

Implementing an API in ASP.NET Web API

Module 1: Introduction

Shawn Wildermuth
Wilder Minds LLC
<http://wilder minds.com>



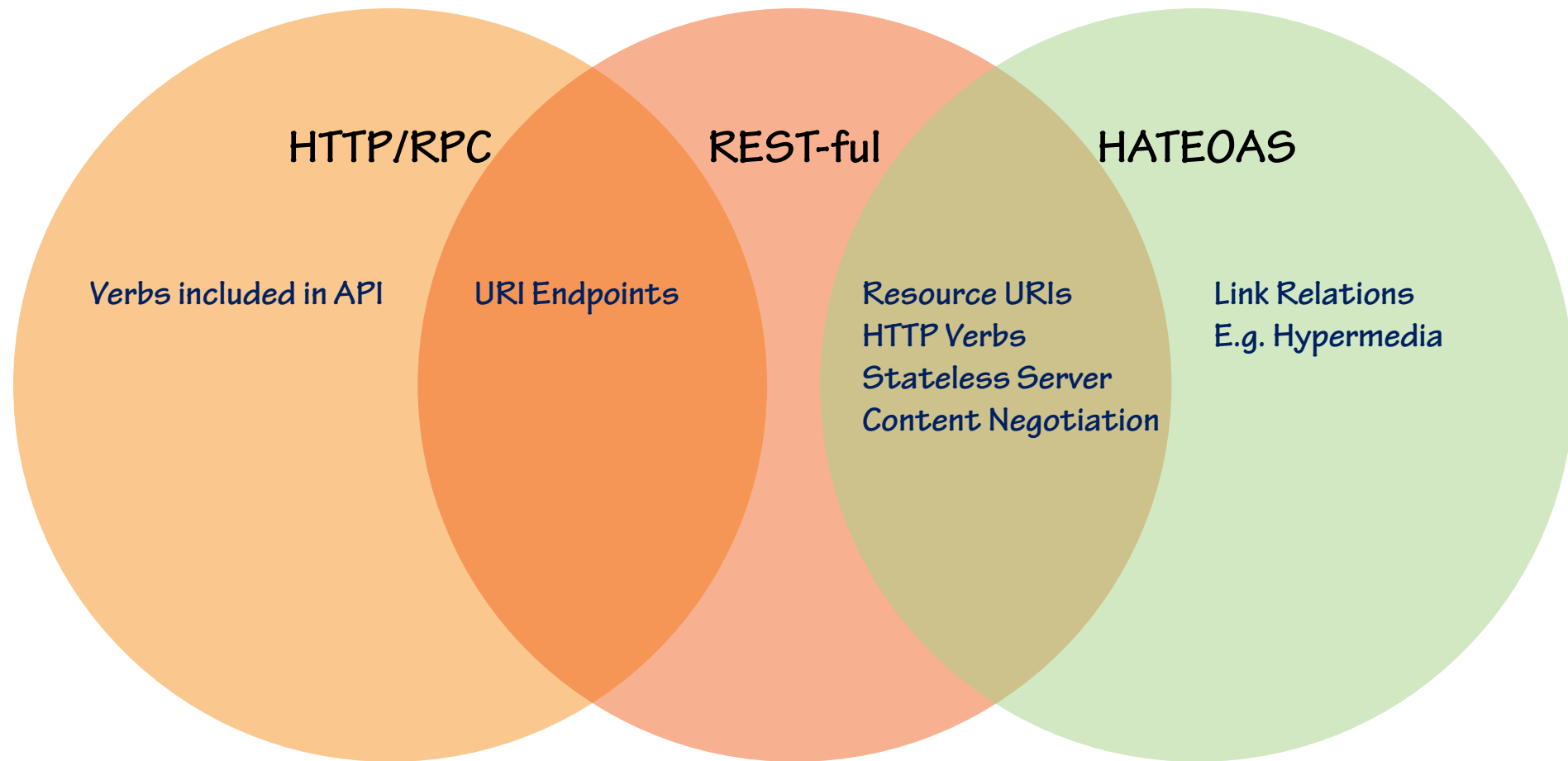
pluralsight
hardcore developer training

Agenda

■ Introduction

- Pragmatic REST?
- Why ASP.NET Web API?
- Getting Web API
- Your First ApiController
- Serialization in Web API
- Dependency Injection
- Models versus Entities
- Routing and Parameter Mapping

APIs on the Web



Pragmatic REST

- **REST is a useful set of rules**
 - But can be dogmatic
 - Developers get caught up ensuring their APIs are completely RESTful
 - Instead of ensuring that the APIs serve their users and customers
 - I propose to use REST without 'throwing out the baby with the bathwater'
 - Do just enough "REST" to ensure you are gaining the benefit of REST

Building REST Services in .NET

- You Have Choices

- ASP.NET WCF

- Shoehorned into SOAP stack

- Configuration can be a headache

- ASP.NET MVC

- Can return JSON but mapping requires overloading and attributes

Building REST Services in .NET

- You Have Choices
 - ASP.NET Web API
 - More natural to implement REST
 - Maps Verbs to Methods
 - Encourages separation and mapping
 - More choices for hosting the API
 - Does require .NET 4.0 or above

