

.NET Web APIs

Tips, Tricks, & Best Practices

Jonathan "J." Tower

Hi, I'm J.

Jonathan "J." Tower

Principal Consultant & Partner

Trailhead Technology Partners



TRAILHEAD
TECHNOLOGY PARTNERS

trailheadtechnology.com

🏆 Microsoft MVP in ASP.NET

🏆 Telerik/Progress Developer Expert

🏠 Organizer of Beer City Code

✉ jtower@trailheadtechnology.com

🌐 trailheadtechnology.com/blog

🐦 [jtowermi](https://twitter.com/jtowermi)

github.com/jonathantower/dotnet-apis

If You Give \$50, So Will I!

bit.ly/api-water

"charity:water is a non-profit organization that provides clean and safe drinking water to people in developing nations. The organization was founded in 2006 and has helped fund 22,936 projects in 24 countries, benefiting over 4.6 million people." - Wikipedia

*"4/4 Stars"
- CharityNavigator.org*



charity: water

Overview



Quick Refresher on RESTful

Routing

Inputs and Outputs

Validation

Error Handling

Authentication

3rd Party tools

Performance tips

A Note about .NET Core...



ASP.NET *and* ASP.NET Core

Where is Web API in .NET Core?

Web API is dead; long live Web API!

RESTful Refresher

Representations

XML

```
<Person>
  <ID>1</ID>
  <Name>M Vaqqas</Name>
  <Email>m.vaqqas@gmail.com</Email>
  <Country>India</Country>
</Person>
```

JSON

```
{
  "ID": "1",
  "Name": "M Vaqqas",
  "Email": "m.vaqqas@gmail.com",
  "Country": "India"
}
```

Headers

Request

Accepts: application/json

Response

Content-Type: application/json

Both

Authorization

HTTP Methods

GET – Provides a read only access to a resource.

PUT – Used to create a new resource.

DELETE – Used to remove a resource.

POST – Used to update an existing resource or create a new resource.

OPTIONS – Used to get the supported operations on a resource.

URIs

GET <http://www.example.com/customers>
GET <http://www.example.com/customers/33245>
POST <http://www.example.com/customers>
PUT <http://www.example.com/products/66432>
DELETE <http://www.example.com/products/66432>

HTTP Response Codes

200 OK

201 Created

400 Bad Request

401 Unauthorized

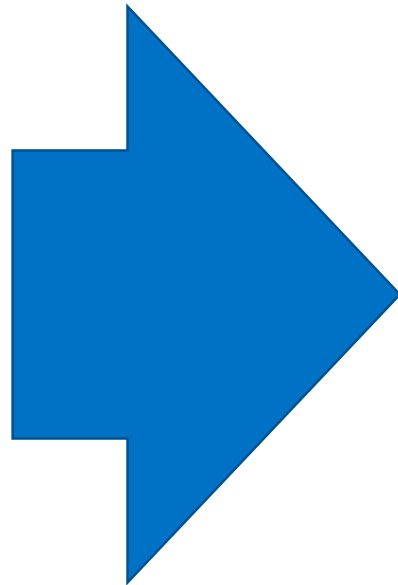
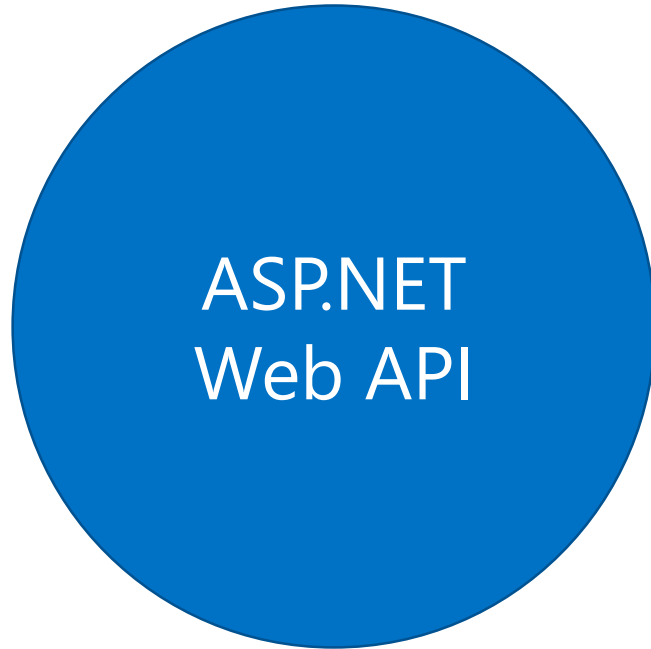
403 Forbidden

