

E-Business Architekturen

Prüfungsleistung (Gruppenaufgabe)

Ergebnisprotokolle der Komplexübungen 1, 2, 3b und 4d im Rahmen der
Veranstaltung E-Business Architekturen

vorgelegt am
07.05.2023

an der
Hochschule für Wirtschaft und Recht Berlin
Fachbereich Duales Studium

von:	Robert Neubert Danny Neupauer Hannes Roever
Fachrichtung:	Wirtschaftsinformatik
Studienjahrgang:	WI20C
Studienhalbjahr:	Wintersemester 2022/23
Dozent:	Prof. Dr. Andreas Schmietendorf

Inhaltsverzeichnis

1	Einleitung	1
2	Aufgabe 1: E-Business Grundlagen	1
2.1	Anwendung des Begriffs E-Business	1
2.2	Beziehung zu domänenspezifischen Lösungen	1
2.3	Ziele und Erwartungen an E-Business Lösungen	1
2.4	Eigenschaften von E-Business Softwarearchitekturen	2
2.5	E-Business im konkreten Unternehmenskontext	2
3	Aufgabe 2: Serviceverzeichnisse	2
3.1	Analyse eines Verzeichnisdienstes	2
3.1.1	Vorgehensweise bei der Auswahl eines Verzeichnisdienstes	3
3.1.2	Allgemeines über den Verzeichnisdienst	3
3.1.3	Vergleich von API- und datenorientierten Schnittstellen	5
3.2	Analyse von Web-APIs	6
3.2.1	Erstellung des Bewertungsmodells	6
3.2.2	Einsatz des Bewertungsmodells	8
3.3	API - Spezifikationen	8
3.3.1	Spezifikationsanalyse	8
3.3.2	Analysetools	9
3.3.3	Einschränkungen und Alternativen	9
4	Übung 3b: Entwicklung eigener Service-Angebote	9
4.1	Möglichkeiten für Implementierung und Deployment	9
4.1.1	Analyse der Möglichkeiten	9
4.1.2	Analytischer Vergleich der Möglichkeiten	9
4.2	Entwicklung	9
4.2.1	Rahmenbedingungen	9
4.2.2	Implementierung API	10
4.2.3	Anbindung Datenbank	10
5	Übung 4d: Sicherheit von Web APIs	10
5.1	Sicherheitsrisiken in Verbindung mit dem HTTP Protokoll	10
5.2	Möglichkeiten zur Risikominderung	10
5.2.1	OWASP	10
5.2.2	OAuth 2 und OIDC	10
5.3	Praktische Anwendung von OAuth2	11
5.3.1	Testwerkzeuge	11
5.3.2	Implementierung	11

1 Einleitung

lalala¹ und sind insofern für komplexe sozialwissenschaftliche Fragestellungen aufgrund des vorhandenen Bias²
lalalala³

2 Aufgabe 1: E-Business Grundlagen

2.1 Anwendung des Begriffs E-Business

Was verbinden Sie mit dem Begriff des E-Business? Versuchen Sie die folgenden Aspekte zu berücksichtigen, nennen Sie ggf. weitere.

- Organisatorische Aspekte
- Prozessbezogene Aspekte (z.B. Geschäftsprozess)
- Technologische Aspekte (z.B. Entwicklung & Betrieb)
- Gesellschaftliche Implikationen (z.B. Soziologische Aspekte)

2.2 Beziehung zu domänenspezifischen Lösungen

Welche Beziehungen sehen Sie zu den folgenden Lösungen?

- Systeme für das e-Learning (z.B. Moodle oder Open HPI)
- Systeme für das e-Government (z.B. ELSTER oder Fahrzeugzulassung)
- Systeme für das e-Banking (z.B. Instant Payment)
- Systeme für das e-Commerce (z.B. Web Shops)

2.3 Ziele und Erwartungen an E-Business Lösungen

Welche Ziele und Erwartungen verknüpfen Unternehmen und ihre Kunden mit e-Business-Lösungen?

- Berücksichtigen sie ggf. unterschiedliche Sichten
- Nennen Sie ihnen bekannte Lösungen (z.B. aus den Praktika)
- Identifizieren Sie mögliche Vor- und Nachteile

¹Vgl. Athey (2018), S.5

²Vgl. Miceli et al. (2022), S.3ff

³Vgl. Korab (2021), online

2.4 Eigenschaften von E-Business Softwarearchitekturen

Über welche Eigenschaften sollten Softwarearchitekturen für e-Business-Lösungen verfügen?

- Fragen des Kommunikationssystems
- Verwendete Rechnerinfrastruktur
- Eigenschaften entwickelter Softwaresysteme

2.5 E-Business im konkreten Unternehmenskontext

Wie könnte eine Strategie zur Einführung einer e-Business-Architektur in einem Unternehmen ihrer Wahl aussehen?