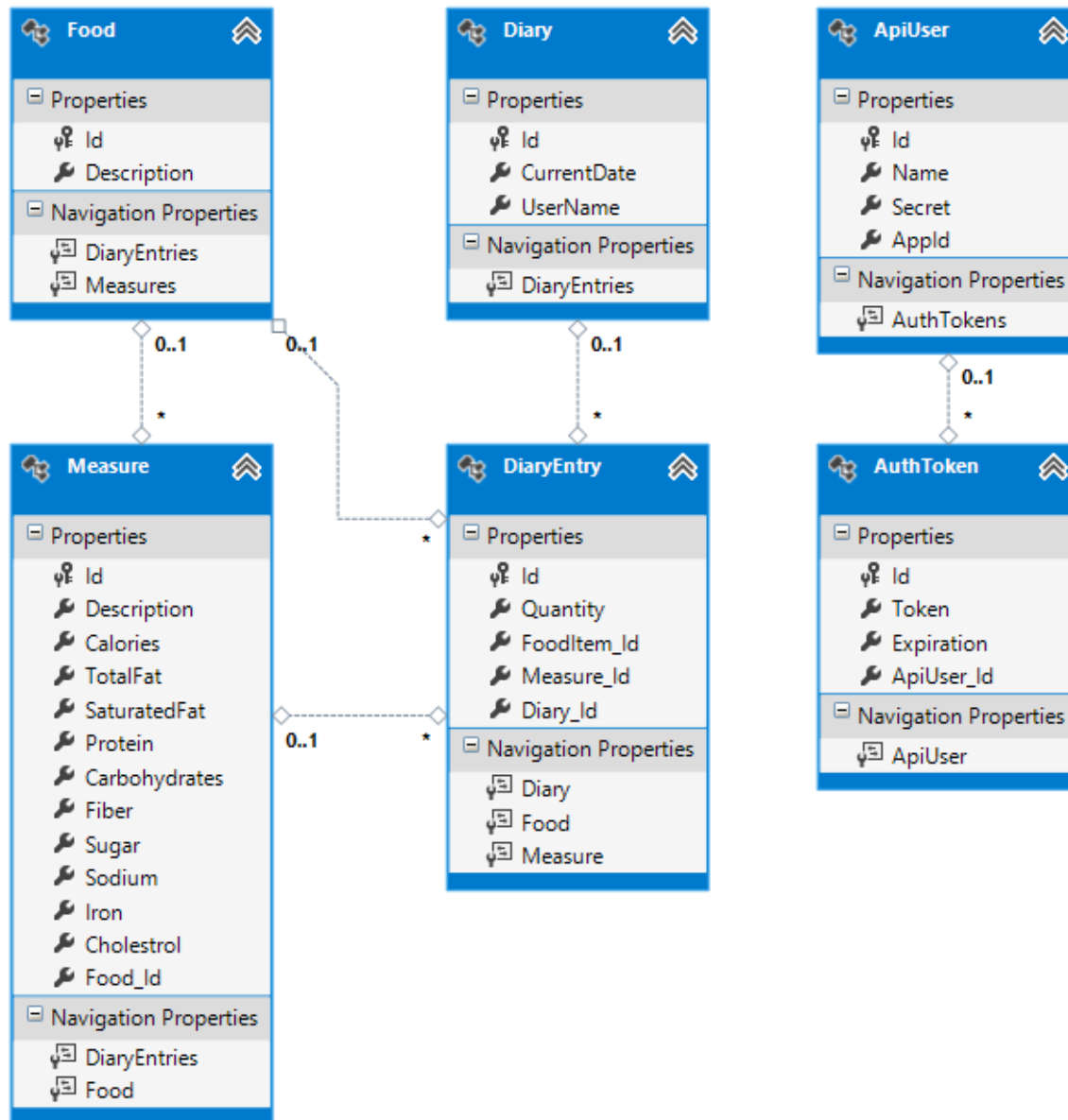
Getting Started with Web API

- **We're Building a Whole API**
 - We're not building the database/data layer in this course
 - We're not building a website to support it
 - We're not building the client side code

Getting Started with Web API

- **Requires Visual Studio 2010+**
 - May require you install ASP.NET MVC 4 to get it in 2010
- **.NET 4.0+**
 - .NET 4.5 added support for async, so that is preferred

The Data



Getting Web API

Adding Web API to Existing Apps

Your First ApiController

Serialization in Web API

Dependency Injection

Models versus Entities

Routing and Parameter Handling

Summary

■ Introduction

- REST is important, but being pragmatic about 'enough REST' works better
- Web API is a good choice for building REST-ful APIs
- You can add Web API to any project that is using .NET 4 and above
- Web API leans on the same HTTP Verb semantic that REST is based on
- Separating your Entities from your API will pay dividends
- Routing and Mapping of Parameters is conventional