

Building an API with ASP.NET Core

PRAGMATIC REST



Shawn Wildermuth

MICROSOFT MVP, INSTRUCTOR, AND FILMMAKER

@shawnwildermuth wildermuth.com

Course Overview



What is this course?

- Build a web API using ASP.NET Core
- Learn by doing
- Feel free to follow along
- You'll see it all

Course Overview



What this course will teach you:

- Creating an API with ASP.NET Core
- Creating API Controllers
- Querying and Modifying Data
- Using Association Controllers
- Defining Operational APIs
- Versioning APIs with MVC 6

Course Overview



What is this course built with?

- Visual Studio 2017 (15.9)
- ASP.NET Core 2.2
- Entity Framework Core 2.2

<http://get.asp.net>

Questions During the Course?

What do you want to learn?

Building a Web App with ASP.NET Core RC1, MVC 6, EF7 & AngularJS

by Shawn Wildermuth

Understanding how to build an app using the full-stack of technologies of web development can be hard. In this course, we'll walk you completely through a complete application to see how everything works together.

[▶ Resume Course](#)

by Shawn Wildermuth

Table of contents | Description | Transcript | Exercise files | **Discussion** | Learning Check | Recommended

Course author

Shawn Wildermuth

Shawn Wildermuth is a thirteen-time Microsoft MVP (ASP.NET/IIS) and is involved with Microsoft as an ASP.NET Insider, ClientDev Insider and Windows Phone Insider.

Course info

Level	Beginner
Rating	★★★★★ (2003)
My rating	★★★★★
Duration	9h 21m
Updated	3 Nov 2015

Do You Already Know ASP.NET Web API?



Microsoft merged the MVC and Web API stacks



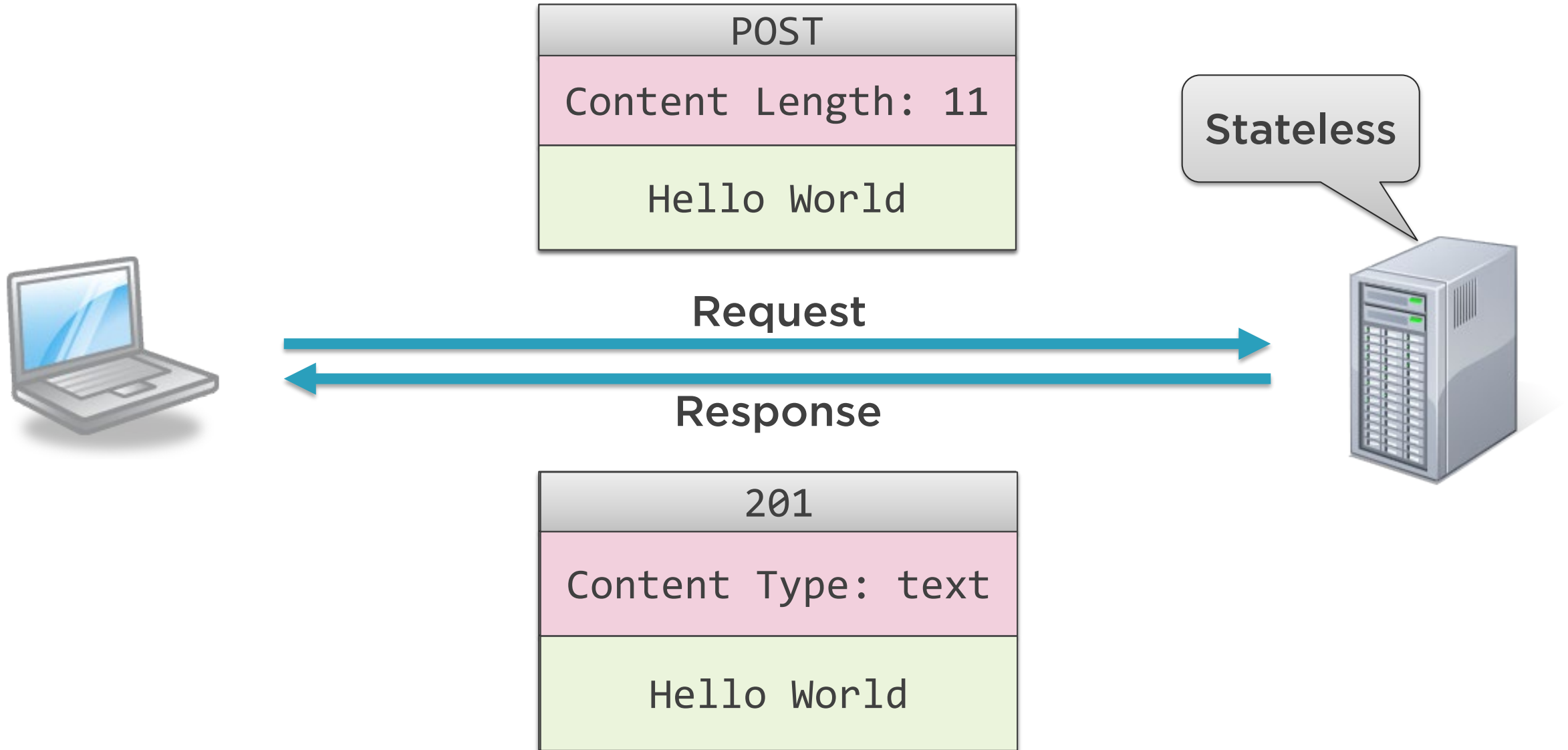
You'll see some parts which are very familiar