- Notwendige Voraussetzungen & Rahmenbedingungen
- Auswirkungen auf das Informationsmanagement (CIO)
- Auswirkungen auf die Entwicklung von Software (Lösungsanbieter)
- Auswirkungen auf den Betrieb von Software (Rechenzentren)
- Mehrwertpotentiale für die Kunden und Lieferanten

Worin sehen Sie weitere Aspekte eines digitalen Unternehmens, die mit dem Begriff des e-Business nicht erfasst werden?

3 Aufgabe 2: Serviceverzeichnisse

3.1 Analyse eines Verzeichnisdienstes

Analysieren Sie die Möglichkeiten eines in Abstimmung mit dem Dozenten zu wählenden Verzeichnisdienstes für Web-APIs.

- Recherche und Auswahl eines Verzeichnisdienstes:
 - Anzahl und Art der registrierten Web-APIs (ggf. auch Open Data)
 - Allgemeiner Funktionsumfang des Verzeichnisdienstes
 - Hinterlegte Klassifikationen – d.h. Organisation der Serviceablage
 - Vorgehensweise zum ggf. Suchen von Serviceangeboten
 - Vorgehensweise zum ggf. Registrieren eigener Serviceangebote
 - Bereitgestellte Entwicklerunterstützung, wie z.B. Beispielcode
- Voraussetzungen zur Nutzung (Registrierung, Kosten, ...)?
- Vergleich von API- und datenorient. Schnittstellen (z.B. Open Data)?

In dieser Unteraufgabe wird sich mit der Nutzung und Analyse von Service-Verzeichnissen beschäftigt. Dazu wird sich zunächst für einen zu betrachtenden Verzeichnisdienst entschieden, welcher im weiteren Verlauf der Aufgabe auf seine Eigenschaften überprüft wird.

3.1.1 Vorgehensweise bei der Auswahl eines Verzeichnisdienstes

Um einen vollumfänglichen Überblick über den Verzeichnisdienst bieten zu können, wurde bei der Auswahl des Verzeichnisdienstes darauf geachtet, einen Verzeichnisdienst ohne Zugangsbeschränkungen mit öffentlich zugänglichen APIs zu wählen. Aufgrund dieser Vorgaben wir uns für das API-Verzeichnis des Bundes entschieden.

Die APIs, die unter der Webadresse <https://bund.dev/apis> zusammengefasst sind, dienen dem Zweck, den Zugang zu verschiedenen Datensätzen und Verwaltungsverfahren der Bundesverwaltung zu erleichtern. Sie ermöglichen es Entwicklern und anderen interessierten Nutzern, auf eine standardisierte und dokumentierte Art und Weise auf diese Informationen zuzugreifen und sie in eigenen Anwendungen zu nutzen. Dabei können die APIs verschiedene Funktionalitäten bereitstellen, wie beispielsweise die Abfrage von Daten, die Bearbeitung von Anträgen oder die Einreichung von Dokumenten. Durch die Bereitstellung dieser APIs im Rahmen der Open Government Umsetzungsstrategie des Bundes wird eine transparentere und effizientere Zusammenarbeit zwischen Verwaltung und Bürgern angestrebt.

3.1.2 Allgemeines über den Verzeichnisdienst

In ihrer Gesamtheit sind auf der Website insgesamt 47 diverse Web-APIs identifizierbar. Diese APIs können hauptsächlich verschiedenen Bundesbehörden zugeordnet werden. Jedoch lassen sich vereinzelt auch APIs von Landesbehörden sowie von Anstalten des öffentlichen Rechts ausmachen. Für jede der identifizierten APIs existiert eine eigene Dokumentation, welche gegebenenfalls auf GIT Hub oder eine andere Webpräsenz verweist. Innerhalb der Dokumentationen finden sich prägnante Beschreibungen und teilweise auch Verwendungsbeispiele für die jeweilige API.

