	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

Konzept einer Entwicklungsumgebung

- für die Programmierung mobiler Progressive Web Applications (PWA)
sowie Native Applications (NA)
für Apple, Android, Windows 10 -

Autor

Tobias Rick

Unterstützt durch

Holger May

Technische Leitung

Christoph Schürmann

Konzeptionierung

Projektunterstützung

Projektnummer

GIS2ALCM

Projektzuweisung für ProCos

9Y200202FA


Stand

DD.MM.JJJJ

		Seite I
		Freigegeben von: TRi
		am: 23.07.2021

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	IV
1. Motivation und Ziel	1
2. Rahmenbedingungen und funktionale Anforderungen	3
3. Aufbau der Entwicklungsumgebung	5
3.1. Varianten der Entwicklungsumgebung	6
3.1.1. Lokale Entwicklungsumgebung	6
3.1.2. Server für die Entwicklungsumgebung	7
3.1.3. Optionaler Testserver	7
3.2. Voraussetzungen	8
3.2.1. Hardware	8
3.2.2. Tools/Frameworks	9
3.2.3. Funktionserweiterungen	11
3.2.4. Unit Test	11
3.3. Optionale Testumgebungen	12
3.3.1. Testserver/Testframeworks	12
3.3.2. Virtuelle Umgebung	13
3.4. Toolchain	14
3.4.1. Grafische Darstellung	15
3.4.2. Detaillierte Beschreibung	16
3.5. Applikationsplattformen	20
4. Lizenzbedingungen	21
4.1. Beschaffung	21
4.2. Freeware Lizenz vs. Kommerzielle Lizenz	23
5. Anwendungsbeispiel: Projekt-Manager als Web-App	24
5.1. Beispielhafter Programmablaufplan	25
5.2. Beispielhaftes ER-Modell der DB	26
6. Ergebnis des Konzepts	28
Glossar	V
A. Begrifflichkeiten	V
B. Linksammlung	X

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

Abkürzungsverzeichnis

A

AFA **A.Fasselt Engineering GmbH**

C

CSS **Cascading Style Sheet**

D

DB **Datenbank**

DBMS **Datenbank-Managementsystem**

E

ER-Modell **Entity-Relationship-Modell**

I

IIS **Internet Information Services**

J

JS **JavaScript**

M

MA **Mitarbeiter**

MS **Microsoft**

N

NA **Native Application**

O

OOP **Objekt orientiertes Programmieren**

OS **Operationg System = Betriebssystem**

P

PAP **Programmablaufplan**

PM **Package Manager**

PWA **Progressive Web Application**

T

TS **TypeScript**


V

VS **Microsoft Visual Studio**

W

Web-Apps **Windows Application, Web Applications sowie Mobile Applications**

		Seite III
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01


Abbildungsverzeichnis

Abbildung 1: Toolchain der Entwicklungsumgebung	15
Abbildung 2: Beispielhafter Programmablaufplan - Projekt-Manager als Web-App	25
Abbildung 3: Beispielhaftes ER-Modell – Projekt-Manager als Web-App	27

Tabellenverzeichnis

Tabelle 1: Rahmenbedingungen für die Entwicklungsumgebung für Web-Apps	3
Tabelle 2: Funktionale Anforderungen der Entwicklungsumgebung für Web-Apps	4
Tabelle 3: Hardware als Voraussetzung für die Entwicklungsumgebung von Web-App	8
Tabelle 4: Tools/Frameworks als Voraussetzung für die Entwicklungsumgebung von Web-Apps	10
Tabelle 5: Vergleich virtuelle Umgebung	13
Tabelle 6: Toolchain - Legende der grafischen Darstellung	14
Tabelle 7: Toolchain - Legende der Tabellarische Übersicht.....	14
Tabelle 8: Die Toolchain der Entwicklungsumgebung in Tabellarischer Übersicht	19
Tabelle 9: Vergleich Progressive Web Application vs. Native Application	20
Tabelle 10: Beschaffungsinformationen für die Tools/Frameworks der Entwicklungsumgebung	22
Tabelle 11: Übersicht Freeware Lizenz vs. kommerzielle Lizenz.....	23
Tabelle 12: Beispielhafte Akteure in einer DB des Projekt-Managers als Web-App.....	26

		Seite IV
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

1. Motivation und Ziel

Die A.Fasselt Engineering GmbH (AFA) plant für die Umsetzung von Windows Applications, Web Applications sowie Mobile Applications (Web-Apps) eine Entwicklungsumgebung zu entwerfen. Hierfür ist es notwendig zu planen, welche Technologien eingesetzt werden sollen. Basierend auf unten genannte Rahmenbedingungen wird die Entwicklungsumgebung eingerichtet.

Die Entwicklungsumgebung soll die Aufgabenstellung lösen, eine Technologie für die Programmierung von Web-Apps bereitzustellen. Weiterhin wird geregelt, wie das Zusammenspiel der eingesetzten Technologien abläuft.

Dazu wird ein Konzept erstellt, welches

- die Rahmenbedingen
- funktionalen Anforderungen
- Aufbau der Entwicklungsumgebung als Grundlage
- Beschaffung der Tools/Frameworks sowie der Hardware
- Entwicklung einer prototypischen Web-App

klärt.


➤ Ziel Entwicklungsumgebung für Web-Apps

Das Konzept dient als Einstiegspunkt für die Programmierung von Web-Apps in einer dazu eingerichteten Entwicklungsumgebung. Dafür beinhaltet dieses Konzept

- eine Toolchain für Erklärung der Funktionszusammenhänge zwischen
 - Tools/Frameworks
 - Hardware
- Einarbeitungsmöglichkeiten in der Linksammlung
 - Beiträge aus dem Wiki
 - Installationsanleitungen
 - Tutorials
 - In Textform
 - Als Video aufgeteilt in Lektionen

➤ Einarbeitung anwendungsfremder Software-Entwickler

		Seite 1
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

Die Entwicklungsumgebung wird für den ersten Schritt zur Entwicklung einer prototypischen mobilen App bereitgestellt. Diese mobile App wird in Zukunft das derzeitige System zur Erfassung von Projektarbeitszeiten „ProCos“ ablösen. Dabei soll diese Anwendung eine Arbeitserleichterung für folgende Bereiche der AFA sein:

- Projekt Controlling (K3)
 - Projektmanagement
 - Zuordnungen der Projektarbeitszeiten zu AFA MA
 - Erkennen von Zeitüberschreitungen
 - Indirekt Rechnungswesen
 - Für die Abrechnung von Projektarbeitszeiten
 - Zuweisung von Projektarbeitszeiten zu Projekten durch MA
 - Anknüpfen an Prozess zur Erfassung von Projektarbeitszeiten
 - Prozessnummer: K3 Projektabwicklung
 - Erfassung durch die Web-App
 - Eventuelle Anpassung des Prozesses im Prozesswiki
- Ziel Arbeitserleichterung für AFA interne Bereiche
- Proof of Concept der konzeptionierten Entwicklungsumgebung


Sollte sich die konzeptionierte Entwicklungsumgebung in der Praxis bewähren, sind weitere Einsatzgebiete geplant. Ein mögliches Einsatzgebiet wäre die Integration von mobilen Apps ins Siemens Comos. Dabei wird ein Fokus auf folgende Funktionalitätserweiterung gesetzt:

- Intelligenter Datentransfer
 - Effizienzsteigerung beim Input/Output von Daten
 - Datenerfassung sowie Datenintegration im Anlagenbau
- Ziel: Individualisierung von Siemens Comos

Es können Apps in MS Teams integriert werden, welche auf React basieren. Die Entwicklungsumgebung bietet die Möglichkeit eigene Anwendungen in MS Teams zu implementieren.

- Ziel: Individualisierung von MS Teams

		Seite 2
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

2. Rahmenbedingungen und funktionale Anforderungen

Die Rahmenbedingen enthalten vorher festgelegten Bedingungen, welche bei der Entwicklung eines Konzeptes berücksichtigt werden. Für die Konzeptionierung der Entwicklungsumgebung gelten festgelegte Bedingungen. Die nachfolgende Ansicht ist eine übersichtliche Auflistung der Rahmenbedingungen.