404 Not Found

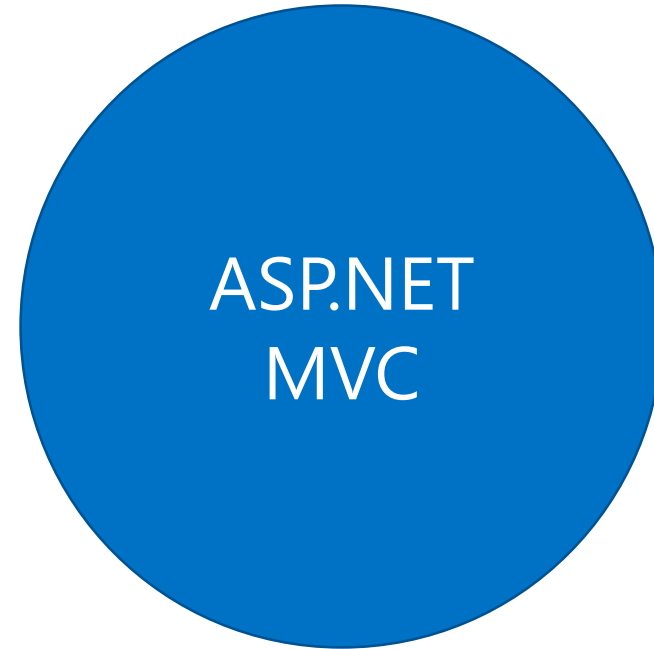
500 Internal Server Error

ASP.NET's RESTful

ASP.NET



ASP.NET Core



Routing

Differences: .NET & .NET Core

.NET – System.Web and System.Http

.NET Core - Merged

Routing Rules

```
routes.MapRoute(  
    name: "default",  
    template: "{controller=Home}/{action=Index}/{id?}");  
  
routes.MapRoute(  
    name: "default",  
    template: "{controller=Home}/{action=Index}/{id:int}");  
  
routes.MapRoute(  
    name: "us_english_products",  
    template: "en-US/Products/{id}",  
    defaults: new { controller = "Products", action = "Details" },  
    constraints: new { id = new IntRouteConstraint() },  
    dataTokens: new { locale = "en-US" });
```

Attribute Routing

```
public class MyDemoController : Controller
{
    [Route("")]
    [Route("Home")]
    [Route("Home/Index")]
    public IActionResult MyIndex()
    {
        return View("Index");
    }
    [Route("Home/About")]
    public IActionResult MyAbout()
    {
        return View("About");
    }
}
```


Attribute Routing

```
[Route("products")]
public class ProductsApiController : Controller
{
    [HttpGet]
    public IActionResult ListProducts() { ... }

    [HttpGet("{id}")]
    public ActionResult GetProduct(int id) { ... }
}
```

Global Config in .NET

```
var json = GlobalConfiguration.Configuration.Formatters.JsonFormatter;
// format datetimes as UTC
json.SerializerSettings.DateTimeZoneHandling = DateTimeZoneHandling.Utc
// Used MS date format "\/Date(1356044400000+0100)\/"
json.SerializerSettings.DateFormatHandling =
    DateFormatHandling.MicrosoftDateFormat;

// camelCase JSON
json.SerializerSettings.ContractResolver =
    new CamelCasePropertyNamesContractResolver();

// Indent JSON
json.SerializerSettings.Formatting = Newtonsoft.Json.Formatting.Indented;

// Ignore reference loops
json.SerializerSettings.ReferenceLoopHandling =
    ReferenceLoopHandling.Ignore;
```

DTOs

Mapping Models to DTOs

```
var config = new MapperConfiguration(cfg =>
    cfg.CreateMap<Order, OrderDto>());

var mapper = config.CreateMapper();

OrderDto dto = mapper.Map<OrderDto>(order);
```

Parameters

Conventions

Mapping done by convention

Can be overridden

[FromBody]

```
[HttpPost]  
public void Post(AddValueCommand command)  
{  
}
```

```
[HttpPost]  
public void Post([FromBody]AddValueCommand command)  
{  
}
```

[FromUri]

```
public ValuesController : Controller
{
    public IActionResult Get([FromUri]GeoPoint loc) { ... }
}

// http://localhost/api/values/?Latitude=47.678558&Longitude=-122.130989
```


