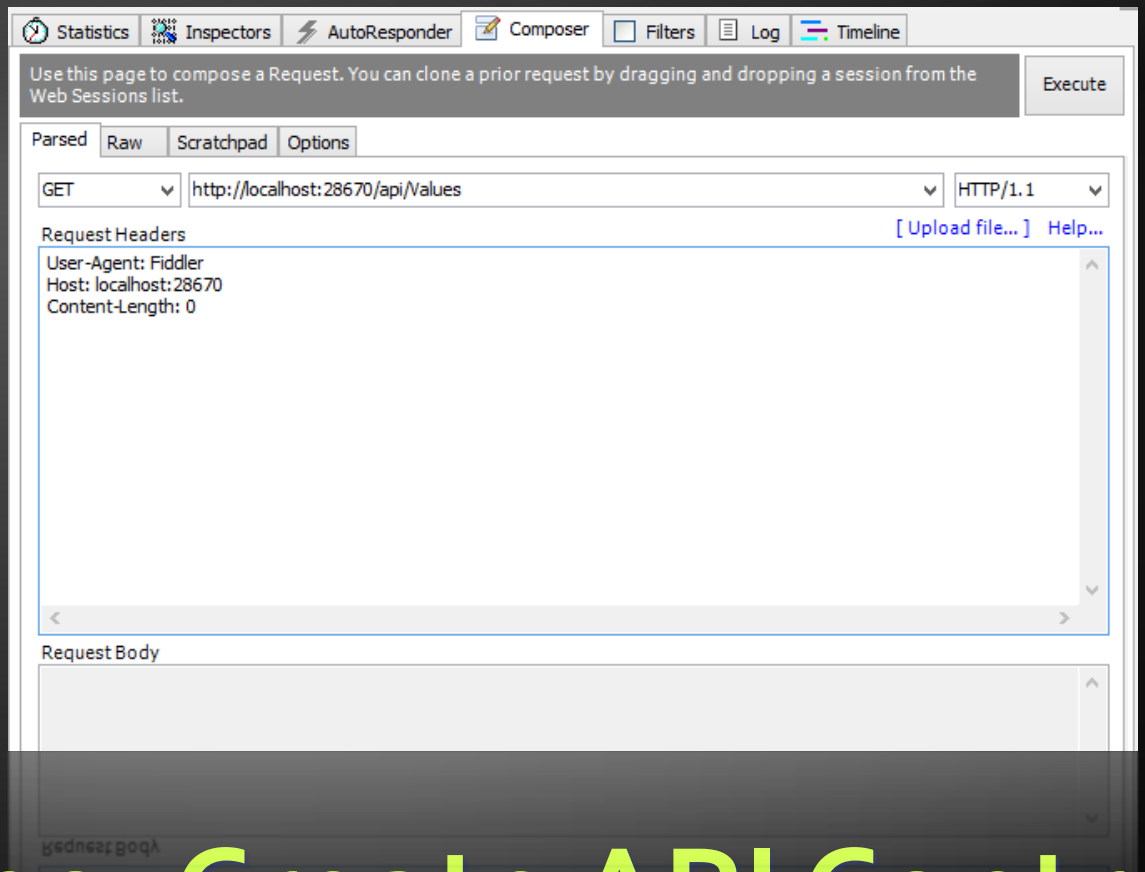
```
routes.MapHtpRoute(name: "DefaultApi",
    routeTemplate: "api/{controller}/{id}",
    defaults: new { id = RouteParameter.Optional });
```

Telerik Academy

Model Binding & Formatters

- By default the Web API will bind incoming data to POCO (CLR) types
  - Will look in body, header and query string
  - ASP.NET MVC has similar model binder
- MediaTypeFormatters are used to bind both input and output
  - Mapped to content types
- Validation attributes can also be used
- To go down further into the HTTP (set headers, etc.) we can use HttpRequestMessage and HttpResponseMessage

Telerik Academy



Demo: Create API Controller

Return Different HTTP Code

- By default when everything is OK, we return HTTP status code 200
- Sometimes we need to return error

```
public HttpResponseMessage Get(int id)
{
    if (dataExists)
    {
        return Request.CreateResponse(
            HttpStatusCode.OK, data);
    }
    else
    {
        return Request.CreateErrorResponse(
            HttpStatusCode.NotFound, "Item not found!");
    }
}
```

OData Query Syntax

- OData (http://odata.org) is a open specification written by Microsoft
  - Provide a standard query syntax on resources
- Implemented by WCF Data Services
- ASP.NET Web API includes automatic support for this syntax
  - Return IQueryable<T> instead of IEnumerable<T>

OData Query Syntax

- To enable OData queries uncomment "config.EnableQuerySupport();" line
- Then we can make OData queries like: "http://localhost/Posts?$top=2&$skip=2"

Option	Description
$filter	Filters the results, based on a Boolean condition.
$inlinecount	Tells the server to include the total count of matching entities in the response. (Useful for server-side paging.)
$orderby	Sorts the results.
$skip	Skips the first n results.
$top	Returns only the first n the results.

