Designing RESTful Web APIs

WHAT IS REST

Shawn Wildermuth
MICROSOFT MVP, INSTRUCTOR AND FILMMAKER

@shawnwildermuth https://wilderminds.com

Course Overview

What is this course?

- Helps you design Web APIs
- Agnostic to how implemented
- Understand REST's role in APIs
- Introduce you to pragmatic design

Course Overview

What this course will teach you:

- How REST can be used to create APIs
- Designing URI Endpoints
- Understand usage of verbs and nouns
- Why association APIs are important
- What role operational APIs take
- Why versioning APIs is so important
- To decide on securing your APIs

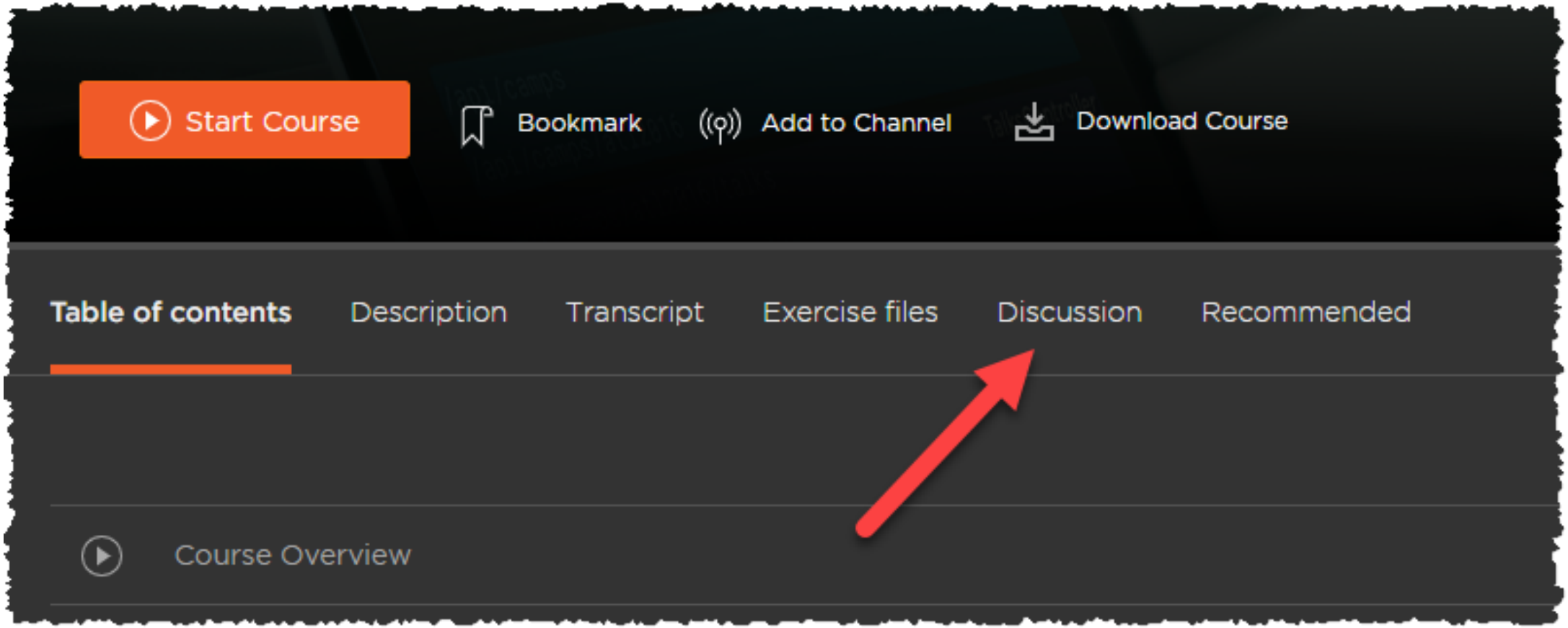
Course Overview

What you will need:

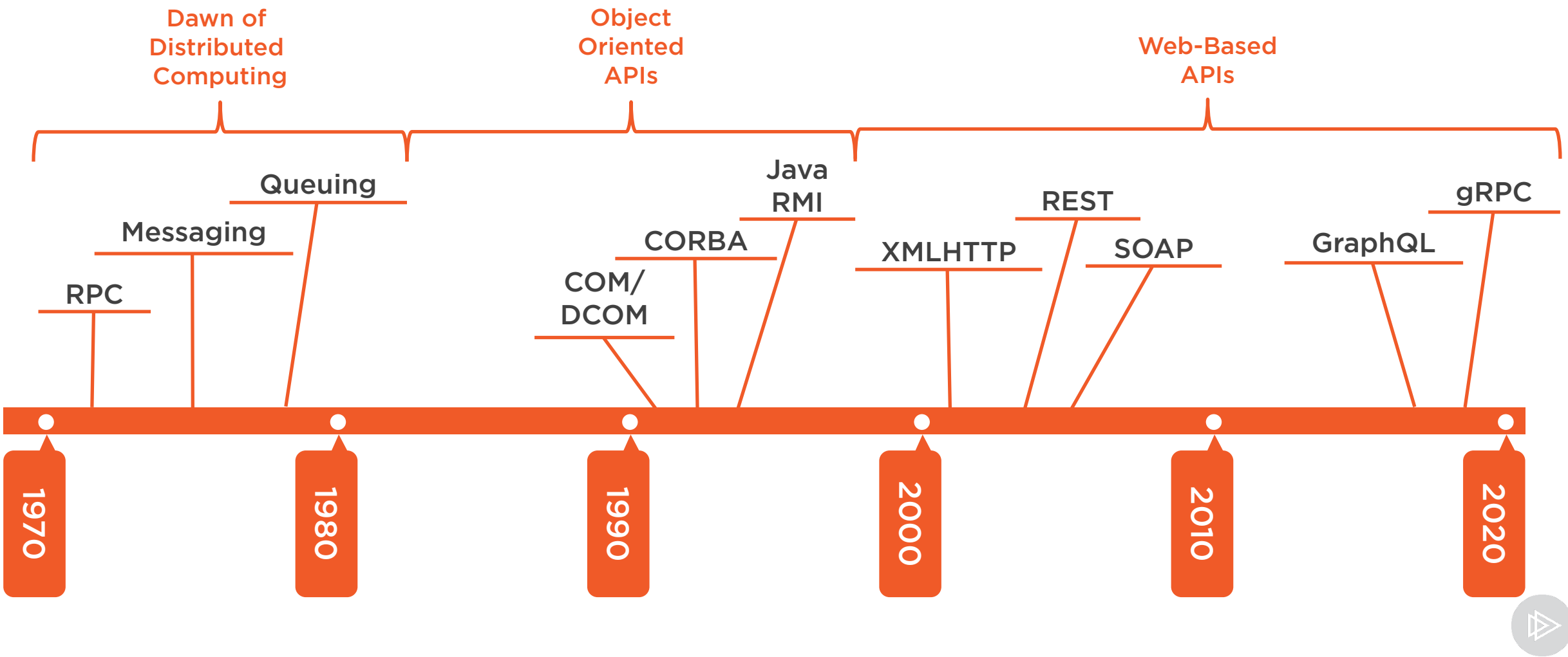
- Postman
  - Operating System doesn't matter

https://www.getpostman.com

Questions During the Course?



