

Lernziele für das iSAQB Modul API (Application Programming Interfaces)

Thilo Frotscher und Erik Wilde (v3, November 2024)

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.

Zielgruppe

Primäres Publikum sind Software Architekten, Entwickler und andere technisch orientierte Personen, die sich einen umfassenden Überblick über das Thema APIs verschaffen möchten.

Neben dem Schwerpunkt auf technischem Themen werden auch strategische Aspekte behandelt wie die Wertschöpfung und die Verwendungen von APIs als Team-Kollaboration im Unternehmen. Damit eignet sich das Modul gut, um sich dem Thema APIs mit einer weiteren Perspektive als dem rein technischen Blick zu nähern.

Gliederung des Lehrplans und zeitliche Aufteilung

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

Lernziele

1. Übersicht und Einführung

- **LZ 1-1: Einordnung von APIs in die Historie der Software-Entwicklung**

Teilnehmer:innen 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.

- **LZ 1-2: Vergleich unterschiedlicher Integrationsstile und -konzepte**

Unterschiedliche Ansätze zur Systemintegration sind den Teilnehmer:innen bekannt und können gegeneinander abgegrenzt werden, insbesondere

- Integration über eine Datenbank
- Dateibasierte Systemintegration
- Integration durch synchrone Kommunikation (RPC)
- Integration durch asynchrone Kommunikation, z.B. über Message Queues

- **LZ 1-3: Verständnis des Begriffs "API Economy"**

Der Begriff der "API Economy" wird in seiner gesamten Breite verstanden und Teilnehmer:innen können beurteilen, wie ihre Produkte und Aktivitäten dort eingeordnet werden können.

2. Wertschöpfung von APIs

- **LZ 2-1: Kenntnis verschiedener Art der API Wertschöpfung**

Teilnehmer:innen kennen und verstehen die verschiedenen Arten, wie APIs zur Wertschöpfung beitragen können. Die verschiedenen Arten können auf der obersten Ebene unterschieden werden:

- Private APIs die innerhalb einer Organisationen eingesetzt werden
- Partner APIs die mit etablierten Partner genutzt werden
- Public APIs die als Produkte nach aussen angeboten werden

Teilnehmer:innen verstehen die verschiedenen Optionen innerhalb und Zusammenhänge zwischen diesen Kategorien und den Wert einer übergreifenden API Strategie.

- **LZ 2-2: API Business Models**

Teilnehmer:innen haben ein detailliertes Verständnis der möglichen Geschäftsmodelle von APIs und einiger ausgewählter Beispiele. Ausgehend von den verschiedenen Arten der API Wertschöpfung können Teilnehmer:innen bestehende Geschäftsmodelle analysieren und mit APIs gezielt ergänzen und erweitern.

- **LZ 2-3: APIs und Digitale Transformation:**

Teilnehmer:innen können APIs einordnen in das grössere Bild digitaler Transformation. Verschiedene Plattform-Begriffe sind bekannt und ihre Unterschiede und ihr Inneinandergreifen wird verstanden. API Plattformen als eine Art der Verteilung von APIs werden verstanden und ihre Relation zum Konzept einer Internal Developer Platform (IDP) aus dem Bereich des Platform Engineering.

3. API Stile und Technologien

- **LZ 3-1: Kenntnis verschiedener grundlegender API-Stile**

Die folgenden API-Stile werden erläutert und ihre Vor- und Nachteile gegeneinander abgewogen:

- RPC

- Ressourcen-basiert
- Hypermedia
- Query
- Events & asynchrone Kommunikation

Teilnehmer:innen kennen Stärken und Schwächen der einzelnen API-Stile im Vergleich miteinander.

- **LZ 3-2: Populäre API-Technologien und Einordnung gemäß ihrer jeweiligen API-Stile**

Teilnehmer:innen können die wichtigsten populären API-Technologien einordnen und kennen den von ihnen implementierten API-Stil, z.B.:

- gRPC
- HTTP APIs (ohne Hypermedia)
- REST
- GraphQL

- **LZ 3-3: Auswahlkriterien und Konsequenzen von API-Stilen & Technologien**

Teilnehmer:innen erwerben ein Verständnis für Kriterien, wann welche Stile/Technologien besser oder weniger gut passen und welche Konsequenzen die jeweilige Auswahl mit sich bringt.