Telerik Academy

Web API Clients

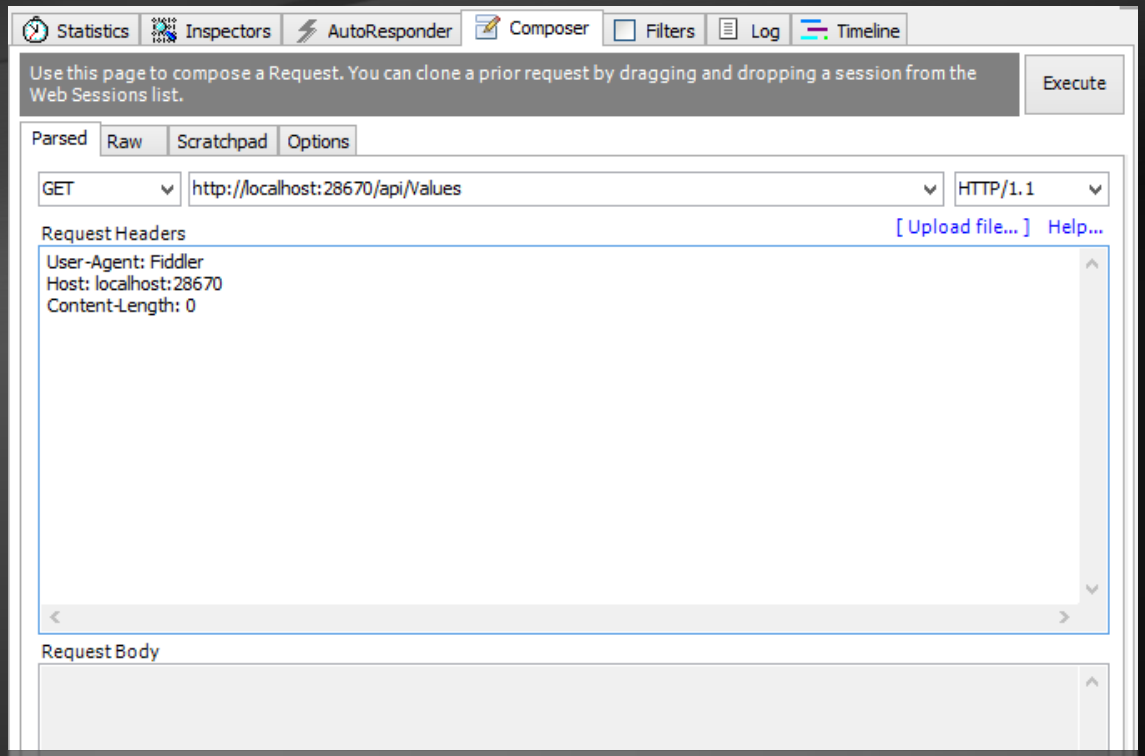
HttpClient Model

- HttpClient is a modern HTTP client for .NET
  - Flexible and extensible API for accessing HTTP
- Has the same programming model as the ASP.NET Web API server side
  - HttpRequestMessage / HttpResponseMessage
- Uses Task pattern from .NET 4.0
  - Can use async and await keywords in .NET 4.5
- Installs with ASP.NET MVC 4
  - Can be retrieved via NuGet

HttpClient Example

```
var client = new HttpClient {
    BaseAddress = new Uri("http://localhost:28670/") };
client.DefaultRequestHeaders.Accept.Add(new
    MediaTypeWithQualityHeaderValue("application/json"));
HttpResponseMessage response =
    client.GetAsync("api/posts").Result;
if (response.IsSuccessStatusCode)
{
    var products = response.Content
        .ReadAsAsync<IEnumerable<Post>>().Result;
    foreach (var p in products)
    {
        Console.WriteLine("{0,4} {1,-20} {2}",
            p.Id, p.Title, p.CreatedOn);
    }
}
else
    Console.WriteLine("{0} ({1})",
        (int)response.StatusCode, response.ReasonPhrase);
```

Telerik Academy



Demo: Consume Web API from Console Application

Consuming Web API from JS

- Web APIs can be consumed using JavaScript via HTTP AJAX request
  - Example with jQuery:

```
<ol id="posts"></ol>
<script>
   $.ajax({
      url: '/api/posts',
      success: function (posts) {
         var list = $('#posts');
         for (var i = 0; i < posts.length; i++) {
            var post = posts[i];
            list.append('<li>' + post.title + '</li>');
         }
      }
   });
</script>
```

Should be encoded

Telerik Academy

ASP.NET Web API

http://academy.telerik.com

Homework

1. Using ASP.NET Web API create REST services for the Student System demo from Code First presentation in the Databases course.
   Use high-quality code, use Repository pattern and create services for all available models.
   Do not use scaffolding.

Homework

2. Using ASP.NET Web API and Entity Framework (database first or code first) create a database and web services with full CRUD (create, read, update, delete) operations for hierarchy of following classes:
   - Artists (Name, Country, DateOfBirth, etc.)
   - Albums (Title, Year, Producer, etc.)
   - Songs (Title, Year, Genre, etc.)
   - Every album has a list of artists
   - Every song has artist
   - Every album has list of songs

Homework (2)

3. Create console application and demonstrate the use of all service operations using the HttpClient class (with both JSON and XML)
4. * Create JavaScript-based single page application and consume the service to display user interface for:
   - Creating, updating and deleting artists, songs and albums (with cascade deleting)
   - Show pageable, sortable and filterable artists, songs and albums using OData

Free Trainings @ Telerik Academy

- C# Programming @ Telerik Academy
  - csharpfundamentals.telerik.com
- Telerik Software Academy
  - academy.telerik.com
- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy
- Telerik Software Academy Forums
  - forums.academy.telerik.com