Zwecke	<ul style="list-style-type: none"> • Programmierung von <ul style="list-style-type: none"> ○ Windows Applications ○ Web Applications ○ Mobile Applications ➤ Web-Apps
Stand des Ausgangspunktes	<ul style="list-style-type: none"> • Funktionierende Toolchain • Hardware erfüllt Bedingungen für <ul style="list-style-type: none"> ○ Entwicklung + Test von <ul style="list-style-type: none"> ▪ Windows Applications ▪ Web Applications ▪ Mobile Applications ➤ Web-Apps
Bereitstellung Server	<ul style="list-style-type: none"> • Entwicklungsumgebung • Testumgebung (Optional)
Windows Web Hosting Environment	<ul style="list-style-type: none"> • MS Internet Information Service (IIS)
Package Manager	<ul style="list-style-type: none"> • NPM für JS • NUGET für C#
Tools/Frameworks	<ul style="list-style-type: none"> • IDE <ul style="list-style-type: none"> ○ Visual Studio 2019 (Community) ○ Bzw. Visual Studio 2022 (Professional) • Web API (Web Application Programming Interface) <ul style="list-style-type: none"> ○ ASP.NET ○ Postman (Testframework) • Java <ul style="list-style-type: none"> ○ JavaScript (JS) + TypeScript (TS) ○ React + React Native ○ Node.js ○ CSS ○ HTML • C# <ul style="list-style-type: none"> ○ Net.Core 5.0 ○ ASP.NET
Testframeworks	<ul style="list-style-type: none"> • JavaScript • C# • Web API
Datenbanken	<ul style="list-style-type: none"> • Integriert in den Anwendungen
Versionskontrolle	<ul style="list-style-type: none"> • GIT

Tabelle 1: Rahmenbedingungen für die Entwicklungsumgebung für Web-Apps

		Seite 3
	Freigegeben	von: TRi
		am: 23.07.2021

Für den laufenden Betrieb der Entwicklungsumgebung sind funktionale Anforderungen vorgegeben. Die Erfüllung der nachfolgend genannten funktionalen Anforderungen muss regelmäßig überprüft werden:

Anforderungsanalyse	<ul style="list-style-type: none"> Entwicklung Web-App <ul style="list-style-type: none"> Prüfen ob ausreichender Funktionsumfang vorhanden Gegebenenfalls Erweiterung des Funktionsumfangs mit Tools/Frameworks + Dokumentation
Programmierungsumgebung	<ul style="list-style-type: none"> IDE Gegebenenfalls Verwendung mehrerer IDEs
Versionskontrolle	<ul style="list-style-type: none"> Backups von früheren Programmversionen Unterstützung bei in Konflikt stehenden Programmcode <ul style="list-style-type: none"> Geteilte Ansicht beider Dateiversionen Markieren der Unterschiede im Programmcode Editieren des Programmcodes
Integrationstests	<ul style="list-style-type: none"> Updates der Tools/Frameworks muss Funktionsintegrität der programmierten Web-Apps gewährleisten <ul style="list-style-type: none"> Abhängigkeiten müssen funktionieren Dummy Funktionen zum Testen der geupdateten Technologien Unit Tests Einsatzfähigkeit der App im Zielsystem nach Technologieupdates gewährleisten
Testumgebung	<ul style="list-style-type: none"> Testframeworks Testserver (Optional) Virtuelle Umgebung (Optional)
Akzeptanztest/Anwendertest	<ul style="list-style-type: none"> Test durch zukünftige Anwender <ul style="list-style-type: none"> Anhand eines Handbuches Nehmen die Anwender die Web-App an? Ist das Ergebnis für die Anwender wie erwartet?
Dokumentation	<ul style="list-style-type: none"> Versionsnutzung Tools/Frameworks Installationsanweisungen Funktionsbeschreibung

Tabelle 2: Funktionale Anforderungen der Entwicklungsumgebung für Web-Apps

3. Aufbau der Entwicklungsumgebung

Dieser Abschnitt klärt grundlegende Aspekte, um die Bereitstellung einer Entwicklungsumgebung für Web-Apps zu ermöglichen. Die Entwicklungsumgebung baut sich, grob zusammengefasst, wie folgt chronologisch auf:

- Lokale Entwicklungsumgebung
 - Testen der konzeptionierten Entwicklungsumgebung auf lokales Notebook
 - Anfertigen von Anleitungen sowie Dokumentationen
- Server für die Entwicklungsumgebung
 - Erfüllung der Anforderungen zur Entwicklung von Web-Apps
 - Bereitstellen von Hardware sowie Tools/Frameworks
- Testserver
 - Testumgebung für die entwickelten Web-Apps
 - Bereitstellen von Hardware sowie Tools/Testframeworks
 - Die Bereitstellung des Testservers ist optional

Für die Programmierung der Funktionslogik der Web-Apps werden Hardware sowie Tools/Frameworks benötigt, welche funktional zusammenhängen. Für die Bereitstellung der Entwicklungsumgebung gelten folgende Voraussetzungen:

- Verfügbarkeit von Hardware
- Einsatz von Tools/Frameworks
- Möglichkeit der Funktionserweiterung der eingesetzten Tools/Frameworks

Die Entwicklungsumgebung kann optional durch folgende Testumgebungen erweitert werden:

- Debugging mittels Testserver sowie Testframeworks
- Virtuelle Umgebungen als Testsysteme


Die Funktionszusammenhänge der Hardware sowie der Tools/Frameworks werden in einer Toolchain wie folgt definiert:

- Grafische Darstellung mit Aufgabenbeschreibung
- Tabellarische Übersicht aller Funktionszusammenhänge

Für die in der Entwicklungsumgebung programmierten Web-Apps muss die Kompatibilität für Ziel-Plattformen gewährleistet sein. Dabei ist zu unterscheiden in:

- Progressive Web Application
- Native Application

Das Ziel dieses Abschnittes ist es alle Aspekte für die Realisierung einer Entwicklungsumgebung von Web-Apps zu klären.

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

3.1. Varianten der Entwicklungsumgebung

Die Entwicklungsumgebung für Web-Apps wird sich aus zwei Varianten zusammensetzen, welche optional um einen Testserver erweitert werden kann. Chronologisch werden die verschiedenen Varianten wie folgt implementiert:

1. Lokale Entwicklungsumgebung
2. Server für die Entwicklungsumgebung
3. Testserver

Die nächste Variante wird jeweils nach erfolgreichem Abschluss der vorherigen Variante eingerichtet.

3.1.1. Lokale Entwicklungsumgebung

Die konzeptionierte Entwicklungsumgebung soll im ersten Schritt auf einem lokalen Laptop umgesetzt werden. Begleitend werden

- Anleitungen
 - Installation
 - Verknüpfen der Tools/Frameworks
- Dokumentationen


erstellt.

Die lokale Entwicklungsumgebung wird

- das Testen der Tools/Frameworks
- die Programmierung von prototypischen Web-Apps
- einen Zugriff auf die Unternehmensdomäne
- das Testen von Anwendungsfällen

ermöglichen.

		Seite 6
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

3.1.2. Server für die Entwicklungsumgebung

Anhand der Ergebnisse aus der lokalen Testumgebung wird ein Server für die Entwicklungsumgebung bereitgestellt. Bei der Einrichtung der Entwicklungsumgebung werden die Anleitungen sowie Dokumentationen, welche begleitend bei der Einrichtung der lokalen Entwicklungsumgebung erstellt wurden, berücksichtigt.

Die Entwicklungsumgebung wird auf einem Server im Citrix der AFA laufen. Mit Realisierung des Servers wird

- Programmierung von Web-Apps
- Zugriff auf die Unternehmensdomäne der AFA
- Parallele Nutzung durch mehrere Software-Entwickler
 - Temporäre Entwicklungsumgebung pro angemeldeten Nutzer
- Testen von Anwendungsfällen

möglich sein.