4. Beschreibung von APIs am Beispiel OpenAPI

- **LZ 4-1: APIs und API Beschreibungen**

Teilnehmer:innen verstehen, warum API-Beschreibungen Nutzen erzeugen und von wem sie wofür benutzt werden können. Das Problem des "API Drift" ist bekannt und Teilnehmer:innen verstehen das Zusammenspiel von Design, Implementierung, Versionierung, Beschreibung, Versionierung der Beschreibung, und wie diese Dinge im Idealfall und in der Realität praktiziert werden.

- **LZ 4-2: OpenAPI als Beschreibungssprache für HTTP APIs**

Teilnehmer:innen verstehen OpenAPI als eine auf HTTP APIs spezialisierte Beschreibungssprache. Sie verstehen die Geschichte von OpenAPI und wie sich die verschiedenen Versionen entwickelt haben. Teilnehmer:innen verstehen wie sich OpenAPI zur Codegenerierung auf der Anbieter- oder Konsumentenseite verwenden lässt. Die generellen Strukturen von JSON und YAML sind bekannt.

- **LZ 4-3: Der Aufbau von OpenAPI**

Teilnehmer:innen verstehen die Hauptelemente einer OpenAPI Beschreibung und wie diese für konkrete APIs verwendet werden. Die Hauptelemente im Sinne von Ressourcen (Paths), Interaktionen, und Repräsentationen sowie Mechanismen wie Tags, Security, Components, und Webhooks.

- **LZ 4-4: Andere Beschreibungssprachen im Vergleich**

OpenAPI ist spezialisiert auf HTTP APIs. Für andere API-Stile können andere Beschreibungssprachen verwendet werden die ähnliche generelle Ziele haben wie OpenAPI. Teilnehmer:innen kennen diese alternativen Beschreibungssprachen und können sie API-Stilen zuordnen.

5. API Design

- **LZ 5-1: Bedeutung von API Design**

Teilnehmer:innen verstehen, weshalb ein gewissenhaftes Design von API-Schnittstellen von großer Bedeutung ist und welchen Einfluss dies auf die Nutzbarkeit, Wartung, Weiterentwicklung und Verbreitung der Schnittstelle haben wird.

- **LZ 5-2: API Design als "Outside-in" Ansatz bei dem das Design auf Konsumenten fokussiert ist**

Zu den wichtigsten Zielen von APIs zählt die effiziente Anbindung von Konsumenten an die Schnittstelle des Betreibers. Vor diesem Hintergrund ist es von großer Bedeutung, das Design auf die Anforderungen der Konsumenten zu fokussieren und nicht etwa entlang eventuell bestehender Systeme, Use Cases oder Datenbanken. Teilnehmer:innen kennen und verstehen die Ansätze von API First und konsumentengetriebenem API-Design.

- **LZ 5-3: Offenheit und Erweiterbarkeit als wichtige Aspekte für die Weiterentwicklung von APIs**

Teilnehmer:innen kennen Postel's Law, die Bedeutung von Forward und Backward Compatibility und welchen immensen Einfluss diese auf die Möglichkeit der Weiterentwicklung von API-Schnittstellen hat. Darüber hinaus erfahren Teilnehmer:innen, welche Konsequenzen dies für die Implementierung von APIs und deren Clients in streng und statisch typisierten Programmiersprache hat.

- **LZ 5-4: Best Practices für gelungenes API Design**

Teilnehmer:innen lernen bewährte und verbreitete Ansätze für das Design von HTTP APIs kennen. Hierzu zählen u.a. URL-Pfade, korrekte Verwendung von HTTP-Methoden oder Operationen für häufig benötigte Funktionalität wie Suchen, Filtern oder Sortieren. Weiterhin sind Formatvorschläge wie JSON:API oder Problem Details for HTTP APIs (RFC 9457) bekannt. Die Vorteile und Herausforderungen eines API Design Style Guides werden von den Teilnehmer:innen verstanden.

- **LZ 5-5: Versionierung von APIs**

Teilnehmer:innen erlangen ein Verständnis für verschiedene Aspekte der Versionierung und wie sie innerhalb des Lebenszyklus eines API Produkts verwendet wird.

- **LZ 5-6: Qualitätsanforderungen: Performance, Skalierung, Caching**

tbd: Sollte dieses Lernziel in einen anderen Abschnitt verschoben werden?

