

# REST API Design

Verständliche, stabile und langfristig nutzbare HTTP-Schnittstellen entwerfen

---

## 01 INTRO

Eine API ist ein Vertrag zwischen Systemen und häufig wesentlich langlebiger als ihre interne Implementierung. Inkonsistente Ressourcenmodelle, unklare Fehlersemantik und unkontrollierte Änderungen führen deshalb schnell zu dauerhaften Integrationsproblemen.

Der Kurs behandelt REST APIs aus der Perspektive des Schnittstellendesigns. Die Teilnehmer modellieren Ressourcen und Operationen, gestalten konsistente HTTP-Schnittstellen und entwickeln Strategien für Fehler, Änderungen, Sicherheit und Dokumentation.

---

### DAUER

2 Tage

---

### FORMAT

Präsenz · Live Online · Inhouse

---

### VORKENNTNISSE

Grundlagen von HTTP und praktische Erfahrung in Softwareentwicklung

---

## 02 TRAININGSPROGRAMM

### Inhalte

#### REST und HTTP einordnen

---

- REST
- Ressourcen
- Repräsentationen
- HTTP
- Client und Server
- Zustandslosigkeit
- Richardson Maturity Model

#### Ressourcen modellieren

---

- fachliche Ressourcen
- URLs
- Collections
- Beziehungen
- Hierarchien
- Aktionen
- stabile Identitäten

## HTTP-Methoden

---

- GET
- POST
- PUT
- PATCH
- DELETE
- Safe
- Idempotent
- Methodensemantik

## Statuscodes und Fehler

---

- Erfolgsantworten
- Clientfehler
- Serverfehler
- konsistente Fehlerobjekte
- Validierungsfehler
- fachliche Fehler
- technische Fehler
- Problem Details

## Repräsentationen

---

- JSON
- Felder
- Datentypen
- Null
- Datumswerte
- Enums
- Links
- konsistente Konventionen

## Collections und große Datenmengen

---

- Pagination
- Filtering
- Sorting
- Searching
- Projection
- Limits
- Performance
- konsistente Parameter

## Nebenläufigkeit und Zustände

---

- ETags
- Conditional Requests
- Optimistic Locking
- Race Conditions
- Idempotency Keys
- Retry
- konkurrierende Änderungen

## API-Sicherheit

---

- Authentifizierung
- Autorisierung
- Tokens
- Input Validation
- Informationspreisgabe
- Rate Limiting
- Transport Security
- Security by Design

## Änderungen und Kompatibilität

---

- Breaking Changes
- additive Änderungen
- Versionierung
- Deprecation
- Consumer
- Evolvability
- Migrationspfade

## OpenAPI und Dokumentation

---

- API Contract
- OpenAPI
- Schema
- Beispiele
- Dokumentation
- Contract First
- Code First

## API-Design bewerten

---

- Konsistenz
- Verständlichkeit
- Robustheit
- Evolvability
- Consumer-Perspektive
- Design Reviews
- typische Anti-Patterns

## API-Design-Workshop

---

- fachlichen Anwendungsfall analysieren
- Ressourcen modellieren
- Operationen definieren
- Fehler gestalten
- Contract erstellen
- Änderungen durchspielen
- Design reviewen

## 03 DURCHFÜHRUNG

## Rahmen und Organisation

|              |                                                             |
|--------------|-------------------------------------------------------------|
| Dauer        | 2 Tage                                                      |
| Durchführung | Präsenz oder Live Online, vorzugsweise als Inhouse-Schulung |
| Schwerpunkt  | Inhouse-Schulungen für Unternehmen und Teams                |
| Sprache      | Deutsch, englische Fachbegriffe und Dokumentation           |
| Unterlagen   | Kursunterlagen, Beispiele und Übungsaufgaben                |
| Technik      | Eigener Rechner für die praktischen Übungen                 |

### Inhouse-Schulungen

Eine eigene bestehende oder geplante API kann als Grundlage für den Design-Workshop dienen.

#### 04 ZIELGRUPPE

## Für wen ist der Kurs gedacht?

Softwareentwickler, Softwarearchitekten und technische API-Verantwortliche.

### Voraussetzungen

Grundkenntnisse von HTTP und praktische Softwareentwicklung.

#### 05 LERNZIELE

## Was Sie nach dem Kurs können

Die Teilnehmer können konsistente REST APIs modellieren, HTTP-Semantik sinnvoll nutzen, stabile Fehler- und Änderungsstrategien entwickeln und Schnittstellen mit OpenAPI dokumentieren.

#### 06 FAQ

## Häufige Fragen

### Wird eine API implementiert?

Kleine Implementierungen können zur Überprüfung des Designs eingesetzt werden. Schwerpunkt ist jedoch das API-Design.

### Wie grenzt sich der Kurs gegen die Testkurse zu APIs ab?

Hier geht es um den Entwurf einer guten API. Der Kurs zu API Testing behandelt, wie eine API systematisch geprüft wird, und der Kurs zur API-Testautomatisierung, wie diese Prüfungen technisch automatisiert werden.

### Werden GraphQL oder gRPC behandelt?

Sie können zur Einordnung herangezogen werden. Schwerpunkt ist REST über HTTP.

---

**UC IT Service** · Ulrich Cuber · kontakt@uc-it.de  
<https://www.uc-it.de/coaching/rest-api-design/>