Building Microservices with .NET Core and Steeltoe

Zach Brown and Matt Horan, Pivotal
Outline

• Introduction
  Welcome to the .NET Renaissance

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

• Demo

• Roadmap & Resources
Be engaged!
Join the dialogue
Share your thoughts, opinions and photos
...even if you disagree

Zach Brown @moredepleys

Steeltoe Project @SteeltoeOSS
Introduction:
Welcome to the .NET Renaissance
The .NET Renaissance

“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?
Cloud-Native .NET

- 12 Factor App principles ([https://12factor.net/](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

Benefits of Microservices Architectures
- 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

NETFLIX OSS and spring

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?
Cloud-Native .NET

Connectors for common data services

- MySQL
- Postgres
- Redis
- RabbitMQ
- OAuth2
Cloud-Native .NET

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
Cloud-Native .NET

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
Cloud-Native .NET

Spring Cloud Config Server
Resilient Microservices

Service Discovery with Eureka

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

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
Open and Flexible

Steeltoe works...

with .NET Core and with the .NET framework

on Windows and on Linux

standalone and running on Cloud Foundry
Demo:

ASP.NET Music Store
ASP.NET Music Store

GitHub repo: https://github.com/SteeltoeOSS/Samples/tree/master/MusicStore
ASP.NET Music Store w/Steeltoe - Config Server

GitHub repo: https://github.com/SteeltoeOSS/Samples/tree/master/MusicStore
GitHub repo: https://github.com/SteeltoeOSS/Samples/tree/master/MusicStore
GitHub repo: https://github.com/SteeltoeOSS/Samples/tree/master/MusicStore
ASP.NET Music Store w/Steeltoe - Management

GitHub repo: https://github.com/SteeltoeOSS/Samples/tree/master/MusicStore
What’s coming on the Steeltoe Roadmap

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

What else would you like to see?
How to Get Started

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
Steeltoe 1.1 and the Cloud-Native .NET Renaissance
https://content.pivotal.io/blog/steeltoe-1-1-and-the-net-cloud-native-renaissance

Adding Circuit Breakers to your .NET applications

Polyglot microservices and Eureka