Override Convention

```
public class BodyParameterBindingConvention : IActionModelConvention
{
    public void Apply(ActionModel action)
    {
        if (action == null)
            throw new ArgumentNullException(nameof(action));

        foreach (var parameter in action.Parameters)
        {
            parameter.BindingInfo = parameter.BindingInfo ?? new BindingInfo();
            parameter.BindingInfo.BindingSource = BindingSource.Body;
        }
    }
}
```

Override Convention

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc(options =>
    {
        options.Conventions.Add(
            new BodyParameterBindingConvention());
    });
}
```

Responses

Types of Responses

void

Single object

List of objects

HttpActionResult (ActionResult in core)

HttpResponseMessage – ASP.NET (not Core)

Exception

Helper methods in base class

Using IHttpActionResult (ActionResult)

Generic and allows for all kinds of different responses

- 200 – object results

- 401 – Unauthorized

- 400 – BadRequest with validation errors

Using Base Helper Methods

ViewResult

JsonResult

Ok

Unauthorized

BadRequest

ContentResult

FileResults

...

Using Exceptions

ASP.NET HttpResponseMessage

ASP.NET Core – custom handler

Validation

Validation

```
using System.ComponentModel.DataAnnotations;

public class LoginDto
{
    [Required]
    public string Username { get; set; }

    [Required]
    [RegularExpression(@"^[a-zA-Z0-9]$",
        ErrorMessage = "Non-alphanumeric characters are not allowed.")]
    public string Password { get; set; }
}
```

Validation

```
public IActionResult Login([FromBody]LoginDto req)
{
    if (!ModelState.IsValid) return BadRequest(ModelState);

    if (req.Username != "jtower" || req.Password == "Password")
        return Unauthorized();

    return Ok();
}
```

Error Handling

Try...Catch

Catch “expected” errors in Try...Catch block

Return meaningful error message and HTTP code

Global Error Handling

Handle unexpected errors centrally

Log and alert sysadmin

Filter some exceptions (HTTP redirect, thread abort, etc...)

Global Error Handling

```
public class UnhandledExceptionLogger : ExceptionLogger
{
    public override void Log(ExceptionLoggerContext context)
    {
        var exceptionString = context.Exception.ToString();
        // log exceptionString
    }
}

config.Services.Replace(typeof(IExceptionLogger),
    new UnhandledExceptionLogger());
```

Global Error Handling

```
GlobalConfiguration.Configuration.IncludeErrorDetailPolicy  
    = IncludeErrorDetailPolicy.Always;
```

Authentication

DEMO: ASP.NET Identity

Demo: JSON Web Tokens (JWT)

3rd Party Tools

Microsoft.AspNet.WebApi.Versioning & Microsoft.AspNetCore.Mvc.Versioning

```
[ApiVersion( "1.0" )]  
[Route( "api/v{version:apiVersion}/helloworld" )]  
public class PeopleController : ApiController  
  
// api/helloworld?api-version=1.0
```

SDammann.WebApi.Versioning

```
config.Routes.MapHttpRoute(  
    name: "DefaultApi",  
    routeTemplate: "api/v{version}/{controller}/{id}",  
    defaults: new { id = RouteParameter.Optional }  
);  
  
ApiVersioning.Configure(config)  
    .ConfigureRequestVersionDetector<DefaultRouteKeyVersionDetector>();  
  
// Namespace MyApi.Version1_1
```

Swashbuckle (Swagger)

```
// .NET
httpConfiguration
    .EnableSwagger(c => c.SingleApiVersion("v1", "My API"))
    .EnableSwaggerUi();

// .NET Core
services.AddMvc();
services.AddSwaggerGen(c =>
{
    c.SwaggerDoc("v1", new Info { Title = "My API", Version = "v1" });
});
```

Swagger UI

petstore.swagger.io

pet

Everything about your Pets

POST

/pet

Add a new pet to the store

PUT

/pet

Update an existing pet

GET

/pet/findByStatus

Finds Pets by status

GET

/pet/findByTags

Finds Pets by tags

GET

/pet/{petId}

Find pet by ID

POST

/pet/{petId}

Updates a pet in the store with form data

DELETE

/pet/{petId}

Deletes a pet

POST

/pet/{petId}/uploadImage

uploads an image

store

Access to Petstore orders

GET

/store/inventory

Returns pet inventories by status

POST

/store/order

Place an order for a pet

GET

/store/order/{orderId}

Find purchase order by ID

pet Everything about your Pets

POST /pet Add a new pet to the store



Try it out

Parameters

Name

Description

body * required

Pet object that needs to be added to the store

(body)

Example Value | Model