3.1.3. Optionaler Testserver

Um das Testen von Anwendungsfällen effektiver zu stellen, kann für diesen Zweck ein Testserver optional eingerichtet werden. Dieser Server stellt dann eine Umgebung mit Tools/Testframeworks bereit.

Ein Testserver bietet die Möglichkeit das Testen von Anwendungsfällen zu automatisieren.

- Ziel: Arbeitserleichterung für die Software-Entwickler der AFA.

		Seite 7
		Freigegeben von: TRi
		am: 23.07.2021

3.2. Voraussetzungen

In diesem Unterabschnitt werden die Voraussetzungen geklärt, welche geschaffen werden müssen, um eine Entwicklungsumgebung für die Programmierung von Web-Apps bereitzustellen. Folgende Voraussetzungen müssen erfüllt werden:

- Bereitstellung
 - Hardware
 - Tools/Frameworks
- Möglichkeit zur Funktionserweiterung der Tools/Frameworks durch Bibliotheken

3.2.1. Hardware

Für die Realisierung der Entwicklungsumgebung von Web-Apps wird folgende Hardware benötigt:

Mobile Testumgebungen	Apple <ul style="list-style-type: none"> • iPhone oder iPad <ul style="list-style-type: none"> ○ Version: Ausreichend, um Anwendungszweck zu ermöglichen ○ OS Version: Ausreichend, um Anwendungszweck zu ermöglichen
	Android <ul style="list-style-type: none"> • Smartphone oder Tablet <ul style="list-style-type: none"> ○ Hersteller: Beliebiger mit Android OS ○ Version: Ausreichend, um Anwendungszweck zu ermöglichen ○ OS Version: Ausreichend, um Anwendungszweck zu ermöglichen
	Windows 10 <ul style="list-style-type: none"> • Tablet <ul style="list-style-type: none"> ○ Version: Ausreichend, um Anwendungszweck zu ermöglichen ○ OS Version: Windows 10
Lokale Testumgebung	Desktop/Notebook <ul style="list-style-type: none"> • OS Version: Windows 10
Server	Lokale Entwicklungsumgebung <ul style="list-style-type: none"> • Desktop/Notebook • Windows 10 lauffähiges Notebook
	Entwicklungsumgebung <ul style="list-style-type: none"> • Windows 10 lauffähiger Server • Im Citrix der AFA
	Optionalen Testserver <ul style="list-style-type: none"> • Windows 10 lauffähiger Server • Im Citrix der AFA

Tabelle 3: Hardware als Voraussetzung für die Entwicklungsumgebung von Web-App


3.2.2. Tools/Frameworks

Für die Realisierung der Entwicklungsumgebung zur Programmierung von Web-Apps werden die nachfolgend genannten Tools/Frameworks benötigt.

Tools	IDE <ul style="list-style-type: none"> Visual Studio 2019 (Community) <ul style="list-style-type: none"> Freie Lizenz Temporäre Nutzung bis Umstieg auf Version 2022 Professional Visual Studio 2022 (Professional) <ul style="list-style-type: none"> Erwerb der Lizenz nach Veröffentlichung Löst Version 2019 Community ab Visual Studio Mobile Development Plugin <ul style="list-style-type: none"> .Net Basis von Xamarin Visual Studio Package Manager <ul style="list-style-type: none"> NPM NUGET
	Package Manager <ul style="list-style-type: none"> NPM <ul style="list-style-type: none"> Installations-Kit für <ul style="list-style-type: none"> JS + TS React + React Native Node.js CSS + HTML Library extensions für JS + TS, React + React Native Basis Visual Studio NUGET <ul style="list-style-type: none"> Library extensions für C# Basis Visual Studio
	Windows Web Hosting Environment <ul style="list-style-type: none"> MS Internet Information Service (MS IIS)
	Datenbanken <ul style="list-style-type: none"> SQL + NOSQL <ul style="list-style-type: none"> Relational + Nicht relational Mögliche Effizienzsteigerung, wenn nicht relational, da „Joins“ vermieden werden können DB für erste prototypische App <ul style="list-style-type: none"> Bereitstellung einer öffentlichen API
	Versionskontrolle <ul style="list-style-type: none"> GIT <ul style="list-style-type: none"> Backups von früheren Programmversionen Unterstützung bei in Konflikt stehenden Programmcode

Frameworks	Java <ul style="list-style-type: none"> JavaScript (JS) <ul style="list-style-type: none"> Zugrunde liegende Scriptsprache für Web-Apps TypeScript (TS) <ul style="list-style-type: none"> Objektorientierter JS Interpreter React <ul style="list-style-type: none"> JS/TS Library UI Programmierung Web-Apps React Native <ul style="list-style-type: none"> Building App für React UI Programmierung Mobile Apps Node.js <ul style="list-style-type: none"> JS dependency Für JS Anwendungen/Server und Kommunikation mit Server CSS HTML
	C# <ul style="list-style-type: none"> Net.Core 5.0 <ul style="list-style-type: none"> ASP.NET <ul style="list-style-type: none"> Web API
	Web API <ul style="list-style-type: none"> ASP.NET <ul style="list-style-type: none"> Kommunikation Server <-> Client über http In C# Integriertes Account-System HTTP request HTTP response Data transfer <ul style="list-style-type: none"> via XML, JSON, other files Postman <ul style="list-style-type: none"> Debugging der Web API Derzeitige Freeware Nutzung

Tabelle 4: Tools/Frameworks als Voraussetzung für die Entwicklungsumgebung von Web-Apps

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

3.2.3. Funktionserweiterungen

Die Entwicklungsumgebung wird für Programmierung unterschiedlicher Web-Apps bereitgestellt. Es besteht die Möglichkeit den Funktionsumfang der Tools/Frameworks durch Bibliotheken zu erweitern. Eine Anforderung zur Erweiterung des Funktionsumfangs kann sich beim Entwurf sowie bei der Entwicklung einer Web-App ergeben. Für folgende Tools/Frameworks gibt es Funktionserweiternde Bibliotheken.

- JavaScript + TypeScript
 - NPM bei Visual Studio
- React + React Native
 - NPM bei Visual Studio
- C#
 - NUGET bei Visual Studio

3.2.4. Unit Test

Bei einem Unit Test werden Komponenten einer Software getestet. Der Vorteil beim Unit Test ist, dass der gesamte Programmcode für die Komponente einbezogen werden.

Wird der Programmcode einer Klasse geändert, kann das Auswirkungen auf andere Klassen im Programmcode haben. Je komplexer das Programm wird, desto unübersichtlicher werden die Zusammenhänge der Klassen.


In Visual Studio ist die Möglichkeit zum Unit Test integriert. Dafür werden in Visual Studio Scripts für eine Testdurchführung geschrieben. Ein Ampelsystem gibt die Art des Ergebnisses wieder. Die Ampelfarben haben folgende Bedeutungen:

- Grün für einen erfolgreichen Test
- Gelb für einen erfolgreichen Test, welcher aber Warnungen enthält
- Rot für eine fehlerhafte Durchführung eines Tests

Nachfolgend werden die Möglichkeiten zum Testen genannt.

- JavaScript
 - Unit testing in Visual Studio
- C#
 - Unit testing in Visual Studio
- Web API
 - Postman

		Seite 11
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

3.3. Optionale Testumgebungen

Die Entwicklungsumgebung kann optional durch die Integration von Testumgebungen erweitert werden. Diese Erweiterungen können implementiert werden, um die Entwicklung von Web-Apps zu erleichtern. Der Einsatz folgender optionaler Testumgebungen wäre dabei denkbar:

- Testserver/Testframeworks
- Virtuelle Umgebung

3.3.1. Testserver/Testframeworks

Ein optional eingesetzter Testserver mit samt Tools/Testframeworks ermöglicht das automatisierte Testen von Anwendungsfällen. Der Ablauf ist wie folgt:

- Liefert Testergebnisse
- Löscht automatisch temporäre Testumgebung nach Abschluss der erfolgreichen Testfälle
 - Bei Misserfolg Testumgebung bleibt erhalten
- Automatischer Prozess kann bei einem Commit ausgelöst werden
 - Commit kann bei Fehler abgelehnt werden
- Continuous Delivery Pipeline für Testautomatisierung

Für die Automatisierung besteht die nachfolgende Möglichkeit.