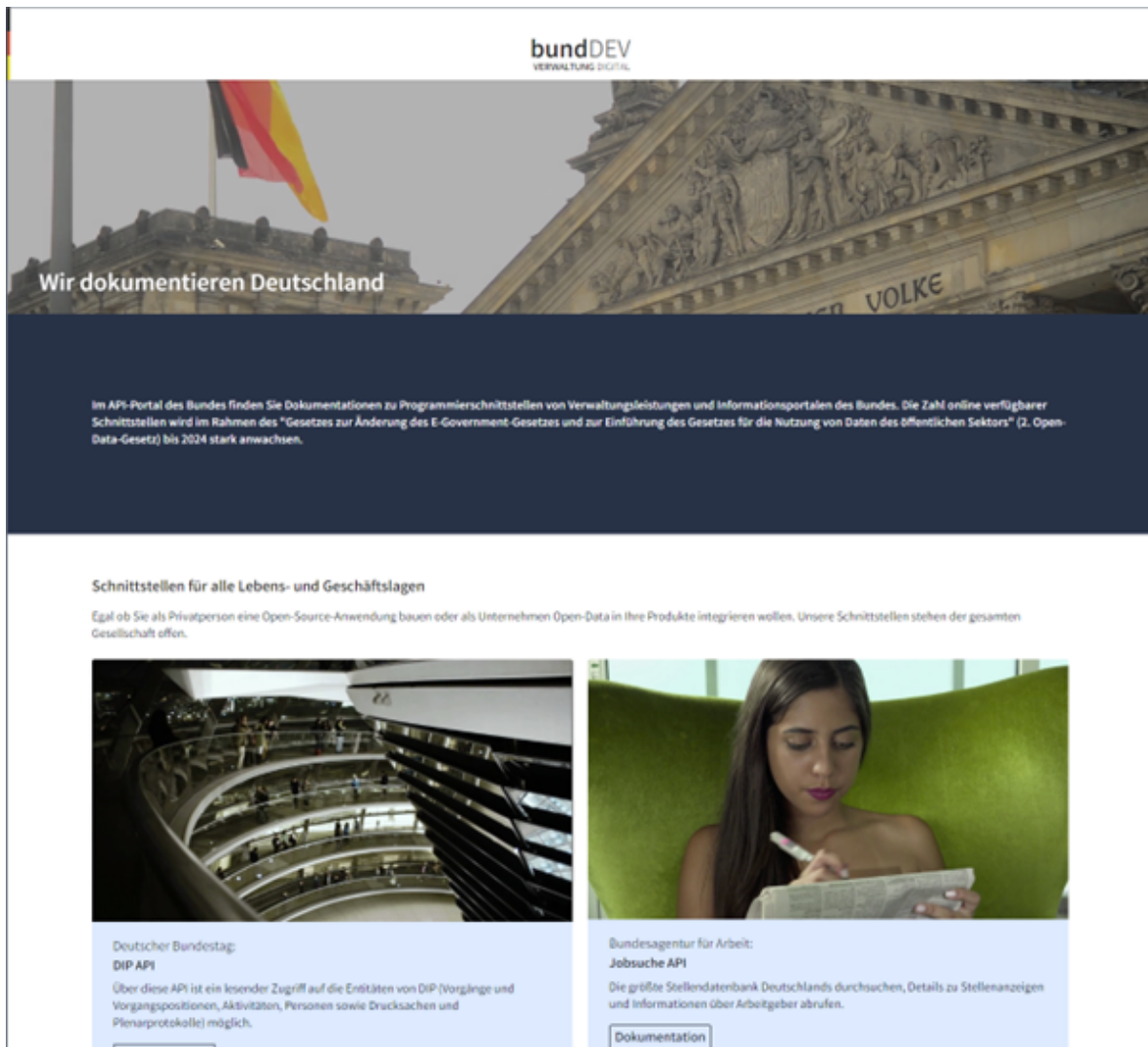


Abbildung 1: Ansicht der Startseite des Verzeichnisdienstes

In diesem Kontext ist die Integration individueller Application Programming Interfaces (APIs) nicht vorgesehen. Ebenso besteht keine Möglichkeit zur Erstellung eigener Code-Zweige sowie der Integration von Eigenprogrammierung. Lediglich die Meldung von Code-Problemen mittels GIT Hub-Funktionalität sowie das durchführen eines Pull Requestes ist gestattet. Bei einigen APIs ist es außerdem möglich, Probleme über alternative Kommunikationskanäle zu melden. Eine standardisierte Vorlage für das Reporting von Schwierigkeiten besteht nicht