```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "doggie",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Parameter content type

application/json

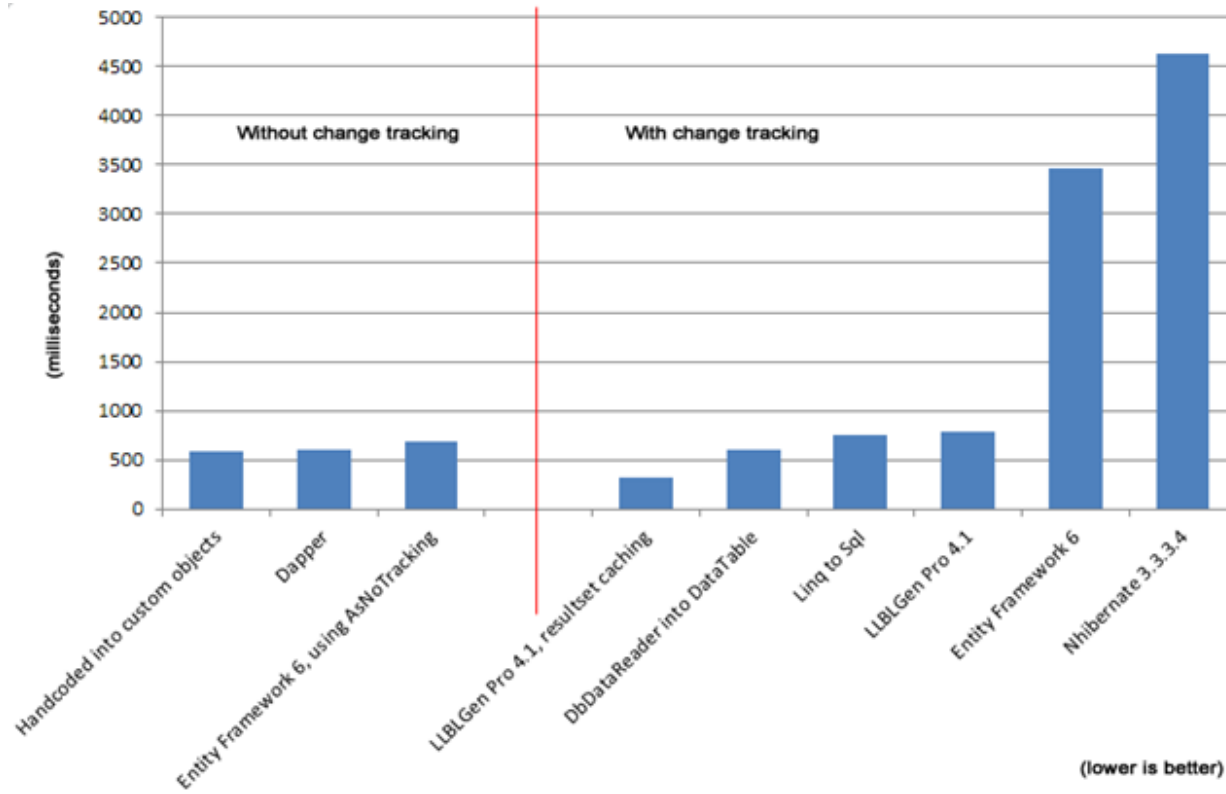
Responses

Response content type

application/xml

Performance

AsNoTracking()



Async/Await

Some tasks require waiting (database access, http calls)

More simultaneous requests!

Analogy: making breakfast

Caching

```
var key = "someKey";  
MemoryCache memoryCache = MemoryCache.Default;  
  
var item = memoryCache.Get(key);  
if (item == null)  
{  
    item = new { }; // todo - go get item  
    memoryCache.Add(key, value, DateTimeOffset.Now.AddMinutes(30));  
}  
  
return item;
```