Some parts will be very different



How Does HTTP Work?





GET

- Retrieve a resource

POST

- Add a new resource

PUT

- Update an existing resource

PATCH

- Update an existing resource with set of changes

DELETE

- Remove the existing resource

What is REST?

REpresentational State Transfer*

- Concepts include:
 - Separation of Client and Server
 - Server Requests are Stateless
 - Cacheable Requests
 - Uniform Interface

* Introduced by Roy Fielding's doctoral dissertation

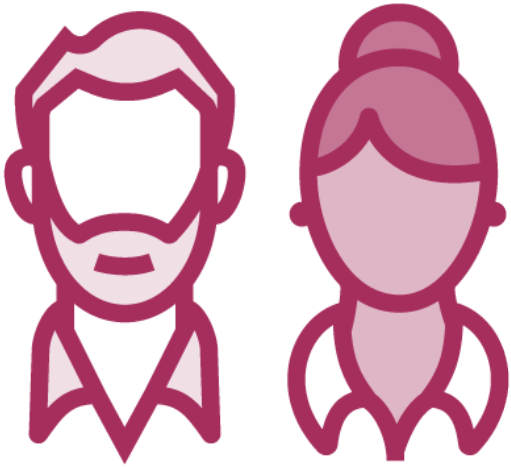
What is REST?

Problems

- Too difficult to be qualified as "REST"
- Dogma of REST vs. Pragmatism
 - Structured architectural style
 - The need to be productive



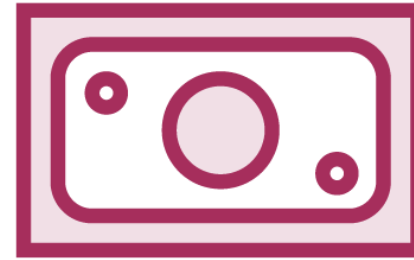
What Are “Resources”?