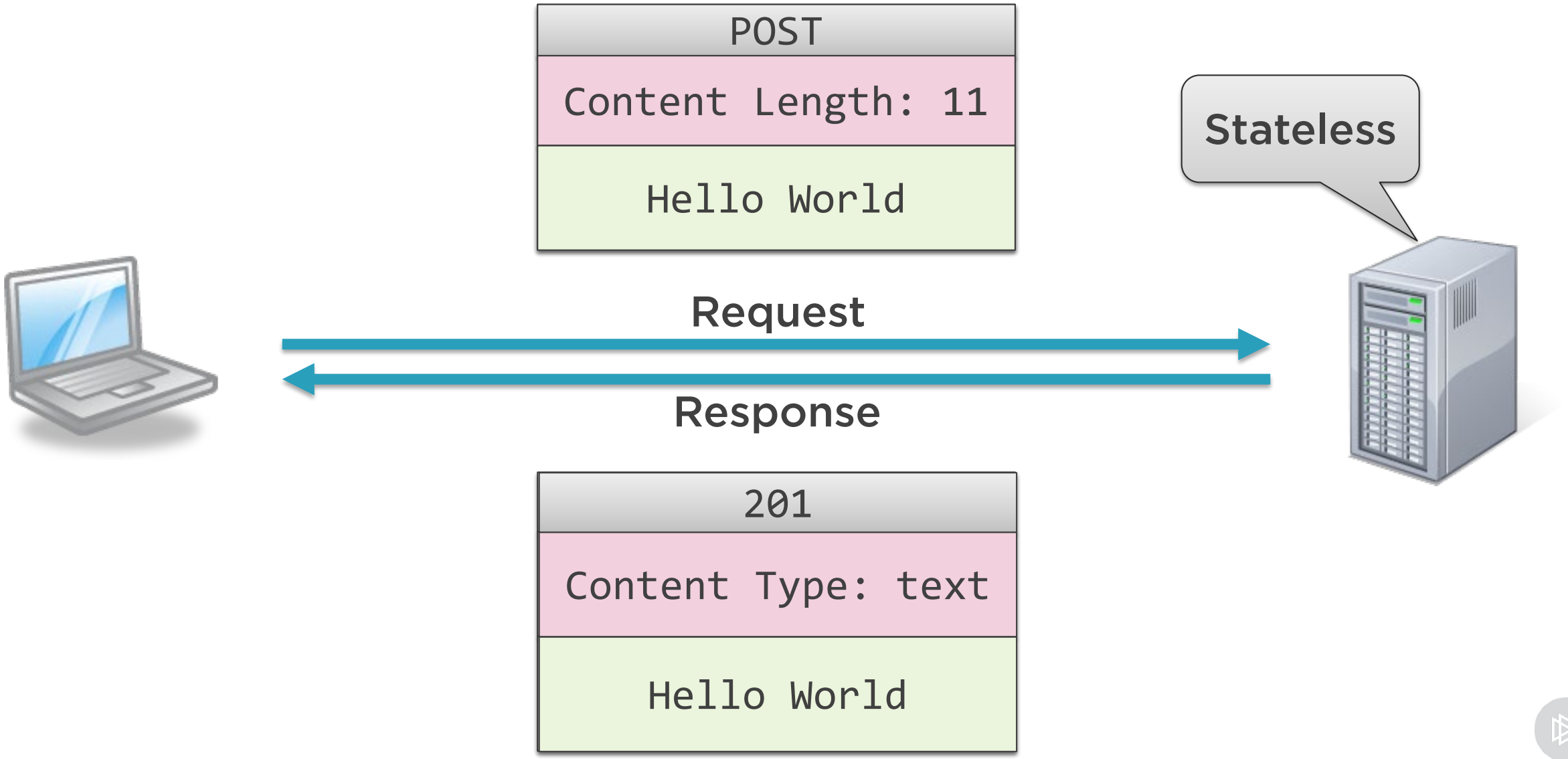
The History of Distributed APIs



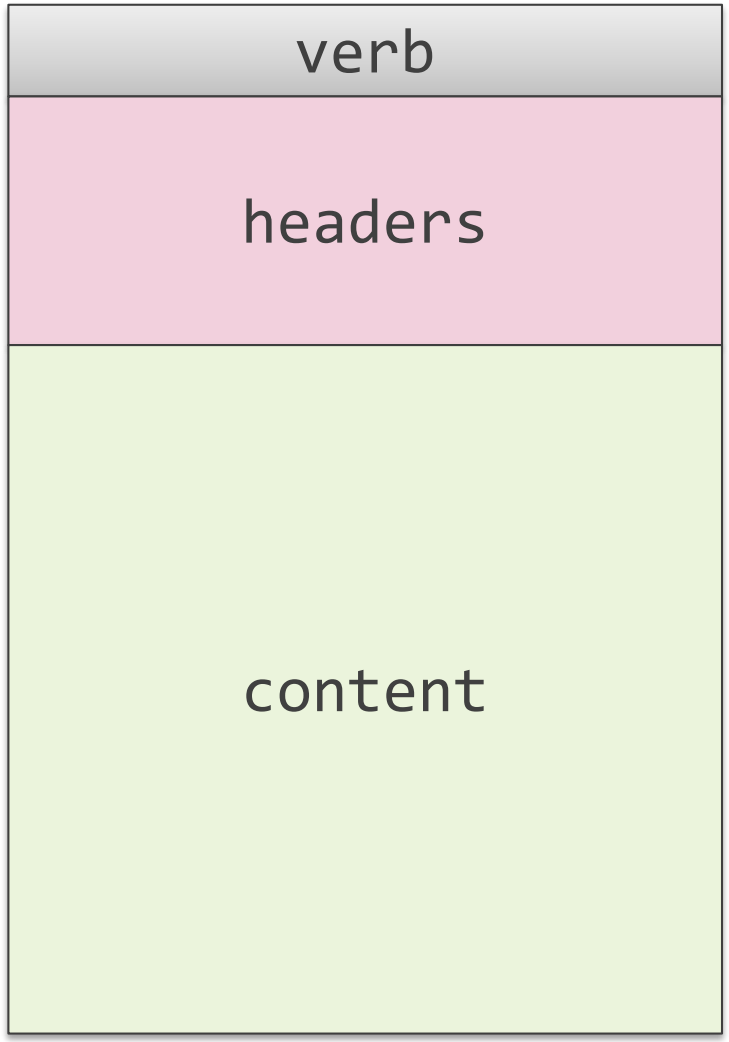
Do You Really Need a Web API?