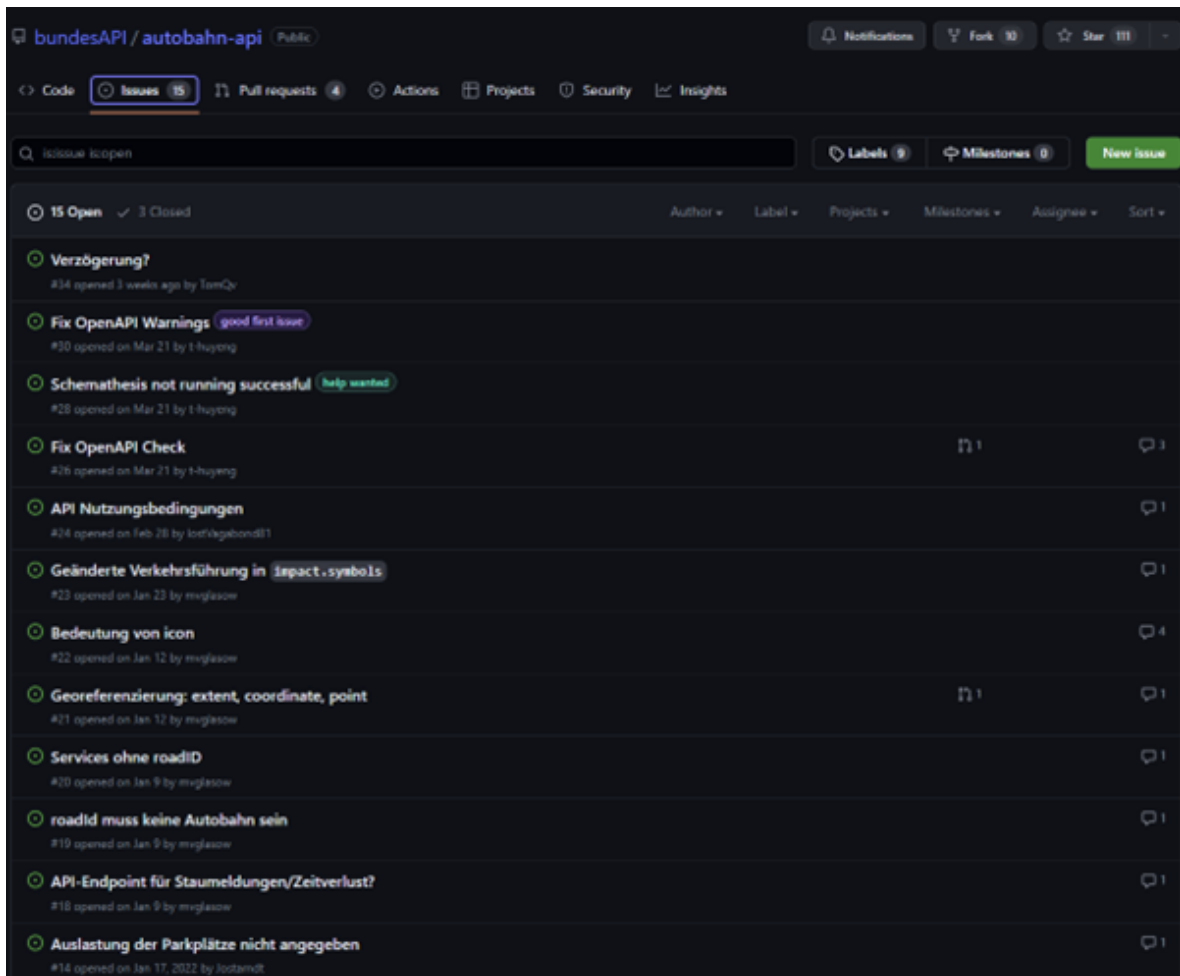


Abbildung 2: Gemeldete Probleme

Wie in Abbildung 2 ersichtlich ist, können bei der Erstellung von Problemen verschiedene Tags zugeordnet werden, um eine Klassifizierung für den Entwickler zu erleichtern.

Der Verzeichnisdienst ist ohne vorherige Anmeldung oder Registrierung zugänglich. Ebenso kann der API-Code ohne GIT-Registrierung oder Anmeldung heruntergeladen werden. Das Melden von Problemen über die GIT-Funktion erfordert jedoch eine vorherige Anmeldung und Registrierung. Analog dazu verhält es sich bei verfügbaren Pull Requests und ähnlichen Funktionalitäten.

3.1.3 Vergleich von API- und datenorientierten Schnittstellen

Beide Schnittstellen sind Softwareschnittstellen, die sich in einzelnen Punkten und Funktionalitäten unterscheiden:

Eine API (Application Programming Interface) ist eine Art von Schnittstelle, die es Anwendungen ermöglicht, mit anderen Anwendungen, Systemen oder Diensten zu interagieren. Es stellt eine Sammlung von Methoden, Funktionen und Protokollen bereit, die von einer Anwendung aufgerufen werden können, um bestimmte Aktionen auszuführen oder Daten abzurufen. Eine API dient als Abstraktionsschicht, die es einer Anwendung ermöglicht, auf die Funktionalität eines anderen Systems zuzugreifen, ohne dass sie das interne Datenmodell oder die Implementierungsdetails kennen muss. APIs sind in der Regel gut dokumentiert und stellen eine klare Schnittstelle bereit, über die Anwendungen miteinander kommunizieren können.

