Building Microservices with .NET Core and Steeltoe

Zach Brown and Matt Horan, Pivotal
Introduction
Welcome to the .NET Renaissance

What is Steeltoe?
(and *why* is Steeltoe? and what’s *in* Steeltoe?)

Cloud Foundry

Demo

Roadmap & Resources
Be engaged!

Join the dialogue
Share your thoughts, opinions and photos
...even if you disagree

Zach Brown
@moredeloys

Steeltoe Project
@SteeltoeOSS
Introduction:

Welcome to the .NET Renaissance
Microsoft ♥ Linux
We are on the cusp of a .NET renaissance.

- Ian Cooper
  NDC Oslo
the .NET Renaissance

a convergence of forces

- It’s about time… (.NET’s 15th Birthday)
- Cross-platform .NET Core
- Rise of Cloud
- Growing OSS community
- But we need more projects that support for 12-factor, cloud-native, and microservices
what is Steeltoe?
why is Steeltoe?
• 12 Factor App principles (https://12factor.net/)
• Configuration stored in environment variables
• Out of process session state
• No r/w to registry, GAC, Local file system
• Loosely coupled backing services (i.e. RDBMS)

“Today’s best practices are tomorrow’s anti-patterns.”
Microservices

Microservice architectures have many benefits

- Quickly iterate and release frequently
- Easy for new devs to join and be productive
- Increased developer velocity
- Shorter test cycles
- Polyglot becomes reality
- Independently scalable components
but Microservices are hard

Microservices = Distributed Systems = Way more complexity

- How to troubleshoot microservices?
- How to set global configuration across the whole application?
- How to look up the dynamic addresses of services you consume?
- What do you do when a service you depend on stops responding?
the Good news is...

There are some great OSS solutions already available

and Netflix OSS

But until now they’ve been focused on Java
and not very accessible to .NET devs…
so *that’s* why we built Steeltoe

To help you build

**Cloud native .NET applications**

and leverage Spring Cloud tooling for

**resilient microservices**
what’s in Steeltoe?
Connectors for common data services

- MySQL
- Postgres
- Redis
- RabbitMQ
- OAuth2
Security Providers

- OAuth2 SSO Provider
- JWT resource protection
- Redis-backed Key Ring Repository
cloud-native .NET

Management Endpoints

- /info
  arbitrary app info, e.g. git build tag

- /health
  application health information

- /trace
  circular buffer of last 100 http requests/responses

- /loggers
  shows and modifies configuration of loggers down to the class level
Configuration Providers

Cloud Foundry
- VCAP_APPLICATION, VCAP_SERVICES, CF_*

Config Server
- Access config stored in Spring Cloud Config Server
  (backed by Git, Vault, local filesystem)
- Across all instances, all apps, all environments
Spring Cloud Config Server

1. Push config
2. Source config
3. Pull config
Service Discovery with Eureka

- .NET Client for Netflix Eureka
- Dynamically discover and call registered services
- Supports direct addressing
Circuit Breaker

- .NET implementation of Netflix Hystrix
- Bypass failing services with elegant fall-back behavior
  (so your users don’t see nasty error messages)
- Rich metrics and monitoring
resilient Microservices

Netflix Hystrix

- **Closed**
  - on call / pass through
  - call succeeds / reset count
  - call fails / count failure
  - threshold reached / trip breaker

- **Open**
  - on call / fail
  - on timeout / attempt reset

- **Half-Open**
  - on call / pass through
  - call succeeds / reset
  - call fails / trip breaker

- **Trip Breaker**
  - attempt reset

- **Reset**
Steeltoe works...

with .NET Core and with the .NET framework

on Windows and on Linux

standalone and running on Cloud Foundry
wait...
what’s this Cloud Foundry thing again?
about Cloud Foundry

DYNAMIC ROUTE SERVICES / API MANAGEMENT

YOUR APPLICATIONS

APP MICROSERVICES TECHNOLOGY
- Spring Boot
- Steeltoe
- Spring Cloud Services

DATA MICROSERVICES TECHNOLOGY
- Spring Cloud Data Flow
- Cloud Cache
- RabbitMQ
- MySQL

PLATFORM
- Elastic Runtime
- Concourse
- App Autoscaler
- PCF Metrics
- CredHub
- Orgs, Spaces, Roles and Permissions

EMBEDDED OS
- Windows
- Linux

CONTAINER ORCHESTRATION

CLOUD ORCHESTRATION
- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- Open Stack
- VMWare

MULTI CLOUD
Demo
Roadmap and Resources
What’s coming on the Steeltoe Roadmap?

- Additional Management endpoints
  - Thread dump, Heap dump
- Netflix Ribbon
  - Load balancing for Eureka client
- Distributed Tracing (Zipkin4Net)

What else would you like to see?
The best way to get started is to take a look at our docs, clone a sample from the GitHub repo, and contact the team in Slack with questions and feedback.

- Learn more → http://steeltoe.io
- Clone a sample → https://github.com/SteeltoeOSS/Samples
- Get the bits → https://www.nuget.org/profiles/steeltoe
- Talk to us → http://slack.steeltoe.io/
- Tell your friends → @SteeltoeOSS
other Resources

Upcoming Training
2-day Hands-on training course at Spring One Platform conference
https://springoneplatform.io/training/steeltoe-net-developer

Blogs on Steeltoe
Polyglot microservices and Eureka

Microservices, .NET, Cloud Foundry and Microsoft’s Face API

.NET Core Microservices and Steeltoe