- Are you building a website?
- Are you building a Single Page App?
- Are you building a mobile app?
- If not, why are you doing it?

How Does HTTP Work?



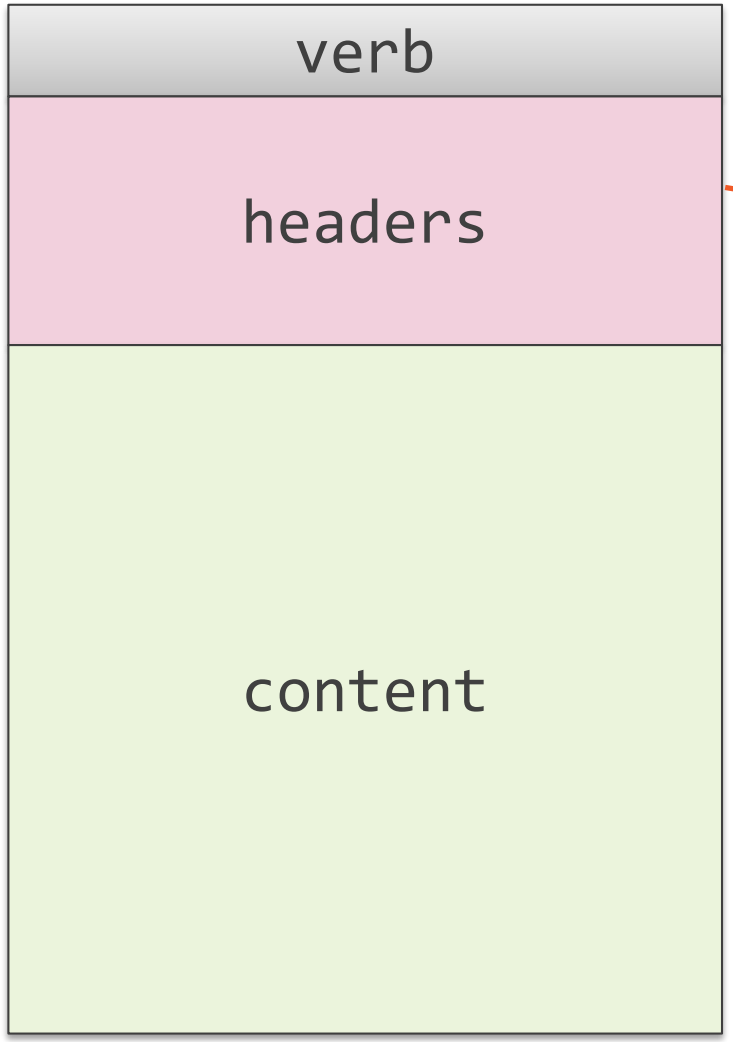
The Request Deconstructed



Action to perform on the server

- **GET**: Request Resource
- **POST**: Create Resource
- **PUT**: Update Resource
- **PATCH**: Update Partial Resource
- **DELETE**: Delete Resource
- More verbs...

