

Application Programming Interfaces

iSAQB Modul API: Vorschlag von [Thilo Frotscher](#) und [Erik Wilde](#)

Was vermittelt das Modul API?

Application Programming Interfaces, oder APIs, ermöglichen die Nutzung von digitalen Diensten über programmierbare Schnittstellen. Ihre Nutzung hat über die letzten 20 Jahre stark zugenommen. Aus Sicht der Softwarearchitektur sind APIs sowohl aus Produzenten- als auch aus Konsumentensicht relevant: Softwarekomponenten sollten idealerweise als wiederverwendbare Produkte verstanden werden; APIs helfen, eine Softwarekomponente einfach wiederverwendbar zu machen. Heute stehen mehr und mehr Dienste über APIs zur Verfügung; als Konsument dieser Dienste kann man sich besser auf Kernaufgaben konzentrieren und andere Aufgaben auslagern an externe Komponenten.

Das Modul betrachtet die verschiedenen Arten, wie APIs Wert erzeugen: Als technische Schnittstellen die es verschiedenen Komponenten ermöglichen miteinander zu kommunizieren; als organisatorische Schnittstellen die es verschiedenen Teams ermöglicht mit weniger gegenseitigen Abhängigkeiten ihre jeweiligen Komponenten zu entwickeln; und als geschäftsorientierte Bausteine die es der Organisation erlauben, neue Dienste schneller und effektiver zu entwickeln.

Im Rahmen dieses Moduls sprechen wir bei APIs immer von Netzwerk-basierten Schnittstellen, also nicht von lokalen Programmierschnittstellen.

Gliederung des Lehrplans und zeitliche Aufteilung

<i>Inhalt</i>	<i>Vortrag (empfohlene Dauer)</i>	<i>Übungen (empfohlene Dauer)</i>
Übersicht und Einführung	90min	keine
Wertschöpfung von APIs	60min	30min
API Stile und Technologien	60min	30min
Beschreibung von APIs am Beispiel OpenAPI	90min	60min
API Design	90min	60min
API Lifecycle und API Tooling	90min	90min
API Security	60min	30min
APIs at Scale: Platforms und Governance	60min	60min
<i>Summe: 14h</i>	10h	4h

Voraussetzungen für das Modul API

Teilnehmer sollten folgende Kenntnisse und/oder Erfahrungen mitbringen:

- Grundlagen strukturierter Daten, insbesondere JSON und YAML
- Basiskenntnisse von Web-Technologien wie HTTP und URIs
- Basiskenntnisse und Erfahrung mit mindestens einer Programmiersprache

Hilfreich für das Verständnis einiger Konzepte sind darüber hinaus:

- Praktische Erfahrungen in der Programmierung von Web-Anwendungen
- Basiskenntnisse von Sicherheitstechnologien wie SSL/TLS

Detaillierte Beschreibung der Teile

Übersicht und Einführung

- Vortrag: 90min
- Übungen: keine
- Lernziele:
 - Teilnehmer können APIs einordnen in die Geschichte der Programmierung, Computernetzwerke, verteilten Systeme, und Softwarearchitektur.
 - Die Entwicklung von lokalen APIs zu Netzwerk-basierten APIs und die aktuelle API Landschaft werden in ihrem Gesamtkontext verstanden.
 - Der Begriff der "API Economy" wird in seiner gesamten Breite verstanden und Teilnehmer können beurteilen, wie ihre Produkte und Aktivitäten dort eingeordnet werden können.

Wertschöpfung von APIs

- Vortrag: 60min
- Übungen: 30min
- Lernziele:
 - Verständnis der drei grundlegenden Arten wie API Produkte die Wertschöpfung ermöglichen
 - Verständnis der möglichen Geschäftsmodelle von APIs und einiger ausgewählter Beispiele
 - Die Verwendung von APIs als eine Grundlage digitaler Transformation und essentielle Eigenschaften der APIs

API Stile und Technologien

- Vortrag: 60min
- Übungen: 30min
- Lernziele:
 - Kenntnis der fünf grundlegenden API Stile und ihrer Stärken und Schwächen im Vergleich
 - Übersicht über populäre API Technologien und deren Einteilung gemäss der fünf grundlegenden API Stile
 - Verständnis der Kriterien wann welche Stile mehr oder weniger gut passen und der Konsequenzen der Auswahl

Beschreibung von APIs am Beispiel OpenAPI

- Vortrag: 90min
- Übungen: 60min
- Lernziele:
 - Verstehen warum API-Beschreibungen Nutzen erzeugen und von wem sie wofür benutzt werden können
 - Übersicht von OpenAPI als einer konkreten Beschreibungssprache und deren Zielen und Design
 - Verständnis der Hauptelemente einer OpenAPI-Beschreibung im Sinne von Ressourcen, Interaktionen, und Repräsentationen
 - Kenntnis anderer Beschreibungssprachen (AsyncAPI, GraphQL, gRPC) und ihrer Unterschiede im Vergleich zu OpenAPI

API Design

- Vortrag: 90min
- Übungen: 60min
- Lernziele:
 - API Design als "Outside-in" Ansatz bei dem das Design auf Konsumenten fokussiert ist
 - Offenheit und Erweiterbarkeit als wichtige Aspekte um API Produkte weiterentwickeln zu können
 - Verständnis verschiedener Aspekte der Versionierung und wie sie innerhalb des Lebenszyklus eines API Produkts verwendet wird

API Lifecycle und API Tooling

- Vortrag: 90min
- Übungen: 90min
- Lernziele:
 - Verschiedene Schritte des Entwicklungszyklus eines API Produkts
 - Verschiedene Phasen des Lebenszyklus eines API Produkts
 - Übersicht über die Unterschiede zwischen der Sichtweise von Produzenten und Konsumenten
 - Tools zur Unterstützung von Produzenten und Konsumenten wie Linting, Testing, Mocking

API Security

- Vortrag: 60min
- Übungen: 30min
- Lernziele:
 - Kenntnisse der Grundlagen von Kommunikationssicherheit und der Einordnung von TCP, HTTP und TLS.
 - Verständnis von HTTPS, HTTP Authentisierung, OAuth, und OpenID Connect
 - API Gateways und andere Modelle um APIs abzusichern

APIs at Scale: Platforms and Governance

- Vortrag: 60min
- Übungen: 60min
- Lernziele:
 - Plattformen, Platform Engineering, und Platform as a Product als Verbesserung der Developer Experience und Effektivität
 - Verwendung und Pflege von API Guidelines um APIs konsistenter zu entwickeln
 - Beispiele von API Guidelines einiger Organisationen
 - Team Topologies als organisationelles Modell für effektiv arbeitende Teams

Referenzen

- [Mike Amundsen, "RESTful Web API Patterns and Practices Cookbook", 2022](#)
- [James Higginbotham, "Principles of Web API Design: Delivering Value with APIs and Microservices", 2023](#)
- [Gregor Hohpe, "Platform Strategy: Innovation Through Harmonization", 2024](#)
- [Ikenna Nwaiwu, "Automating API Delivery", 2024](#)
- ["Platform Engineering Maturity Model", Cloud Native Computing Foundation, 2023](#)