

# Live Workshop: RESTful APIs in ASP.NET Core



Falko Janak

[contact@falkojanak.com](mailto:contact@falkojanak.com)

<https://www.linkedin.com/in/falko-janak-304b6390/>

Organisiert von

<https://codingmitjannick.de>

# Agenda

- Was ist REST?
- Projektsetup / API Grundstruktur
- Entitäten & API Design
- HTTP Verben und Status Codes
- Webshop-Businesslogik + Datenpersistenz
- API-Dokumentation und Testing
- Q&A

# Was ist REST

“**Representational State Transfer**” - ein Architekturprinzip für verteilte Systeme

- **Ressourcenbasiert:** Jede Entität wird als Ressource betrachtet und über eine eindeutige URL identifiziert.
- **Standardisierte HTTP-Methoden:**
  - **GET:** Abrufen von Daten
  - **POST:** Erstellen neuer Daten
  - **PUT:** Aktualisieren bestehender Daten
  - **DELETE:** Löschen von Daten
- **Zustandslosigkeit:** Jeder Request enthält alle notwendigen Informationen, sodass der Server keinen Client-Zustand speichern muss.
- **Cachebarkeit:** Antworten können markiert werden, um das Caching zu ermöglichen und so die Effizienz zu steigern.

# Best Practices für REST APIs

- **Verwendung von Nomen statt Verben in URLs:**
  - Beispiel: `/orders` statt `/getOrders`.
- **Einheitliche Namenskonventionen:**
  - Nutze konsistente Namenskonventionen (z. B. CamelCase, snake\_case oder kebab-case) für Ressourcen und Parameter.
- **Umgang mit Fehlern:**
  - Implementiere aussagekräftige Fehlercodes und Fehlermeldungen. Zum Beispiel: 404 für „Nicht gefunden“, 400 für „Ungültige Anfrage“.
- **Dokumentation und Selbstbeschreibungsfähigkeit:**
  - Stelle eine API-Dokumentation (z. B. mittels OpenAPI/Swagger) zur Verfügung.
- **Einsatz von HTTP-Headern:**
  - Für Content Negotiation, Authentifizierung und Caching (z. B. `Accept`, `Content-Type`, `Cache-Control`).
- **Sicherheitsaspekte:**
  - Nutze HTTPS, schütze sensible Endpunkte durch Authentifizierung.
- **Versionierung:**
  - Plane von Anfang an eine sinnvolle Strategie zur API-Versionierung ein, um Änderungen ohne Störungen der Clients vornehmen zu können.

# Projektsetup / API Grundstruktur

vgl Projektmappe in Visual Studio

## Entitäten

vgl Projektmappe in Visual Studio

# API Design

- Produkte auflisten
- Produktdetails anzeigen
- Produkt anlegen
- Produkt bearbeiten
- Produkt löschen
- Warenkorb anzeigen
- Produkt zu Warenkorb hinzufügen
- Produkt aus Warenkorb entfernen
- Bezahlvorgang
- Bestellungen auflisten
- Bestelldetails anzeigen

# HTTP Verben und Status Codes

<<live coding>>

# Webshop-Businesslogik + Datenpersistenz

<<live coding>>



# API-Dokumentation und Testing

vgl Swagger

## Nacharbeiten

- DTOs
- Sortierung
- zirkuläre Referenzen

# Q&A

- Fragen?
- Feedback?

Vielen Dank