People



Invoices

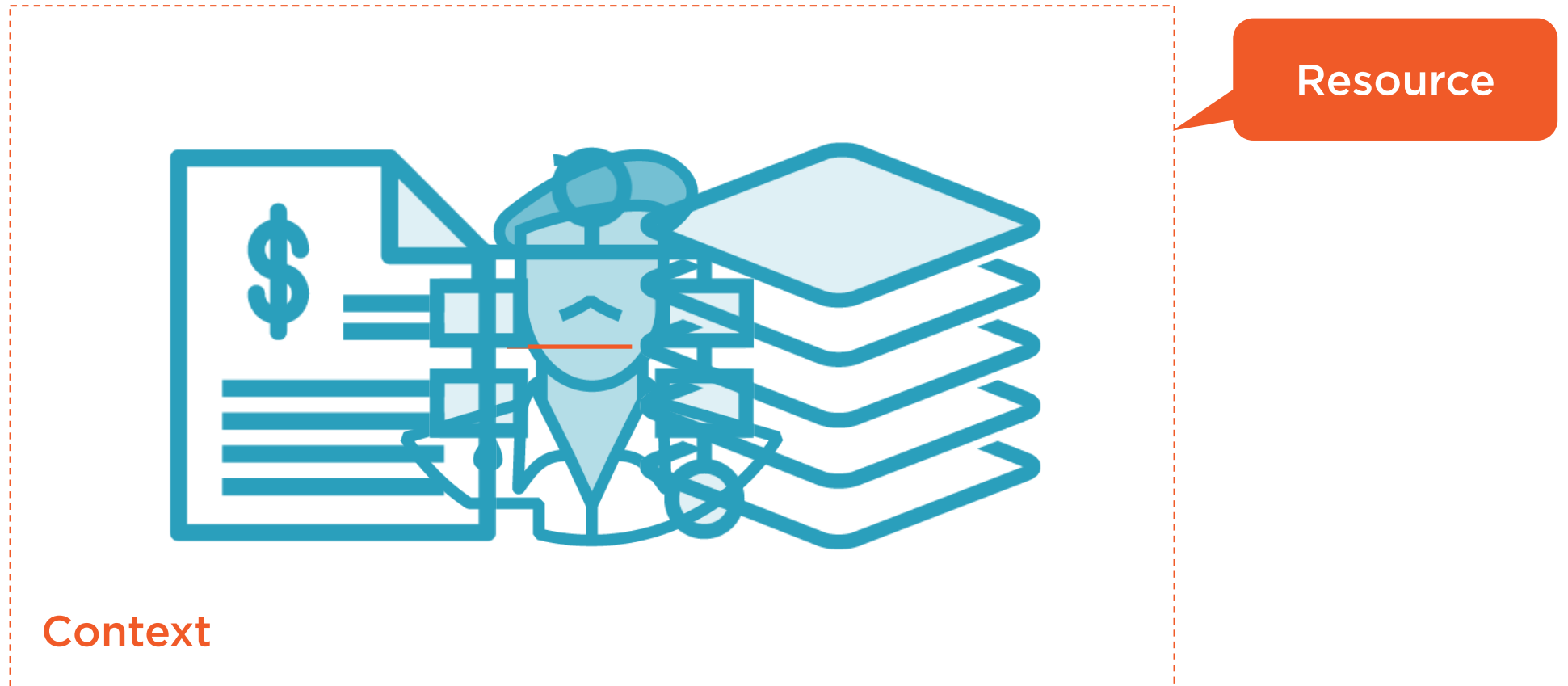


Payments



Products

Are Resources Just Entities?



In Designing APIs, What Are URIs?