Recap

Quick Refresher on RESTful

Routing

Inputs and Outputs

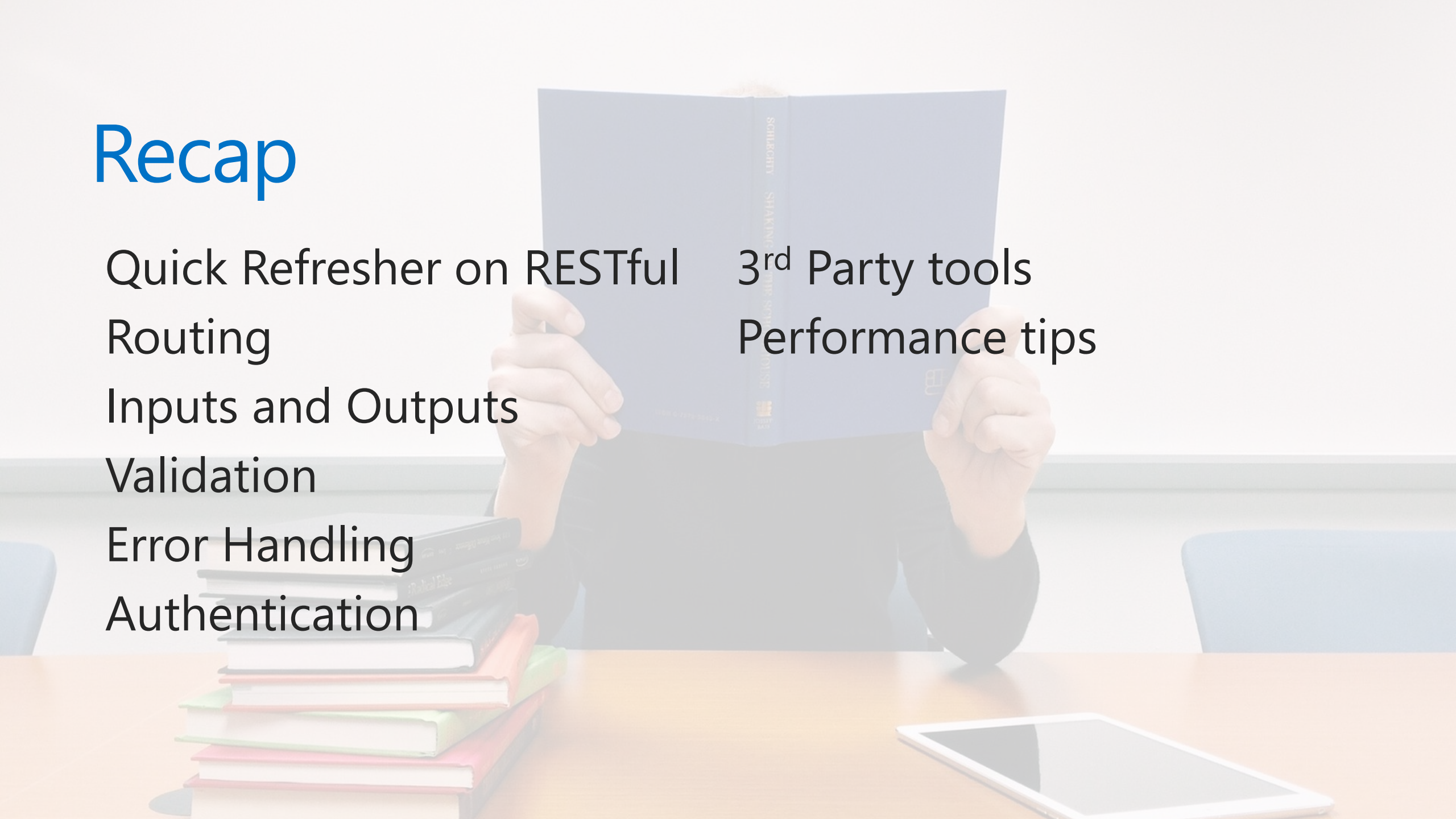
Validation

Error Handling

Authentication

3rd Party tools

Performance tips



Thank You! Questions?

Jonathan "J." Tower

Principal Consultant & Partner

Trailhead Technology Partners



TRAILHEAD
TECHNOLOGY PARTNERS

trailheadtechnology.com

🏆 Microsoft MVP in ASP.NET

🏆 Telerik/Progress Developer Expert

📁 Organizer of Beer City Code

✉ jtower@trailheadtechnology.com

🌐 trailheadtechnology.com/blog

🐦 [jtowermi](https://twitter.com/jtowermi)

github.com/jonathantower/dotnet-apis

If You Give \$50, So Will I!

bit.ly/api-water

"charity:water is a non-profit organization that provides clean and safe drinking water to people in developing nations. The organization was founded in 2006 and has helped fund 22,936 projects in 24 countries, benefiting over 4.6 million people." - Wikipedia

*"4/4 Stars"
- CharityNavigator.org*



charity: water