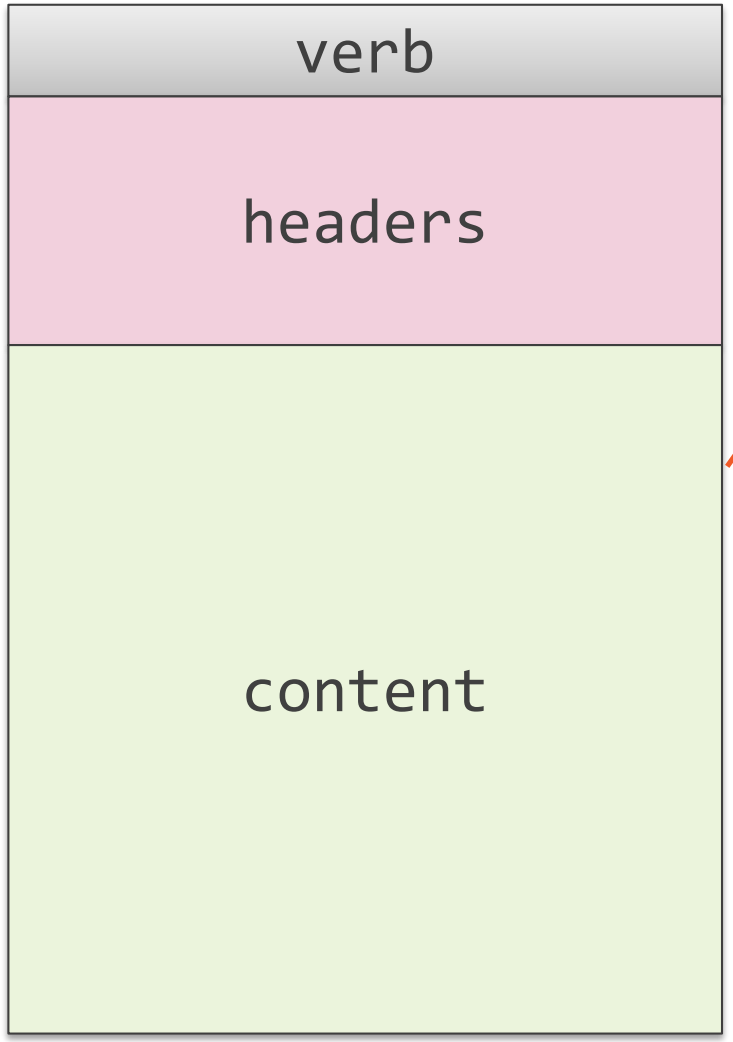
The Request Deconstructed



Metadata about the request

- **Content Type**: The format of Content
- **Content Length**: Size of Content
- **Authorization**: Who's making the call
- **Accept**: What type(s) can accept
- **Cookies**: Passenger data in the request
- More headers...