- Optionaler Einsatz von Jenkins
 - Prüfen der Machbarkeit
- Arbeitserleichterung für Software-Entwickler der AFA

Folgende Tools/Testframeworks werden für die Testumgebung verwendet:

- JavaScript
 - Unit testing in Visual Studio
- C#
 - Unit testing in Visual Studio
- Web API
 - Postman

		Seite 12
	Freigegeben	von: TRi
		am: 23.07.2021

3.3.2. Virtuelle Umgebung

Virtuelle Umgebungen schaffen Testumgebungen für die prototypischen Web-Apps. Dort können Testergebnisse erzielt werden, ohne die eigentliche Entwicklungsumgebung zu beeinträchtigen. Der Einsatz von virtuellen Umgebungen bietet eine optionale Erweiterung der Testumgebung für die Entwicklung von Web-Apps.

Bei der Durchführung eines bestimmten Testfalles existiert eine temporär eingerichtete Testumgebung. Diese wird automatisch für einen bestimmten zu testenden Anwendungsfall eingerichtet. Die temporäre Testumgebung wird nach erfolgreichem Test gelöscht und stellt als Ergebnis Logs der Testdurchführung bereit. Bei einem Fehlschlag bleibt die temporäre Testumgebung bestehen. Weiterhin werden Logs zur Durchführung des Tests erstellt.

Die nachfolgende Tabelle zeigt zwei Beispiele Möglichkeiten zum Einsatz von virtuellen Umgebungen auf.

VMware	Virtual Box
<ul style="list-style-type: none"> • Von VMware • Lizenzpflichtig • Windows 10 OS Lizenzen <ul style="list-style-type: none"> ○ Windows 10 Evaluation Image als freie Lizenz, fährt nach einer Stunde herunter ○ Windows 10 Professional Image als kostenpflichtige Lizenz • Deployment <ul style="list-style-type: none"> ○ Durch Nutzung von Templates ○ Beinhalten Tools/Frameworks <p>➤ Testumgebung</p>	<ul style="list-style-type: none"> • Von Oracle • Freeware • Windows 10 OS Lizenzen <ul style="list-style-type: none"> ○ Windows 10 Evaluation Image als freie Lizenz, fährt nach einer Stunde herunter ○ Windows 10 Professional Image als kostenpflichtige Lizenz • Deployment <ul style="list-style-type: none"> ○ Durch Nutzung von Templates ○ Beinhalten Tools/Frameworks <p>➤ Testumgebung</p>

Tabelle 5: Vergleich virtuelle Umgebung

3.4. Toolchain

Die Toolchain beschreibt den Funktionszusammenhang der Tools/Frameworks sowie der Hardware der Entwicklungsumgebung. Bei der Entwicklung einer Web-App wird eine Funktionslogik für den **Server (Backend)** sowie den **Client (Frontend)** programmiert. Diese Funktionslogik wird auf der Hardware eines Servers sowie der Hardware eines Clients ausgeführt. Dabei werden die Funktionszusammenhänge folgendermaßen erklärt:

Grafische Darstellung Toolchain mit folgender Legende	
Funktion	Beschreibung
Aufteilung der Grafik	<ul style="list-style-type: none"> Aufteilung in orange Blöcke nach <ul style="list-style-type: none"> Ebene Tool Ressource Aufgabe
Ebene	<ul style="list-style-type: none"> Darstellung der Funktionslogik für den Server (Backend) sowie den Client (Frontend) in zwei Ebenen <ul style="list-style-type: none"> Elemente des Servers in Grüntönen Elemente des Clients in Blautönen
Tool/Framework + Hardware	<ul style="list-style-type: none"> Darstellung der Tools/Frameworks + Hardware <ul style="list-style-type: none"> Tools/Frameworks des Servers in Grün Hardware des Servers in Dunkelgrün Tools/Frameworks des Clients in Blau Hardware sowie Browser des Clients in Dunkelblau
Ressource	<ul style="list-style-type: none"> Verwendung von Programmiersprachen als Ressource der Tools/Frameworks <ul style="list-style-type: none"> Serverseitige Programmiersprachen in Grün Clientseitige Programmiersprachen in Blau
Aufgabe	<ul style="list-style-type: none"> Enthält Aufgabenbeschreibung zu den Tools/Frameworks <ul style="list-style-type: none"> Aufgaben in der Ebene des Servers in Grün Aufgaben in der Ebene des Clients in Blau
Linienarten	<ul style="list-style-type: none"> Erklärung des Funktionszusammenhanges in unterschiedlichen Arten der Elementverbindungen <ul style="list-style-type: none"> Komponentenzusammenhang in gestrichelter Linie Funktionszusammenhang in Linie mit normaler Helligkeit Funktionskomponenten eines Package Managers in heller Linie Zusammenhang der Client-Anwendung zur Schnittstelle zum Server in dunkler Linie

Tabelle 6: Toolchain - Legende der grafischen Darstellung

Tabellarische Übersicht der Funktionszusammenhänge mit folgender Legende
<ul style="list-style-type: none"> Beschreibung der Funktionszusammenhänge in einer tabellarischen Übersicht <ol style="list-style-type: none"> Beginnend bei der Hardware des Servers Tools/Frameworks und Ressourcen des Servers Tools/Frameworks und Ressourcen des Clients Endend bei der Hardware sowie des Browsers beim Client

Tabelle 7: Toolchain - Legende der Tabellarische Übersicht

3.4.1. Grafische Darstellung

In der folgenden Grafik wird die Toolchain für die Entwicklungsumgebung von Web-Apps dargestellt.

Jede Ebene setzt sich aus Tools/Frameworks sowie Hardware zusammen. Die Tools/Frameworks nutzen Programmiersprachen als Ressource. Zu jedem Tool/Framework ist eine Aufgabenbeschreibung enthalten. Die Grafik ist beginnend mit der Ebene des Servers und fortlaufend mit der Ebene des Clients zu lesen.