Eine datenorientierte Schnittstelle ist eine Art von Schnittstelle, die es Anwendungen ermöglicht, auf strukturierte oder unstrukturierte Daten zuzugreifen, sie zu manipulieren und zu speichern. Im Gegensatz zu einer API, die eine Sammlung von Methoden und Funktionen bereitstellt, um bestimmte Aktio-

nen auszuführen oder Daten abzurufen, liegt der Schwerpunkt bei einer datenorientierten Schnittstelle auf dem direkten Zugriff auf die Daten selbst. Datenorientierte Schnittstellen können beispielsweise in Form von Datenbankzugriffen oder Dateisystemzugriffen bereitgestellt werden. Im Allgemeinen erfordern sie spezielle Kenntnisse und Fähigkeiten, um sie zu nutzen, und sind oft weniger benutzerfreundlich als APIs.

Kriterium	API-Schnittstelle	Datenschnittstelle
Flexibilität	Hoch APIs bieten in der Regel mehr Flexibilität, da sie in der Lage sind, unterschiedliche Arten von Anwendungen und Plattformen zu bedienen.	Niedrig Datenorientierte Schnittstellen sind in der Regel weniger flexibel, da sie sich auf spezifische Datenmodelle und -typen konzentrieren.
Sicherheit	Höher APIs bieten in der Regel mehr Sicherheitsfunktionen wie Autorisierung, Authentifizierung und Verschlüsselung.	Niedriger Datenorientierte Schnittstellen bieten in der Regel weniger Sicherheitsfunktionen, da sie sich auf den Zugriff auf Daten konzentrieren.
Kompatibilität	Hoch APIs sind in der Regel kompatibel mit einer Vielzahl von Plattformen, Sprachen und Systemen.	Niedriger Datenorientierte Schnittstellen sind in der Regel auf bestimmte Datenquellen oder Datenbanken beschränkt.
Geschwindigkeit	In der Regel Höher APIs bieten in der Regel schnellere Verarbeitungsgeschwindigkeiten, da sie oft auf leichte und effiziente Übertragung von Daten ausgelegt sind.	In der Regel niedriger Datenorientierte Schnittstellen sind in der Regel langsamer, da sie oft auf komplexe Datenstrukturen und Datenbankabfragen zugreifen müssen.
Verwendung Granularität	Für Entwickler und Programmierer Höhere Granularität APIs bieten in der Regel eine höhere Granularität, da sie in der Lage sind, einzelne Funktionen und Dienste aufzurufen.	Für Endbenutzer Geringere Granularität Datenorientierte Schnittstellen bieten in der Regel eine geringere Granularität, da sie sich auf den Zugriff auf bestimmte Datensätze oder Datenquellen konzentrieren.

Tabelle 1: Vergleich von API-Schnittstellen und Datenschnittstellen

Es ist zu beachten, dass die Unterschiede zwischen den beiden Typen von Schnittstellen nicht immer so klar abgegrenzt sind und dass einige APIs datenorientiert sein können und einige Datenorientierte Schnittstellen APIs-ähnliche Funktionen bieten können.

3.2 Analyse von Web-APIs

3.2.1 Erstellung des Bewertungsmodells

Erstellen Sie ein Bewertungsmodell für angebotene Web APIs

- Welche Informationen halten Sie für einen Einsatz notwendig?
 - Spezifikation/Technologie (SOAP, REST, JSON, MIME, ...)

- Servicebeschreibung (technisch & fachlich)
 - Funktionstüchtigkeit (Qualitätsvereinbarungen)
 - Kontaktinformationen
 - Beispiele zur programmiertechnischen Einbindung
- Informationen zu einem Ansatz für ein Bewertungsmodell – siehe Anlage

Wir haben uns dazu entschieden alle API's aus dem BundDev Verzeichnis zu bewerten. So ist es möglich eine Übersicht der vom Bund bereitgestellten Schnittstellen zu gewinnen.

Zu Beginn wurde das Bewertungsmodell in fünf Kategorien aufgeteilt. Diese lauten: Übersicht, Offenheit, Qualität, Dokumentation und Verfügbarkeit.