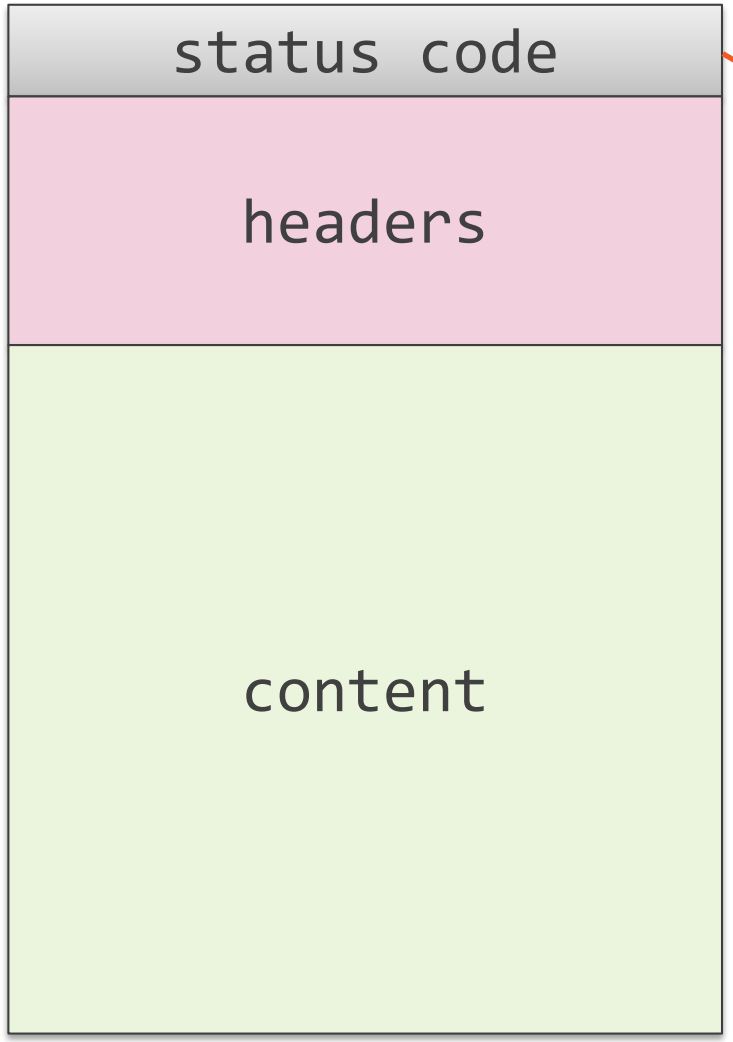
The Request Deconstructed



Content Concerning Request

- HTML, CSS, JavaScript, XML, JSON
- Content is not valid with some verbs
- Information to help fulfill request
- Binary and blobs common (e.g. .jpg)

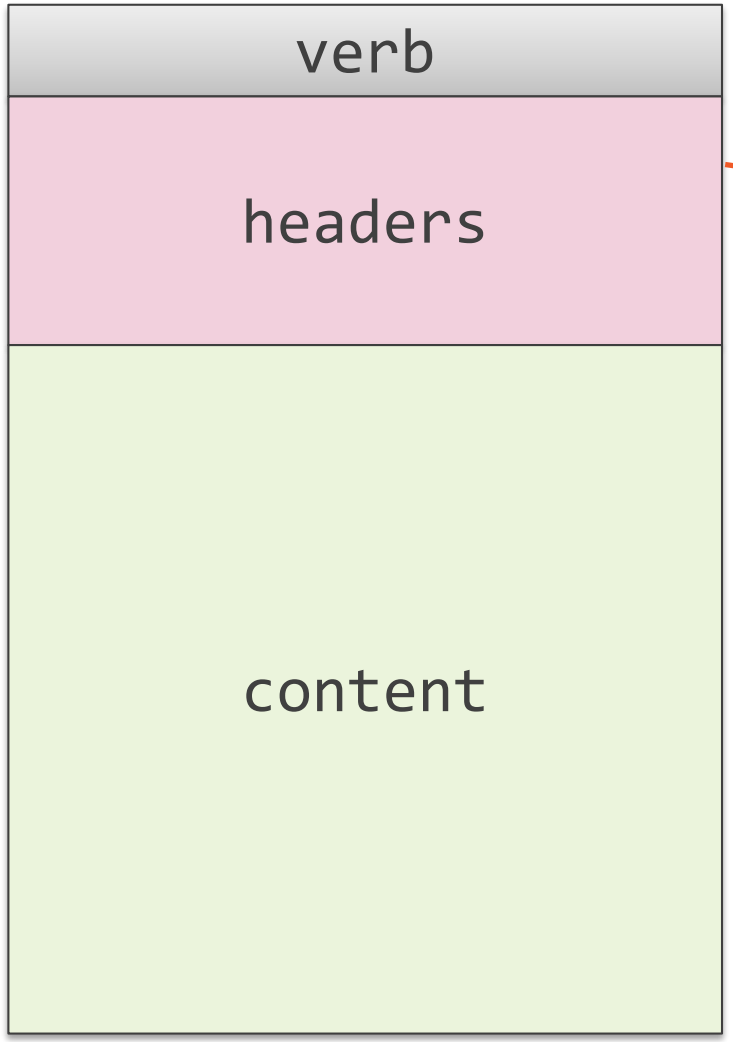
The Response Deconstructed



Operation Status

- **100-199:** Informational
- **200-299:** Success
- **300-399**: Redirection
- **400-499**: Client Errors
- **500-599**: Server Errors