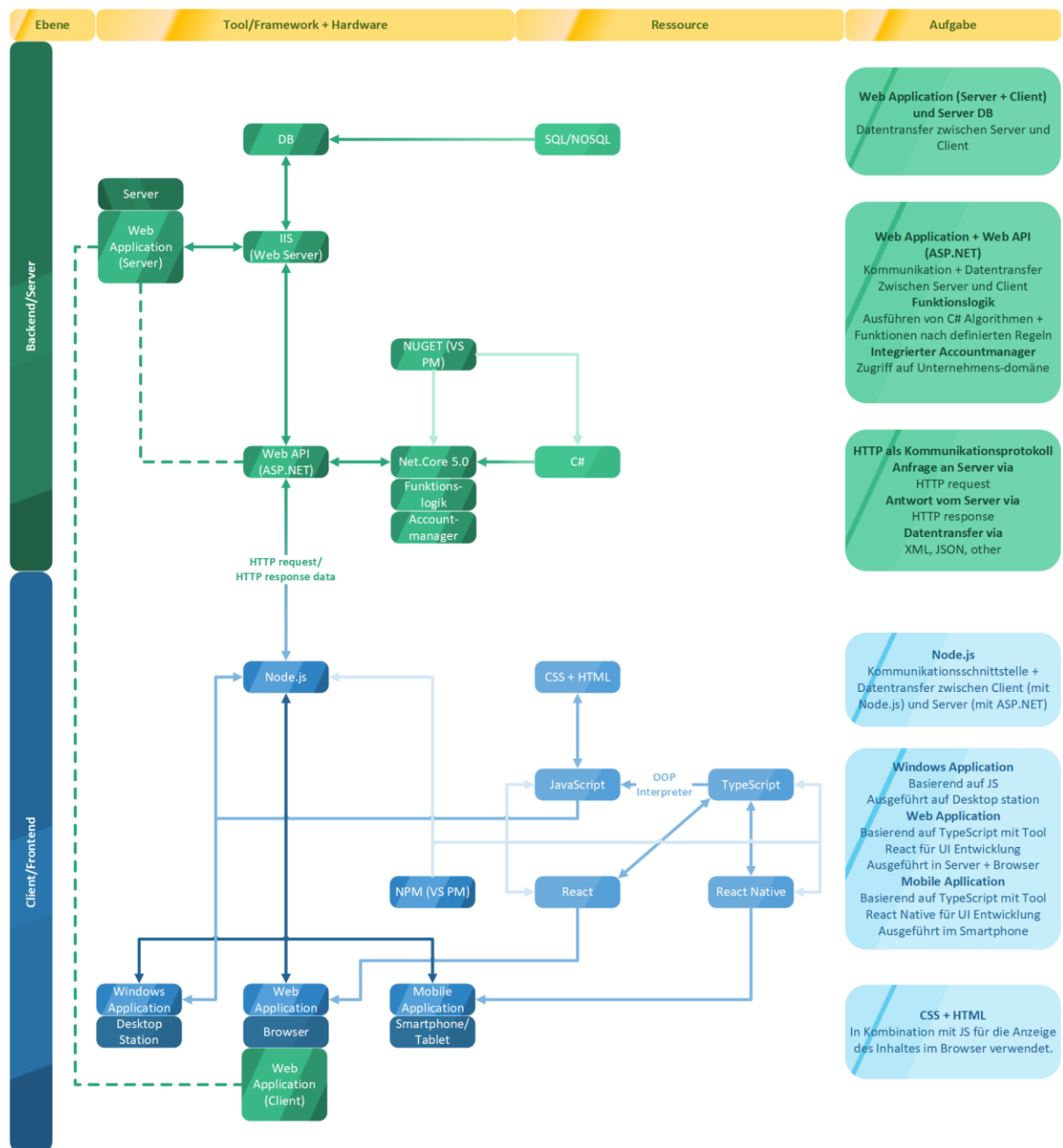


Abbildung 1: Toolchain der Entwicklungsumgebung

3.4.2. Detaillierte Beschreibung

Zwischen der Hardware sowie den Tools/Frameworks bestehen Funktionszusammenhänge wie in der grafischen Toolchain dargestellt. Die Funktionszusammenhänge werden in der nachfolgenden Tabelle in der folgenden Reihenfolge definiert:

1. Beginnend bei der Hardware des Servers
2. Tools/Frameworks und Ressourcen des Servers
3. Tools/Frameworks und Ressourcen des Clients
4. Endend bei der Hardware sowie des Browsers beim Client

Ebene	Startkomponente	Zielkomponente	Beschreibung
Server (Backend)	Server (Hardware)	Web Application (Server)	<ul style="list-style-type: none"> • Stellt dem Server Funktionslogiken bereit <ul style="list-style-type: none"> ➤ Kommunikation + Datenübertragung mit den Clients
	Web Application (Server)	Web Application (Client)	<ul style="list-style-type: none"> • Zusammenhänge Web Application zwischen Server und Client • Bereitstellung einer Clientseitigen Web Application im Browser <ul style="list-style-type: none"> ○ Funktionslogiken für clientseitige Aufgabe ○ Schutz vor Infiltration des Servers muss gewährleistet sein ○ Verhindern des Missbrauches der Funktionslogik
	Web Application (Server)	Web API (ASP.NET)	<ul style="list-style-type: none"> • Web API als integrierte Komponente der Web Application (Server) <ul style="list-style-type: none"> ○ Bereitstellung klar definierter Funktionen als Interface
	DB	SQL/NOSQL	<ul style="list-style-type: none"> • Programmierung der DB Bedienungslogik durch <ul style="list-style-type: none"> ○ SQL <ul style="list-style-type: none"> ▪ Für relationale DBs ○ NOSQL <ul style="list-style-type: none"> ▪ Für nicht relationale + relationale DBs

Server (Backend)	IIS	DB	<ul style="list-style-type: none"> • Stellt Laufzeitumgebung für Server DB bereit • Ermöglicht Ausführung der DB Bedienungslogik durch die Web Application
	IIS	Web Application (Server)	<ul style="list-style-type: none"> • Laufzeitumgebung für die serverseitige Web Application • Stellt Kommunikation serverseitig zwischen Web Application (Server) über Web API und Client-Anwendung über Node.js sicher
	IIS	Web API (ASP.NET)	<ul style="list-style-type: none"> • Ausführen der Funktionslogik des ASP.NET zur Laufzeit • Auf Basis von <ul style="list-style-type: none"> ○ ASP ○ .NET
	Net.Core 5.0 + Funktionslogik + Accountmanager		<ul style="list-style-type: none"> • Bibliotheksammlung zur Programmierung einer Web API aus Basis ASP.NET <ul style="list-style-type: none"> ○ Paket mit Funktionslogiken <ul style="list-style-type: none"> ▪ C# ○ Integrierter Accountmanager mit Zugriff auf Unternehmensdomäne <ul style="list-style-type: none"> ▪ Verwendung bestehender Accountinformationen der AFA
	NUGET (VS PM)	Net.Core 5.0 + Funktionslogik + Accountmanager	<ul style="list-style-type: none"> • Package Manager integriert in Visual Studio für <ul style="list-style-type: none"> ○ Net.Core 5.0 ○ C#
		C#	
	Net.Core 5.0 + Funktionslogik + Accountmanager	C#	<ul style="list-style-type: none"> • Bibliotheksammlung für C# <ul style="list-style-type: none"> ○ Auf Basis von Net.Core 5.0
	Web API (ASP.NET)	Net.Core 5.0 + Funktionslogik + Accountmanager	<ul style="list-style-type: none"> • Framework für die Programmierung der Web API auf Basis von ASP.NET

Server	Client	Web API (ASP.NET)	Node.js	<ul style="list-style-type: none"> Schnittstelle zwischen Client und Server Kommunikationsprotokoll via <ul style="list-style-type: none"> HTTP HTTPS Datenübertragung als <ul style="list-style-type: none"> XML JSON Other file types
Client (Frontend)		Node.js	JavaScript	<ul style="list-style-type: none"> Clientseitige Laufzeitumgebung <ul style="list-style-type: none"> Auf Basis von JS
		Node.js	Windows Application	<ul style="list-style-type: none"> Laufzeitumgebung für die clientseitige Anwendung Stellt Kommunikation clientseitig zwischen Client-Anwendung über Node.js und Web Application (Server) über Web API sicher Kommunikationsprotokoll via <ul style="list-style-type: none"> HTTP HTTPS Datenübertragung als <ul style="list-style-type: none"> XML JSON Other file types
			Web Application	
			Mobile Application	
		NPM (VS PM)	Node.js	<ul style="list-style-type: none"> Package Manager integriert in Visual Studio für <ul style="list-style-type: none"> Node.js JS + TS React + React Native
			JavaScript	
			TypeScript	
			React	
			React Native	
		JavaScript	CSS + HTML	<ul style="list-style-type: none"> Erweiterung von JS basierten Dokumenten durch Festlegung von Stylekonfigurationen Mittels Stylesheet-Sprache CSS Festlegung von <ul style="list-style-type: none"> Rahmendicke Schriftgröße Stil der Überschriften Etc... HTML als Grundlage für die Anzeige des Inhaltes im Webbrowser
		TypeScript	JavaScript	<ul style="list-style-type: none"> OOP Interpreter von JS Erweiterung der Funktionalität von JS <ul style="list-style-type: none"> Programmieren in JS + TS möglich Der Funktionsumfang von JS ist eine Teilmenge des Funktionsumfangs von TS

