

SpringOne Platform

Building .NET Microservices

Kevin Hoffman & Chris Umbel

@KevinHoffman , @ChrisUmbel

Pivotal

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

About Us



@KevinHoffman

@ChrisUmbel

SpringOne Platform

Open Day

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

§



Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

Building Microservices with ASP.NET Core

DEVELOP, TEST, AND DEPLOY CROSS-PLATFORM SERVICES
IN THE CLOUD

Kevin Hoffman &
Chris Umbel

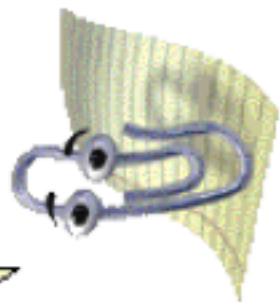
SpringOne Platform

Agenda

- Create ASP.NET Core project
- Add middleware
- Add controller
- Inject a Repository
- Communicate with a Database
- Consume Spring Cloud Config Server
- Participate in Service Discovery
- All on a Mac
 - No Windows were harmed for any of these demos!

It looks like you're trying to build a microservice. How would you like to do that?

- ASP.NET Core
- Java
- Go



ARINI web and service development today

- Maintaining Windows servers carrying pre-cloud baggage
- Traditional MSI-based Windows Server Application Installs
 - Production deploy with RDP, run MSI!
 - Deploy via Active Directory Group Policy
- Maintain and Configure IIS
- Servers are long-lived, bloated pets.
- Extremely difficult for 3-Rs security (Rotate, Repave, Repair)
- Tightly coupled, closed ecosystem
- Mostly closed source
- All-or-nothing monolithic framework





- Modern, immutable, dependency-vendored artifact deploy
- Modular - import only the frameworks you need!
- Easier to maintain Linux servers
- NO IIS.
 - That's right... NO IIS.
- Linux servers are *cloud ready*.
 - Easier to treat servers as disposable
- Open Source
- Did we mention that there is no IIS?
- Core on Windows is ideal stepping-stone from legacy to pure Core
- Core is Microsoft's stated future direction for .NET

- .NET Core
 - Cross-platform, open-source runtime
 - Bare-bones, minimum required to start an app
 - Application types: Console, Web (ASP.NET Core), UWP
 - Part of .NET Foundation <http://www.dotnetfoundation.org/projects>
- Implementation of lessons learned since ASP.NET 1.0 released 14 years ago
 - Yes, we're **THAT OLD.** 😊
- MVC Framework (optional module, like everything else)
 - Server-side rendering/templating
 - Routing
 - **Microservices**

- dotnet new
- dotnet restore
- dotnet run

DEMO

ASP.NET Middleware

SpringOne Platform

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

- Components added to request pipeline to handle requests & responses
- Build pipeline with Run, Map, and Use methods.
- Common middleware:
 - Serving static files
 - Error handling
 - Authentication
 - MVC (Routing and Processing)
- Code can defer, branch, or terminate pipeline

Adding Middleware

DEMO

RESTful Routes

- Route pattern at Class level

SpringOne Platform

Unless otherwise indicated, these slides are © 2013-2016 SpringOne, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

- `[Route("api/[controller]")]`
- Route pattern on HTTP methods
 - `[HttpGet("{id}")]`
- Automatic deserialization
 - **[FromBody]** Monster monster
- Return ActionResult, supports **async** keyword for async actions
- Explicit
 - Can add global context roots (useful in PCF)
 - <http://bit.ly/2ap5gGE>
- Mix-and-Match

Dependency Injection

- Scoped services

- Global services
- Configuration system
- Middleware
- Default IOC container included, but can be customized
 - Autofac
 - Pattern should be familiar to Spring developers



Dependency Injection & Controllers



Delivering Continuously

- “Always Be Deploying”
- Every commit would end up in production today

DEMO

SpringOne Platform

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

- Every commit triggers a build pipeline in CI way
- CI Flow
 - Check in
 - Build Code
 - Run Unit Tests
 - Run UI/JavaScript Tests (if any)
 - Run Integrations
 - Deploy to Dockerhub
 - Push Docker Image to Target Cloud Environment

More CI

- Docker Images + Cloud CI tool:
 - Spin up backing services in containers



- Run integration tests against isolated backing services
- Higher degree of confidence in builds
- *Tested artifact* is the *deployed artifact*, with **no changes**.
- Backing services can be 3rd party products or your own services
- Acceptance Test environments spun up on demand
 - Choose which services are real and which are simulators
 - “Battlefield simulator”

DEMO

SpringOne Platform

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

TF

Introducing SteelToe

- <http://steeltoe.io/>
- Spring Cloud Clients in ASP.NET and ASP.NET Core



- Spring Cloud Configuration Server
- Service Discovery with Eureka
- Spring Cloud Connectors
- Works with OSS products or Pivotal Spring Cloud Services
- Initial Release Q3 2016
- Open Source
 - <https://github.com/SteelToeOSS>

External Configuration

- *application.json file* provides local defaults
- config.AddEnvironmentVariables() adds raw environment vars to config



- config.AddCloudFoundry() adds bound service metadata to config
- Add Configuration Server client to get config from SCCS.
- All configuration is *last-added-wins* - you choose override order

Connectors

- SQL Server + Entity Framework 6 (Windows)
- mySQL
- Redis



- RabbitMQ*

SpringOne Platform

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/0/>

OS

Data Services in Core

- Database Clients
 - SQL Server
 - Postgres
 - ...

- MongoDB
- RESTful DBs
- Not available
 - DB2
 - Oracle (yet)
- Entity Framework 1.0 (not to be confused with EF6)

Database Backing Services via External Configuration



DEMO

SpringOne Platform

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

SS

Service Discovery with Eureka



- Create DiscoveryHttpClientHandler instance
 - Pass as parameter to HttpClient
- Make a simple HTTP request
- SteelToe takes care of URL changes

- Application <https://github.com/joewatson/stackify-api-grabs>

Service Discovery and Consumption



SpringOne Platform

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

ES



DEVIO

SpringOne Platform

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

PS

Summary and Take-aways

- Small gap from empty project to working service
 - No more giant monolith framework
 - “Micro” is the default way of doing things
- Build modern microservices in C# on any device (Mac, Linux, Windows) and deploy to modern PaaS targets

- .NET and Java apps can interoperate in the same microservices ecosystem.
- *We can stop using IIS.*

SpringOne Platform

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

SpringOne Platform

Learn More. Stay Connected.

Pre-order our book, check out the github repo:
<https://github.com/microservices-aspnetcore>
Contribute to Steel/Toe!

Related Session: *Migrating .NET Apps to CF*

 @springcentral
spring.io/blog

 @pivotal
pivotallabs.com

 @pivotalcf
engineering.pivotallabs.com

Unless otherwise indicated, these slides are © 2013-2016 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>