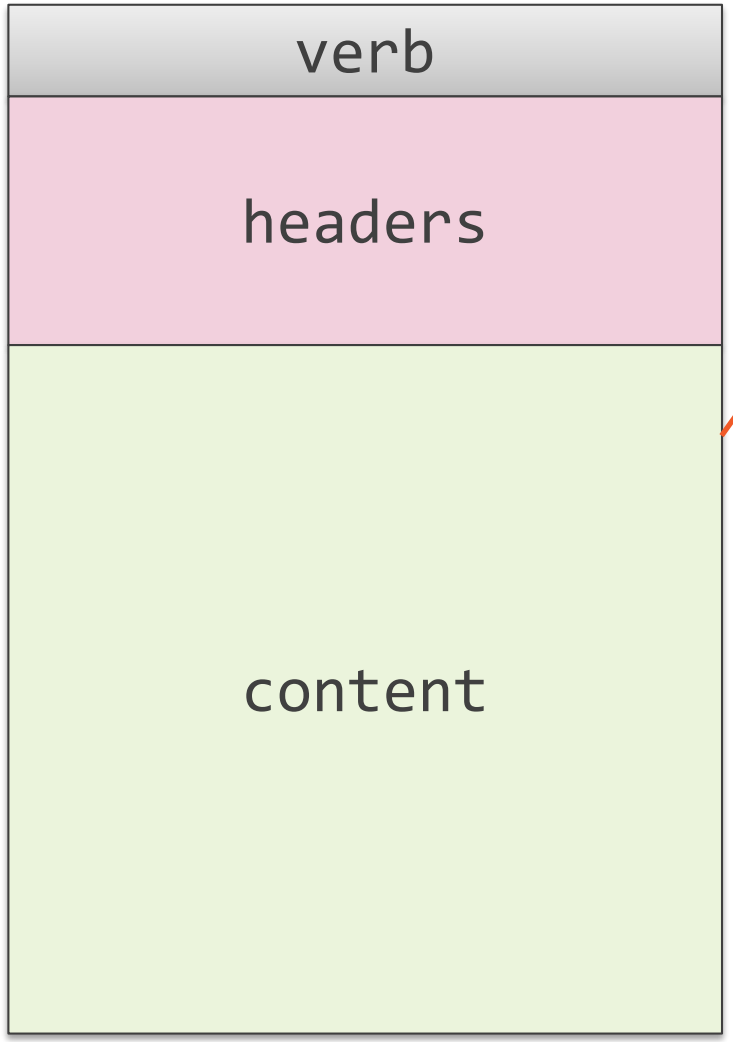
The Response Deconstructed



Metadata about the response

- **Content Type**: The format of Content
- **Content Length**: Size of Content
- **Expires**: When to consider stale
- **Cookies**: Passenger data in the request
- More headers...

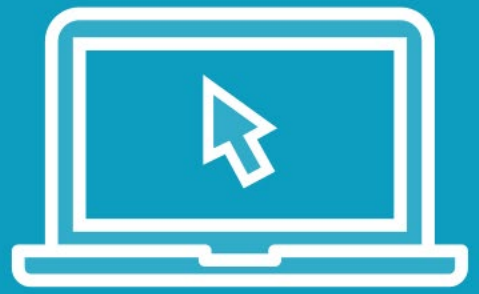
The Response Deconstructed



Content

- HTML, CSS, JavaScript, XML, JSON
- Binary and blobs common (e.g. .jpg)
- APIs often have their own types

Demo



HTTP in Action

What is REST?

REpresentational State Transfer*

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

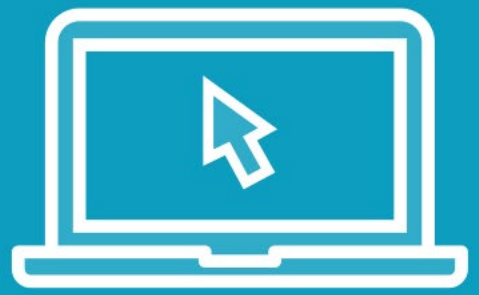
* Introduced by Roy Fielding's doctoral dissertation

What is REST?

Problems

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

Demo



A Well Designed API

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 designing a simple API to learn the basics

Next Up:
Designing a RESTful API