Client (Frontend)	TypeScript	React	<ul style="list-style-type: none"> TS Bibliothek zur UI Programmierung von Web Applications
	TypeScript	React Native	<ul style="list-style-type: none"> TS Bibliothek zur UI Programmierung von Mobile Applications
	Windows Application	JavaScript	<ul style="list-style-type: none"> Programmierung von Windows Applications <ul style="list-style-type: none"> Auf Basis von JS
	Web Application	TypeScript	<ul style="list-style-type: none"> Programmierung von Web Applications <ul style="list-style-type: none"> Auf Basis von TS Mit TS Bibliothek React zur UI Programmierung
		React	
	Mobile Application	TypeScript	<ul style="list-style-type: none"> Programmierung von Mobile Applications <ul style="list-style-type: none"> Auf Basis von TS Mit TS Bibliothek React zur UI Programmierung <ul style="list-style-type: none"> Inklusive Responsive Design
		React Native	
	Desktop Station	Windows Application	<ul style="list-style-type: none"> Ausführen der Windows Application <ul style="list-style-type: none"> Auf einer Desktop Station
	Browser	Web Application	<ul style="list-style-type: none"> Ausführen der Web Application <ul style="list-style-type: none"> Im Browser
	Smartphone/ Tablet	Mobile Application	<ul style="list-style-type: none"> Ausführen der Mobile Application <ul style="list-style-type: none"> Auf dem Smartphone/Tablet

Tabelle 8: Die Toolchain der Entwicklungsumgebung in Tabellarischer Übersicht

3.5. Applikationsplattformen

Die Web-Apps werden auf Basis von zwei unterschiedlichen Applikationsplattformen entwickelt. Nachfolgend werden die Applikationsplattformen beschrieben.

Progressive Web Application (PWA)	Native Application (NA)
<ul style="list-style-type: none"> • Anwendung auf stationäre sowie mobile Geräte ausführbar • Ersetzt die doppelte Entwicklung <ul style="list-style-type: none"> ○ der browserbasierten Anwendung ○ der mobilen Anwendung • Aufruf über URL <ul style="list-style-type: none"> ○ Integrierte Offline-Funktionalitäten ○ Caching von zu verarbeitenden Daten im Offline-Betrieb • Hybrid aus responsiver Website und App <ul style="list-style-type: none"> ○ Besitzt Merkmale einer Native Application • Im Offline-Betrieb Caching von zu verarbeitenden Daten • Programmierung mit Sprachelemente <ul style="list-style-type: none"> ○ HTML5 ○ CSS3 ○ JS • Entwicklung für folgende OS möglich <ul style="list-style-type: none"> ○ Apple ○ Android ○ Windows 10 	<ul style="list-style-type: none"> • Ist eine Mobile Application • Bietet umfangreiche Funktionalitäten <ul style="list-style-type: none"> ○ Implementierung von einfachen Anwendungen ○ Bis hin zu komplexen Anwendungen • Beschaffung der App über einen entsprechenden App Store • Remote-Control möglich <ul style="list-style-type: none"> ○ Via Remote-Apps • Entwicklung für folgende OS möglich <ul style="list-style-type: none"> ○ Apple ○ Android ○ Windows 10

Tabelle 9: Vergleich Progressive Web Application vs. Native Application

4. Lizenzbedingungen


Die A.Fasselt Engineering GmbH plant die Umsetzung einer Entwicklungsumgebung für die Realisierung von Web-Apps. Die Umsetzung erfordert die Beschaffung von Hardware sowie Tools/Frameworks.

Für die Beschaffung der Hardware sowie den Tools/Frameworks sind Kosten zu berücksichtigen. Weiterhin sind die Lizenzbedingungen der Tools/Frameworks zu beachten. Es ist geltendes Recht zu befolgen.

4.1. Beschaffung

Bezüglich Hardware kann davon ausgegangen werden, dass bereits alles Notwendige vorhanden ist. Die Beschaffung der Tools/Frameworks wird nachfolgend aufgelistet:

Tool/Framework	Beschaffungsinformation
Visual Studio	<ul style="list-style-type: none"> Version 2019 Community Edition <ul style="list-style-type: none"> Freeware Lizenz Version 2022 Professional Edition <ul style="list-style-type: none"> Kostenpflichtige Lizenz Downloadlink in der Linksammlung
Package Manager	<ul style="list-style-type: none"> NPM <ul style="list-style-type: none"> Freeware aber Anmeldung notwendig Beinhaltet <ul style="list-style-type: none"> JS + TS React + React Native Node.js CSS + HTML NUGET <ul style="list-style-type: none"> Freeware Beinhaltet <ul style="list-style-type: none"> C# Net.Core 5.0
Web API	<ul style="list-style-type: none"> ASP.NET <ul style="list-style-type: none"> In Visual Studio integriert Auf Basis von Net.Core 5.0 Testframework <ul style="list-style-type: none"> Postman <ul style="list-style-type: none"> Freeware Version <ul style="list-style-type: none"> Anmeldung notwendig Für bis zu drei Anwender Upgrade auf kostenpflichtige Version möglich Downloadlink in der Linksammlung

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

Windows Web Hosting Environment	<ul style="list-style-type: none"> • IIS-Server <ul style="list-style-type: none"> ○ Kostenpflichtige Lizenz ○ Downloadlink in der Linksammlung
Datenbanken	<ul style="list-style-type: none"> • Bei jeder Web-App <ul style="list-style-type: none"> ○ Prüfen ob Beschaffung kostenpflichtiger Lizenz notwendig • Zentrale DB auf Server <ul style="list-style-type: none"> ○ Bei Funktionszusammenhängenden Web-Apps ○ Prüfen ob Beschaffung kostenpflichtiger Lizenz notwendig • Downloadlink in der Linksammlung <ul style="list-style-type: none"> ○ Als Vorschlag für ein DB-System auf dem Server
Versionskontrolle	<ul style="list-style-type: none"> • GIT <ul style="list-style-type: none"> ○ Downloadlink in der Linksammlung
Virtuelle Umgebung	<ul style="list-style-type: none"> • VMware <ul style="list-style-type: none"> ○ Kostenlose Testlizenz ○ Kostenpflichtige Lizenz • Virtual Box <ul style="list-style-type: none"> ○ Freeware Lizenz • Downloadlinks in der Linksammlung

Tabelle 10: Beschaffungsinformationen für die Tools/Frameworks der Entwicklungsumgebung


		Seite 22
		Freigegeben von: TRi
		am: 23.07.2021

4.2. Freeware Lizenz vs. Kommerzielle Lizenz

Bei der Beschaffung von Tools/Frameworks für die Entwicklungsumgebung müssen die Lizenzbedingungen berücksichtigt werden. Weiterhin ist auf geltendes Lizenzrecht zu achten. Die nachfolgende Tabelle fasst die Bedingungen der Freeware und der kommerziellen Lizenz zusammen.

Freeware Lizenz	Kommerzielle Lizenz
<ul style="list-style-type: none"> • Freie Lizenznutzung <ul style="list-style-type: none"> ○ Prüfen welche Lizenzbedingungen gelten ○ Prüfen ob kommerzielle Nutzung möglich ist • Lizenzbeschränkung <ul style="list-style-type: none"> ○ Prüfen, ob Funktionsumfang der Freeware Lizenz ausreichend ist ○ Gegebenenfalls Beschaffung einer kostenpflichtigen Lizenz • Geänderte Lizenzbedingungen <ul style="list-style-type: none"> ○ Freie Lizenz wird kostenpflichtig ○ Einschränkung des Funktionsumfanges • Bei Umstellen auf kostenpflichtige Lizenz <ul style="list-style-type: none"> ○ Vergleichen des Funktionsumfanges zwischen aktuelles und alternatives Tool/Framework • Bei Kündigung der Lizenznutzung <ul style="list-style-type: none"> ○ Bereitstellung eines alternativen Tools/Frameworks muss gewährleistet sein ○ Integrationstest werden für die Alternative notwendig • Open Source Lizenz GNU GPL <ul style="list-style-type: none"> ○ Modifizierte Anwendung muss ebenfalls Open Source und GNU GPL lizenziert sein 	<ul style="list-style-type: none"> • Kostenpflichtige Lizenz • Funktionsumfang nach Lizenzstufe <ul style="list-style-type: none"> ○ Prüfen welche Lizenzstufe alle Anforderungen der Entwicklungsumgebung erfüllt • Nutzermenge für die Lizenz <ul style="list-style-type: none"> ○ Eventuell Beschaffung mehrerer Lizenzen notwendig • Lizenzbeschränkung <ul style="list-style-type: none"> ○ Prüfen ob Einschränkungen das Tool/Framework unbrauchbar machen ○ Bereitstellung eines alternativen Tools/Frameworks muss gewährleistet sein ○ Integrationstest werden für die Alternative notwendig

Tabelle 11: Übersicht Freeware Lizenz vs. kommerzielle Lizenz

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

5. Anwendungsbeispiel: Projekt-Manager als Web-App

Nach Realisierung der konzeptionierten Entwicklungsumgebung ist der Entwurf sowie die Entwicklung eines prototypischen Projekt-Managers als Web-App vorgesehen. Auf Basis des Ergebnisses der Anwendung soll das Einsatzgebiet der Entwicklungsumgebung auf die Individualisierung von Siemens Comos sowie möglicherweise MS Teams erweitert werden.