1. Übersicht: In diesem Teil wird nur eine Übersicht über die einzelnen API's dargestellt. Unter welchen URL lässt sich die Dokumentation finden? Wie viele get, post, put, delete und andere werden werden in der API verwendet? Hierbei wird keine Bewertung vorgenommen, sondern es wird nur aufgezählt.
2. Offenheit:
 - Quellcode: Die Offenlegung des Quellcodes einer API ist ein wichtiger Faktor für die Offenheit einer API, da es Entwicklern ermöglicht, die API zu verstehen, anzupassen und zu erweitern. Wenn der Quellcode einer API offen zugänglich ist, können Entwickler die Funktionalität der API an ihre spezifischen Anforderungen anpassen und Fehler beheben, die in der API auftreten können. Offener Quellcode fördert auch die Zusammenarbeit und Innovation, indem er es anderen Entwicklern ermöglicht, den Code zu überprüfen und Verbesserungen und Erweiterungen vorzuschlagen. Dies kann dazu beitragen, die Qualität und Sicherheit der API zu verbessern.
 - Token/ Registrierung: Eine einfache Registrierung oder Token-Nutzung kann dazu beitragen, die API-Nutzung für die Benutzer zu vereinfachen. Anstatt komplizierte Authentifizierungsmethoden zu verwenden, können Benutzer einfach einen Token generieren oder sich registrieren, um Zugriff auf die API zu erhalten.
 - Zugänglichkeit: Eine offene API sollte für Entwickler leicht zugänglich sein. Die API sollte einfach zu finden und zu registrieren sein, und es sollte keine unnötigen Hürden geben, die die Nutzung der API einschränken.
 - Request Limit: Ein Request-Limit ist eine Einschränkung für die Anzahl der API-Anfragen, die ein Benutzer innerhalb eines bestimmten Zeitraums an den API-Server senden darf. Ein Request-Limit ist wichtig, um die API-Performance, Zuverlässigkeit und Sicherheit zu gewährleisten. Eine API kann nur eine begrenzte Anzahl von Anfragen gleichzeitig bearbeiten. Wenn zu viele Anfragen gleichzeitig eingehen, kann die API-Performance beeinträchtigt werden. Wenn zu viele Anfragen gleichzeitig eingehen und die API überlasten, kann dies zu Systemausfällen oder Verzögerungen bei der Verarbeitung von Anfragen führen. Einige Benutzer können versuchen, die API durch eine hohe Anzahl von Anfragen zu überlasten, um Schwachstellen auszunutzen oder unerwünschte Aktivitäten durchzuführen. Ein Request-Limit stellt sicher, dass alle Benutzer die gleiche Chance haben, die API zu nutzen.
 - Kontakt: Es ist wichtig, dass zu einer API ein Kontakt hinterlegt ist, damit Benutzer der API einen Ansprechpartner haben, falls sie Fragen oder Probleme haben. Wenn Benutzer auf Probleme bei der Nutzung der API stoßen, können sie sich an den hinterlegten Kontakt wenden, um Hilfe und Unterstützung zu erhalten. Ein Kontakt kann Feedback von den Benutzern der API sammeln und an den API-Anbieter weiterleiten. Dies kann dazu beitragen, die API zu verbessern, indem die Benutzeranforderungen und -bedürfnisse berücksichtigt werden. Ein Kontakt kann dazu beitragen, die Sicherheit der API zu erhöhen, indem er Benutzer über Sicherheitsbedrohungen informiert und sicherstellt, dass diese angemessen behandelt werden.
3. Qualität:

- **Granularität:** Die Granularität einer API bezieht sich auf die Anzahl und Komplexität der verfügbaren API-Endpunkte und der darin enthaltenen Funktionalität. Eine API mit zu hoher Granularität enthält möglicherweise zu viele spezialisierte Endpunkte, die für bestimmte Anwendungsfälle entwickelt wurden, während eine API mit zu niedriger Granularität möglicherweise nicht genügend Endpunkte enthält, um die erforderliche Funktionalität bereitzustellen. Es ist daher wichtig, dass die Granularität der API weder zu hoch noch zu niedrig ist.
- **TLS:** HTTPS bei einer API eingesetzt wird, um die Sicherheit und Integrität der übertragenen Daten zu gewährleisten. HTTPS ist eine Verschlüsselungsprotokoll, das auf dem HTTP-Protokoll basiert und eine sichere Verbindung zwischen einem Client und einem Server herstellt. TLS ist ein Protokoll, das die Datenverschlüsselung, Authentifizierung und Integritätssicherung während der Übertragung von Daten über ein Netzwerk ermöglicht.
- **Statuscode:** Es ist wichtig, dass alle Statuscodes für eine API angegeben sind, da sie den Client über den Zustand der Anfrage informieren und ihm ermöglichen, geeignete Maßnahmen zu ergreifen. Statuscodes sind dreistellige Zahlen, die von einem Server zurückgegeben werden, um den Erfolg oder Misserfolg einer Anfrage anzuzeigen.

