



Agenda - Beispiel

Dies ist unsere empfohlene Agenda für das Thema .NET Advanced Backends. Wir verfügen darüber hinaus über weiteres Kursmaterial, um Themen einfach auszutauschen.

Architekturen und Entwurf von verteilten Systemen

- Motivation für Microservices und Cloud Native Architekturen
- Alternativen zu Microservices: Macroservices / Nanoservices / Event Driven Architectures
- Vergleich von Deployment-Monolithen und verteilten Systemen
- Zerlegen von großen System in Microservices mit Hilfe von Event Storming und Domain Driven Design
- Bounded Contexts und Context Mapping
- Verteilen von Daten über mehrere Services hinweg
- Kommunikation zwischen Microservices
- Umgang mit Konsistenzanforderungen

Containerization

- Motivation für die Containerisierung von Anwendungen
- Erstellung provisionierbarer ASP.NET Core Anwendungen
- Erstellung von Dockerfiles für ASP.NET Core Anwendungen
- Injektion alternativer Konfigurationen via Volume Mounts
- Arbeiten mit eigenen Netzwerken
- Kompilieren von ASP.NET Core Anwendungen innerhalb eines Containers
- Nutzung von Docker Compose um mehrere Container zu orchestrieren
- Debuggen von ASP.NET Core Anwendungen innerhalb von Containern

API Gateways und Backend for Frontend

- Motivation für API Gateways und Backend for Frontend
- Nutzung von YARP Reverse Proxy als einfaches API Gateway
- Aggregation von Daten aus verschiedenen Backend Systemen
- Erzeugen von Json Objekten die für ein Front-End optimiert sind
- Optimierung der Response Zeit und Lastreduzierung mit Caching

Asynchrone Kommunikation

- Grundlagen der asynchronen Kommunikation
- Publish / Subscriber Pattern zwischen mehreren Services
- Zuverlässigkeit und Robustheit mit dem Outbox Pattern erhöhen
- Orchestrierung vs. Choreografie von Microservices
- Saga Pattern mit Kompensationen für verteilte Transaktionen
- Asynchrone Kommunikation zum Front-End via Web Sockets / SignalR

Telemetrie und Monitoring

- Motivation für das sammeln von Telemetriedaten
- Grundlagen von Open Telemetry und des Ökosystems
- Sammeln von System- und benutzerdefinierten Traces
- Sammeln von System- und benutzerdefinierten Metriken
- Sammeln von System- und benutzerdefinierten Logs
- Auswertung von Telemetrie mit open source tools

Authentifizierung und Autorisierung

- OAuth2 und OpenId Connect als moderne Authentifizierungs-Protokolle
- Umgang mit Refresh-Tokens, insbesondere im Browser
- Nutzung eines Auth Gateways für Web und Single Page Applications
- User sessions und sliding expirations
- Weitergabe der Authentifizierungsinformationen an Microservices

Bonus: HATEOAS

Am Ende des Trainings betrachten wir noch den HATEOAS-Ansatz (**H**ypermedia **A**s **T**he **E**ngine **O**f **A**pplication **S**tate) mit dem auf einfachste Weise Zustandsinformationen vom Backend an das Frontend übermittelt werden können und auf dem das State Management im Frontend insbesondere für Single Page Applications (Angular, React, View.js) aufbauen kann.