Der Projekt-Manager wird ein Erfassungssystem von Projektarbeitszeiten schaffen. Für das Management von Projekten wird eine Übersicht aller Projektarbeitszeiten der MA in einem Projekt aufrufbar sein. Die Release-Version soll das derzeitige Erfassungssystem von Projektarbeitszeiten der AFA ablösen. Derzeitig wird für diesen Zweck ProCos genutzt.

Das Ziel der Ablösung von ProCos soll eine Arbeitserleichterung für die, wie die unter den Punkt „Ziele“ genannten, AFA internen Bereiche sein.

Die Entwicklung dieser prototypischen Web-App wird als Proof of Concept dienen. Anhand der Ergebnisse dieser Anwendung wird die Funktionalität der Entwicklungsumgebung getestet. In diesem Konzept wird beispielhaft auf

- einen möglichen Programmablaufplan
- ein mögliches ER-Modell der DB

eingegangen. In einem Konzept für den Projekt-manager als prototypische Web-App, wird diese spezifiziert.

		Seite 24
		Freigegeben von: TRi
		am: 23.07.2021

5.1. Beispielhafter Programmablaufplan

Der Programmablaufplan beschreibt, wie der Algorithmus des Programmes übersetzt wird. Der Ablauf wird in einem grafischen Diagramm dargestellt. Die nachfolgende Grafik, zeigt einen beispielhaften Programmablaufplan des prototypischen Projekt-Managers Web-App.

Die elliptischen Felder markieren sowohl den Programmstart also auch das Programmende. Die Rechtecke beschreiben Prozesse der Anwendung. Entscheidungen in der Anwendung werden mit Rauten gekennzeichnet. Die einzelnen Schritte der Anwendungen werden mit richtungsweisenden Verbindungen beschrieben. Die Grafik ist von oben links beim Programmstart nach unten rechts zum Programmende zu lesen.

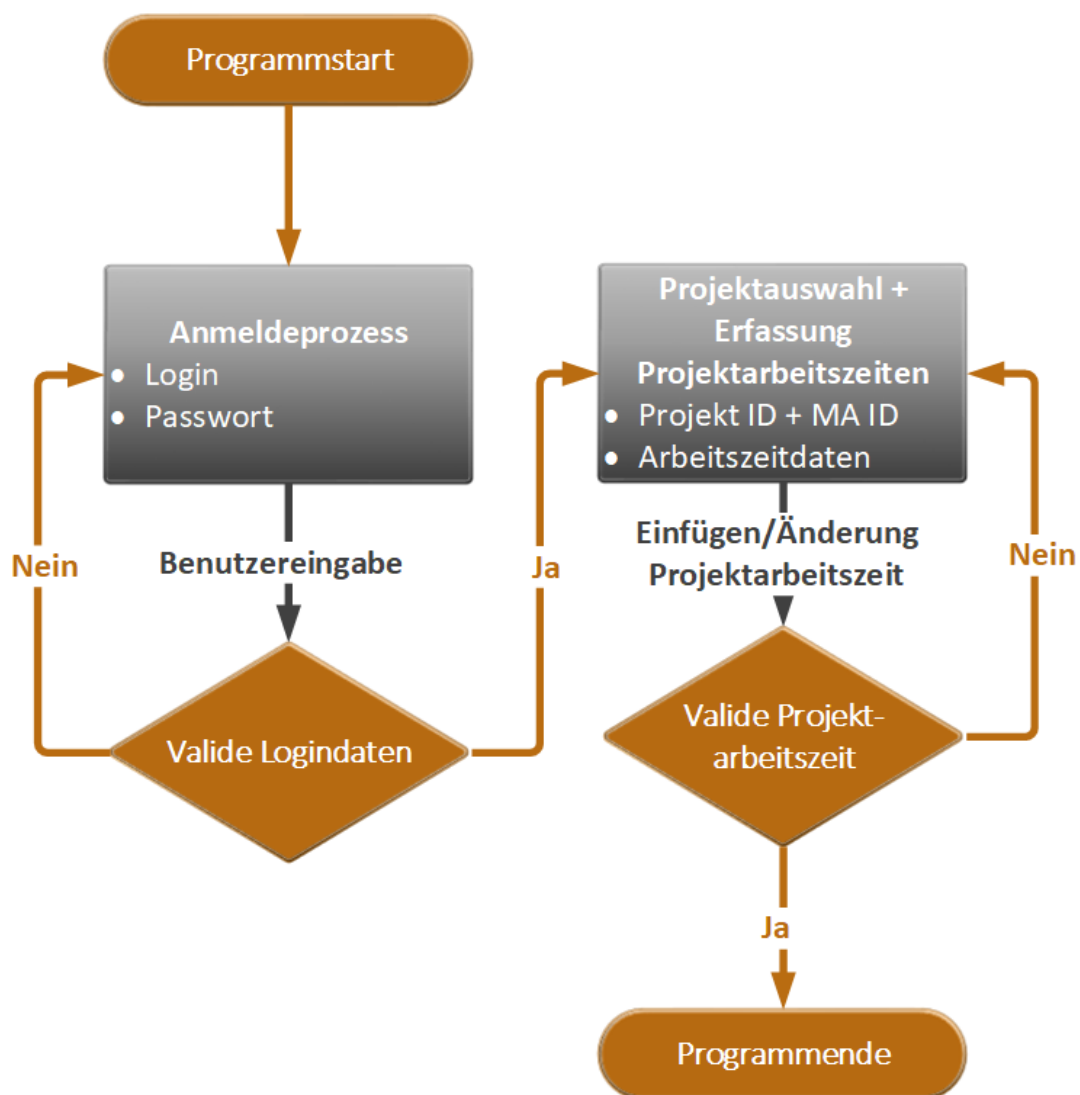


Abbildung 2: Beispielhafter Programmablaufplan - Projekt-Manager als Web-App

5.2. Beispielhaftes ER-Modell der DB

Das Entity-Relationship-Modell (ER-Modell) beschreibt für dieses Konzept den Zusammenhang zwischen der DB einer Anwendung sowie dessen Akteure. Die Akteure besitzen Eigenschaften, welche als Attribute bezeichnet werden. Wie die Akteure untereinander in Beziehung (Relation) stehen, wird mit Verbindungen in der Grafik gekennzeichnet.

Eine Beispielhafte Zusammenstellung für den Projekt-Manager als prototypische Web-App könnten drei Akteure in der Datenbank sein. Folgende drei Akteure wären möglich:

Person	Projekt	Projektarbeitszeit
<ul style="list-style-type: none"> • ID • Name (Vorname, Name) • Login • Projekt.ID (Fremdschlüssel, Mehrfachattribut) • Projektarbeitszeit.ID (Fremdschlüssel, Mehrfachattribut) 	<ul style="list-style-type: none"> • ID • Projektname • Projektbeschreibung • Person.ID (Fremdschlüssel, Mehrfachattribut) • Projektarbeitszeit.ID (Fremdschlüssel, Mehrfachattribut) 	<ul style="list-style-type: none"> • ID • Stunden • Minuten • Datum • Beginn • Ende • Person.ID (Fremdschlüssel) • Projekt.ID (Fremdschlüssel)

Tabelle 12: Beispielhafte Akteure in einer DB des Projekt-Managers als Web-App

In der nachfolgenden Grafik werden die Beziehungen der DB Akteure dargestellt. Die Rechtecke sind die Entitäten, also die Akteure der DB. Die Ellipsen kennzeichnen die Attribute also die Merkmale der Entität. Die Beziehungen zwischen den Entitäten werden mit den Rauten hervorgehoben. Die Grafik ist beginnend bei der Person, über das Projekt und zum Ende bei der Projektarbeitszeit zu lesen.