6. API Lifecycle und API Tooling

- **LZ 6-1: API Lifecycle**

Teilnehmer:innen kennen verschiedene Schritte des Entwicklungszyklus eines API Produkts und kennen die typischen Aufgaben die bei diesen Schritten anfallen. Die Aufteilung in Planung und Anforderungsanalyse, Design und Prototyping, Entwicklung, Testing und Qualitätsprüfung, Deployment und Veröffentlichung, Betrieb und Wartung, und Verbesserung und Iteration sind bekannt. Ebenfalls bekannt sind die verschiedenen Lebensphasen eines APIs wie Prototyp, produktiver Betrieb, Deprecation, und das Abschalten.

- **LZ 6-2: APIs als Produkte managen**

APIs kommen oft in lose gekoppelten Szenarien zum Einsatz und aus diesem Grund ist es sinnvoll, sie als Produkte zu managen. Teilnehmer:innen verstehen, was es bedeutet, ein API als Produkt zu managen. Dies beginnt beim Zielgruppenfokus, berücksichtigt Fragen der Nutzbarkeit, beinhaltet Fragen von Feedback und Verbesserung, und behandelt auch Fragen von Deprecation und der Bereitstellung von Alternativen.

- **LZ 6-3: API Lifecycle Tooling**

Teilnehmer:innen kennen typische Tools für den Einsatz im API Lifecycle zur Unterstützung von Produzenten und Konsumenten wie Linting, Testing (z.B. Consumer-driven Contract Testing), Mocking, Betrieb (API Gateways). Teilnehmer:innen können mit einigen dieser Tools praktisch arbeiten und verstehen diese Tools einzuordnen in das Gesamtbild des API Lifecycle Tooling.

7. API Security

- **LZ 7-1: Grundlagen von Kommunikationssicherheit**

Teilnehmer:innen kennen die Grundlagen von Kommunikationssicherheit und können Technologien wie TCP, HTTP und TLS einordnen. Zusätzlich wird den Teilnehmer:innen ein Bewusstsein für die Notwendigkeit sicherer Kommunikation - auch bei unternehmensinternen Schnittstellen - vermittelt.

- **LZ 7-2: Verständnis relevante Technologien im API-Umfeld**

Teilnehmer:innen verstehen, wie HTTPS, HTTP-Authentisierung, OAuth und OpenID Connect funktionieren, inwiefern diese sich unterscheiden und wie die Technologien bei der Absicherung von APIs helfen.

- **LZ 7-3: API Gateways und andere Modelle um APIs abzusichern**

tbid: Welche "anderen Modelle" sollten hier berücksichtigt werden?

8. APIs at Scale: Platforms and Governance

- **LZ 8-1: Vergleich verschiedener Plattform-Begriffe**

Teilnehmer:innen verstehen die verschiedenen Bereiche, in denen der Begriff der Plattform verwendet wird. Sie verstehen insbesondere den Begriff des Platform Engineering, seine Herkunft, und das Konzept einer Internal Developer Platform (IDP). Teilnehmer:innen können zwischen dem mehr nach innen gerichteten Fokus einer IDP und dem sowohl nach innen als auch nach aussen gerichteten Fokus einer API Plattform unterscheiden.

- **LZ 8-2: API Guidelines**

Teilnehmer:innen verstehen die Motivation von API Guidelines, und dort vor allem das Ziel, innerhalb einen gewissen Bereiches eine Harmonisierung von API Design und von API Entwicklungspraktiken zu erreichen. Tools zur Unterstützung von API Guidelines wie Linting Tools sind bekannt und ihre Funktionsweise ist geläufig.

- **LZ 8-3: API Guidelines by Example**

Beispiele der API Guidelines einiger Organisationen sind bekannt. Des weiteren verstehen Teilnehmer:innen in welcher Art API Guidelines entwickelt und gepflegt werden sollten, nämlich als partizipativ und kontinuierlich gepflegtes Dokument von gelebten und unterstützten Praktiken in einer Organisation.

- **LZ 8-4: Team Topologies**

Teilnehmer:innen kennen Team Topologies als organisationelles Modell für effektiv arbeitende Teams. Dabei ist der Fokus vor allem darauf, an welchen Stellen des Team Topologies Modells APIs essentiell notwendig für die Umsetzung des Modells sind. Dies sind zum einen der Konsum von Diensten X-a-a-S Modell, und zum anderen das Anbieten von Diensten vor allem von Platform Teams die ihren Platform Dienst im Self Service anbieten wollen.