4. Dokumentation:

- **Routen:** Es ist wichtig, dass alle Routen einer API in der Dokumentation erwähnt sind, um den Entwicklern zu helfen, die API vollständig zu verstehen und korrekt zu nutzen.
- **Ressourcen:** Wenn alle Ressourcen in der Dokumentation erwähnt werden, können Entwickler sicherstellen, dass sie alle verfügbaren Entitäten und Objekte der API verstehen und nutzen. Wenn eine Ressource in der Dokumentation fehlt, kann dies dazu führen, dass Entwickler fehlerhafte Anfragen senden oder unerwartete Ergebnisse erhalten.
- **Parameter:**

5. Verfügbarkeit

3.2.2 Einsatz des Bewertungsmodells

Analysieren Sie stichpunktartig 20 registrierte Web APIs

- Verwenden Sie ihr entwickeltes Bewertungsmodell
- Ausführung der Services mittels Musterlösung (keine Programmierung)

Die zu analysierenden Service-APIs sollten möglichst aus unterschiedlichen Serviceverzeichnissen stammen.

3.3 API - Spezifikationen

3.3.1 Spezifikationsanalyse

Analysieren Sie Struktur und Elemente einer WSDL, OpenAPI-(Swagger) oder auch GraphQL(Schema)-Spezifikation.

- Verwenden Sie zur Analyse 5 ausgewählte Services
- Stichpunktartige Beschreibung der Struktur/Unterelemente
- Metrische Erfassung der Struktur bzw. eingesetzten Elemente
- Statistische Auswertung Informationen (z.B. Tabellen, Diagramme)

3.3.2 Analysetools

Nutzen Sie ggf. verfügbare Hilfsmittel und gehen Sie auf die entsprechende Funktionsweise der Tools ein

- Beispiele: soapUI, SOAPSonar, Postman <https://www.postman.com>
- Grafische WSDL-Editoren (z.B. XMLSpy ab Version 8)

3.3.3 Einschränkungen und Alternativen

- Welche Informationen fehlen bei der gewählten Spezifikationen?
- Recherchieren Sie nach alternativen Beschreibungsformen?

4 Übung 3b: Entwicklung eigener Service-Angebote

4.1 Möglichkeiten für Implementierung und Deployment

4.1.1 Analyse der Möglichkeiten

Analysieren Sie mit Hilfe des Internets mögliche Alternativen zur Implementierung und Deployment von Web APIs (speziell WSDL/XML, REST/OpenAPI und GraphQL), wie z.B.:

- IDE NetBeans und GlassFish Server
- IDE Eclipse und Tomcat & Axis-Erweiterung
- Postman API Builder
- Cloud-basierte Entwicklung/Deployment

4.1.2 Analytischer Vergleich der Möglichkeiten

Vergleichen Sie die gefunden Alternativen anhand eines eigenen Bewertungsmodells, mit Hilfe von Kriterien wie z.B.:

- Voraussetzungen zur Verwendung (HW- und SW-Ressourcen)
- Integration von Entwicklung- und Ausführungsplattform
- DevOps orientierte Vorgehensweise (Automationsaspekte)
- Verbreitung, Entwicklersupport, Kosten, Lizenzen

4.2 Entwicklung

4.2.1 Rahmenbedingungen

Wählen Sie für die weiteren Aufgaben dieser Übung eine konkrete Entwicklungsumgebung aus, begründen Sie Ihre Entscheidung

- Benötigte Softwareversionen und Werkzeuge
- Installation und Konfiguration der Entwicklungsumgebung
- Cloud-basierte Implementierung und Betrieb

4.2.2 Implementierung API

Entwicklung einer Web-API (mind. 6 Operationen bzw. Datenressourcen - ggf. CRUD) und eines korrespondierenden Client

- Berücksichtigen Sie in der Doku Analyse, Design, Implementierung und Test
- Deployment (Installation) innerhalb der Laufzeitumgebung

4.2.3 Anbindung Datenbank

Originäre Verwendung eines DBMS (auch NoSQL) als Service-Schnittstelle

- Prototypisches Aufsetzen eines konkreten Datenbanksystems (ggf. Cloud)
- Details der Konfiguration und Administration – ggf. Probleme
- Eigene Kapselung mit Hilfe einer WSDL, Swagger oder GraphQL
- Performanter Umgang mit XML/JSON-basierten Datenströmen

5 Übung 4d: Sicherheit von Web APIs

5.1 Sicherheitsrisiken in Verbindung mit dem HTTP Protokoll

- Beschreiben Sie stichpunktartig die Eigenschaften von TLS (SSL) in Verbindung mit HTTPS
- Welche Verfahren zur Authentifizierung können im Rahmen des HTTP Protokolls **direkt** verwendet werden? Gehen Sie auf Stärken und Schwächen ein
- Authentifizierungs- und Benutzerinformationen werden bei Webanwendungen häufig mit Hilfe von Cookies übertragen. Gehen Sie auf die damit einhergehenden Nachteile ein.