URIs are just paths to Resources

- E.g. `api.yourserver.com/people`

Query Strings for non-data elements

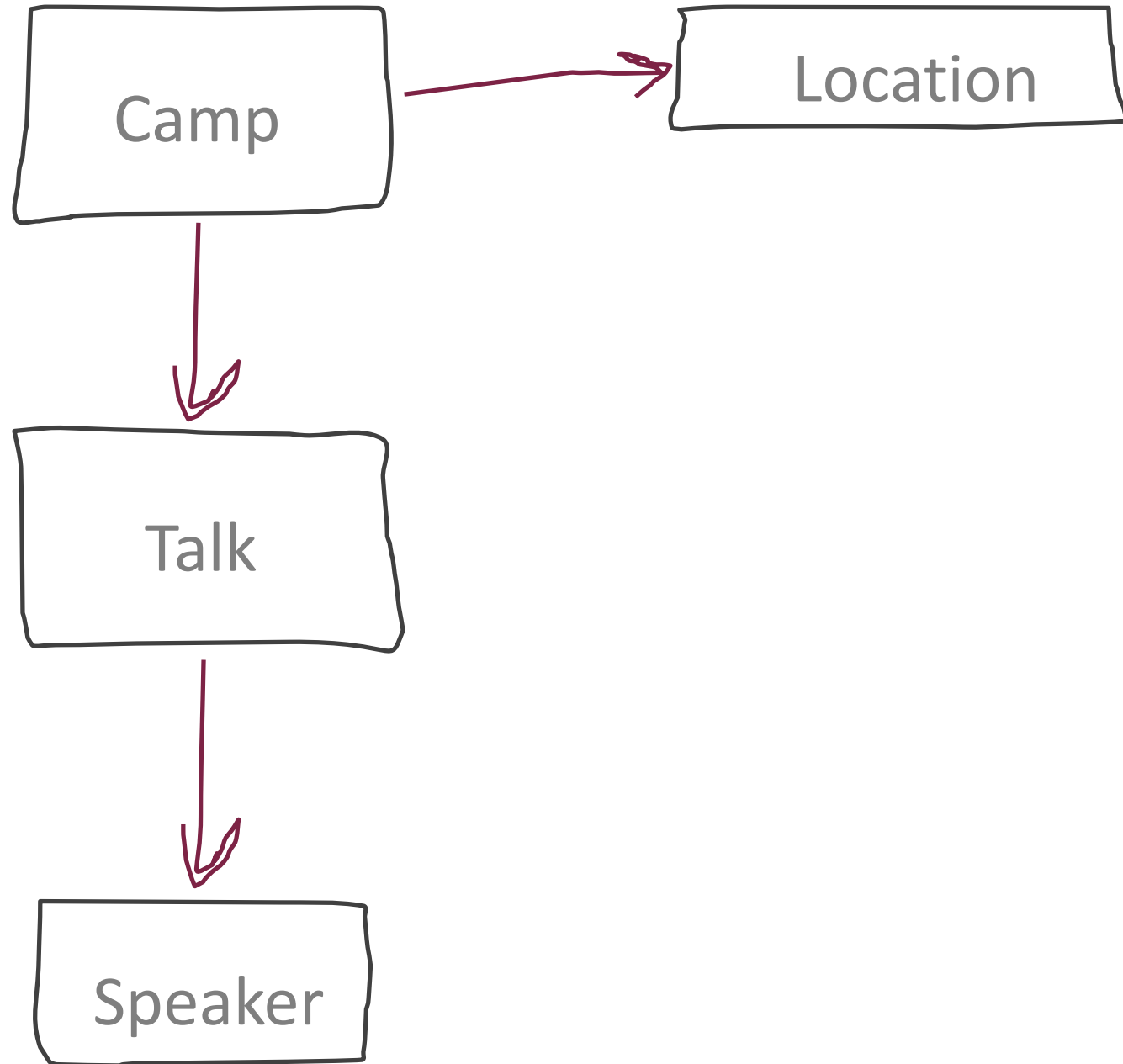
- E.g. format, sorting, searching, etc.

Where We're Going...



We're Building a Whole API

- We're not building the database
- We're not building a website
- We're not building the client-side code



Design the API

`http://.../api/camps`

`http://.../api/camps/ATL2018`

`http://.../api/camps/ATL2018/talks`

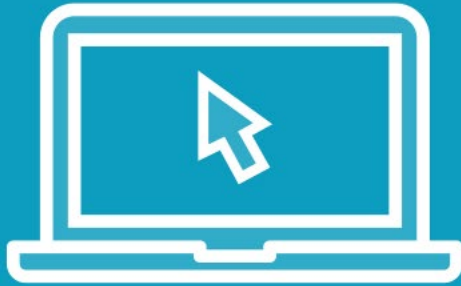
`http://.../api/camps/ATL2018/talks?topic=database`

`http://.../api/camps/ATL2018/talks/1`

`http://.../api/camps/ATL2018/talks/1/speaker`

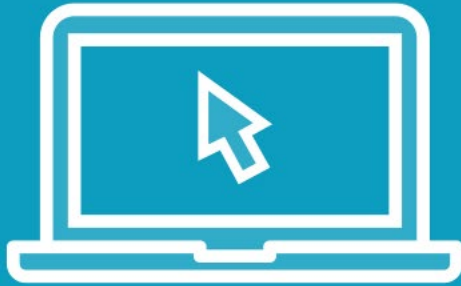
`http://.../api/reloadconfig`

Demo



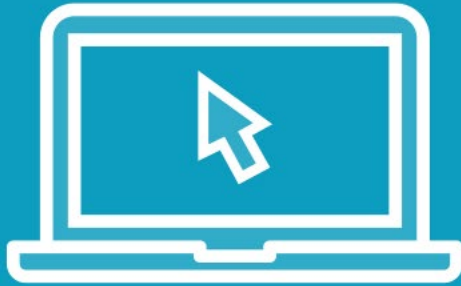
Starting the Project

Demo



Using Postman

Demo



Getting the Project Ready

What We've Learned



HTTP is crucial to the way that APIs on the Web



Using REST is important, but be pragmatic about it



We're building a simple project to learn the basics

Consider adding a “Coming
Up Next” slide here