5.2 Möglichkeiten zur Risikominderung

5.2.1 OWASP

Machen Sie sich mit den OWASP Top 10 API Security Risiken vertraut

- Gehen Sie für 5 Sicherheitslücken auf entwicklerseitige oder betriebliche Möglichkeiten zur Verminderung bzw. Abmilderung ein
- Gehen Sie für 2 Sicherheitslücken auf die Möglichkeiten von Tests zur Aufdeckung potentieller Schwachstellen ein.

5.2.2 OAuth 2 und OIDC

Analysieren Sie die grundlegende Arbeitsweise von OAuth2

- Welche grundlegenden Rollen werden in OAuth2 unterschieden?
- Welche Zusammenhänge bestehen zwischen OAuth2 und OIDC?

5.3 Praktische Anwendung von OAuth2

5.3.1 Testwerkzeuge

Verwenden Sie ein Testwerkzeug zur Nutzung von Web APIs und beschreiben Sie exakt die Schritte und Konfigurationen von mindestens 3 OAuth2 konformen Funktionsaufrufen (request), die deren erhaltenen Antworten (response)

5.3.2 Implementierung

Gehen Sie auf die Möglichkeiten einer programmiertechnischen Einbindung von OAuth2 konformen Aufrufen innerhalb der Programmiersprache Java oder ggfs. auch JavaScript ein.

- Voraussetzungen dokumentieren (genutzte APIs)
- prototypische Verwendung mit Hilfe eines lauffähigen Quellcodefragments
- Test des entwickelten Prototypen, d.h. OAuth2 Zugriff auf Web-APIs

nur Template für Codesnippet: :-)

Listing 1: Filtern von nan- und 0-values

```
1 df = df.dropna(subset=['obj_lotArea'])
2 df = df.dropna(subset=['obj_livingSpace'])
3 df = df.dropna(subset=['obj_purchasePrice'])
4 df = df.query('obj_lotArea != 0')
5 df = df.query('obj_livingSpace != 0')
6 df = df.query('obj_purchasePrice != 0')
```

Literatur

- Albora, Giambattista, Luciano Pietronero, Andrea Tacchella, and Andrea Zaccaria (2023), “Product progression: a machine learning approach to forecasting industrial upgrading.” *Scientific Reports*, 13, 1481.
- Angione, Claudio, Eric Silverman, and Elisabeth Yaneske (2022), “Using machine learning as a surrogate model for agent-based simulations.” *PloS one*, 17, e0263150.
- Athey, Susan (2018), “The impact of machine learning on economics.” In *The economics of artificial intelligence: An agenda*, 507–547, University of Chicago Press.
- Athey, Susan and Guido W Imbens (2019), “Machine learning methods that economists should know about.” *Annual Review of Economics*, 11, 685–725.
- Basheer, Mohammed, Victor Nechifor, Alvaro Calzadilla, Claudia Ringler, David Hulme, and Julien J Harou (2022), “Balancing national economic policy outcomes for sustainable development.” *Nature Communications*, 13, 5041.
- Bertoletti, Alice, Jasmina Berbegal-Mirabent, and Tommaso Agasisti (2022), “Higher education systems and regional economic development in europe: A combined approach using econometric and machine learning methods.” *Socio-Economic Planning Sciences*, 82, 101231.
- Blogger, ML (2022), “Rl in economics.” <https://medium.com/@mlblogging.k/rl-in-economics-794b43ffc995>, zuletzt abgerufen: März 2023.
- Buckmann, Marcus, Andreas Joseph, and Helena Robertson (2022), “An interpretable machine learning workflow with an application to economic forecasting.” Technical report, Bank of England.
- Giglio, Stefano, Bryan T Kelly, and Dacheng Xiu (2021), “Factor models, machine learning, and asset pricing.” *Machine Learning, and Asset Pricing (October 15, 2021)*.
- Jomthanachai, Suriyan, Wai Peng Wong, and Khai Wah Khaw (2023), “An application of machine learning to logistics performance prediction: An economics attribute-based of collective instance.” *Computational economics*, 1–52.
- Korab, Petr (2021), “Use of machine learning in economic research: What the literature tells us.” <https://towardsdatascience.com/use-of-machine-learning-in-economic-research-what-the-literature-tells-us-28b473f26043>, zuletzt abgerufen: März 2023.
- Kumbure, Mahinda Mailagaha, Christoph Lohrmann, Pasi Luukka, and Jari Porras (2022), “Machine learning techniques and data for stock market forecasting: A literature review.” *Expert Systems with Applications*, 116659.
- Merler, Silvia (2018), “Machine learning and economics.” <https://www.bruegel.org/blog-post/machine-learning-and-economics>, zuletzt abgerufen: März 2023.
- Miceli, Milagros, Julian Posada, and Tianling Yang (2022), “Studying up machine learning data: Why talk about bias when we mean power?” *Proceedings of the ACM on Human-Computer Interaction*, 6, 1–14.