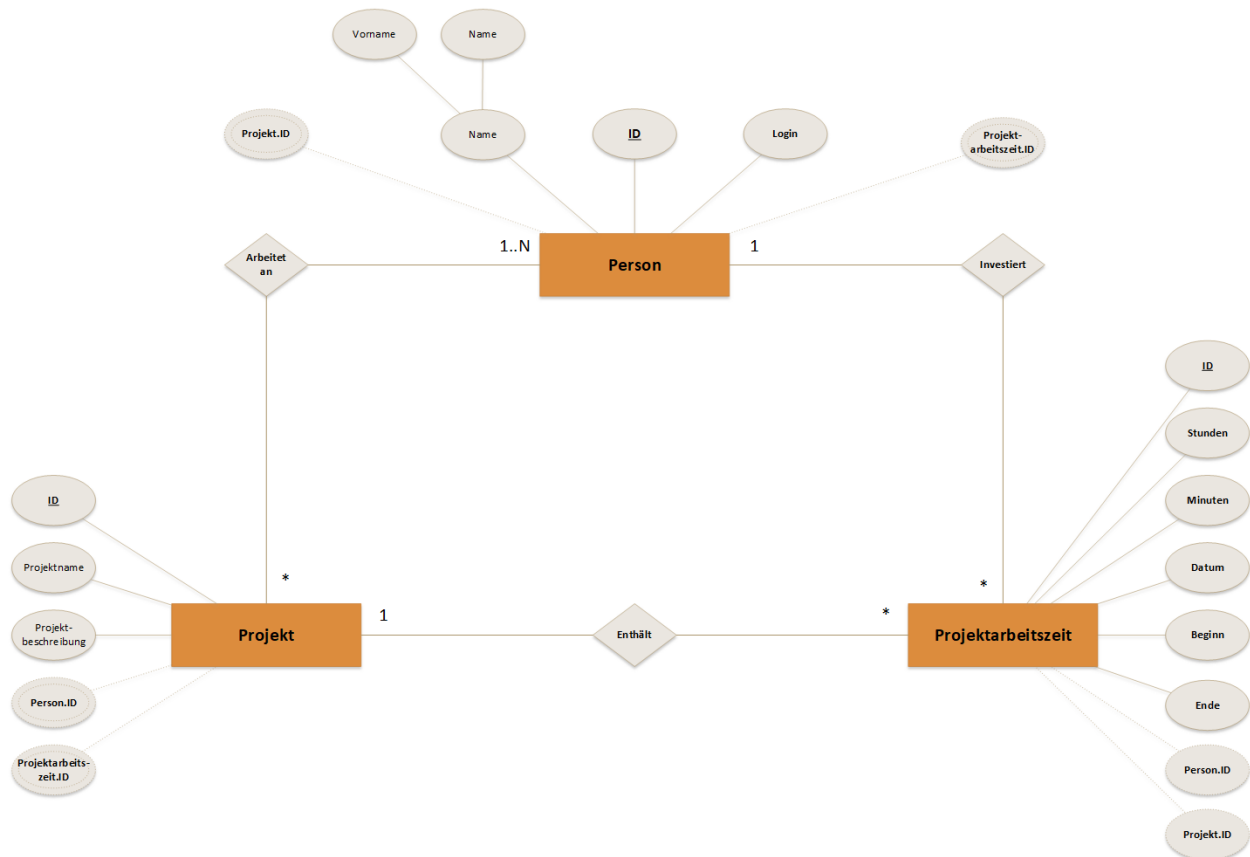


Abbildung 3: Beispielhaftes ER-Modell – Projekt-Manager als Web-App

6. Ergebnis des Konzepts

Das Konzept wird als Anleitung dienen, eine Entwicklungsumgebung für die Programmierung von Web-Apps in die Citrix-Server der AFA zu implementieren. Für anwendungsfremde Software-Entwickler ist dieses Konzept ein Einstiegspunkt in die Programmierung von Web-Apps. Folgende Ziele verfolgt die AFA:

- Customizing von Siemens Comos
- Möglicherweise Customizing von MS Teams

Bei der Beschaffung der Hardware sowie der Tools/Frameworks für die Entwicklungsumgebung sind folgende Punkte zu beachten:

- Kosten bei der Anschaffung/Lizenzierung
- Lizenzbedingungen und geltendes Recht bei der Lizenzierung

Die konzeptionierte Entwicklungsumgebung wird auf ein lokales System getestet. Zuvor wird ein Testplan erstellt, auf dessen Basis die lokale Entwicklungsumgebung mit samt seinen Tools/Frameworks schrittweise eingerichtet und getestet wird. Basierend auf den Ergebnissen, bestehend aus:

- Installationsanleitungen
- Dokumentationen

wird die konzeptionierte Entwicklungsumgebung für Web-Apps auf einen Server im Citrix der AFA eingerichtet.

Das in diesem Dokument erstellte Konzept für die Entwicklungsumgebung von Web-Apps mit den Bestandteilen

- Lokale Entwicklungsumgebung oder Server mit Entwicklungsumgebung
- Tools/Frameworks zum Programmieren von Web-Apps
- Bestehende sowie optionale Testumgebungen

findet Anwendung in einer prototypischen Web-App

- zur Erfassung von Projektarbeitszeiten
- mit Übersicht der Arbeitszeiten, welche von den MA der AFA investiert wurden

welche in einem Konzept detailliert spezifiziert wird. Diese Anwendung wird als Proof of Concept dienen, um die Funktionalität der Entwicklungsumgebung zu testen. Anhand der Ergebnisse dieses Prototyps wird das Einsatzgebiet der Entwicklungsumgebung erweitert.

Glossar

Das Glossar beinhaltet Anhänge für das Konzept, was Folgendes beinhaltet

- Definition von Begrifflichkeiten
 - Mit Verweis zu einer detaillierten Definition
- Linksammlung mit
 - Website zum Hersteller
 - Wiki
 - Installationsanleitungen
 - Tutorials

A. Begrifflichkeiten

Backend/Frontend

- Schichteinteilung der Funktionalität einer Web-App
- Frontend näher am Nutzer
- Backend näher am System
- https://de.wikipedia.org/wiki/Front-End_und_Back-End

Benefit

- Synonym für Vorteil
- <https://de.wikipedia.org/wiki/Benefit>

Business Case


- Prüfverfahren für die Rentabilität eines Geschäftsszenarios
- <https://de.wikipedia.org/wiki/Business-Case>

Cascading Style Sheets (CSS)

- Style sheet scripting language für festlegung des Erscheinungsbildes der Webseite
- Migration in HTML sowie JavaScript
- https://de.wikipedia.org/wiki/Cascading_Style_Sheets

Continuous Delivery Pipeline

- Sammlung von
 - Techniken
 - Prozessen
 - Tools
- Für die Verbesserung des Softwareentwicklungsprozesses
- https://de.wikipedia.org/wiki/Continuous_Delivery

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

Deployment

- Prozesse zur Installation von Software auf Rechnern
 - Automatisierte Installation durch Anweisung von der Ferne möglich
- Findet vor allem in der Software-Entwicklung statt
 - Temporäres Erstellen von Testumgebungen
- <https://de.wikipedia.org/wiki/Softwareverteilung>

Entity-Relationship-Modell (ER-Modell)

- Verfahren für eine Datenmodellierung
- Beinhaltet Akteure (Entity) sowie Beziehungen (Relations) zwischen den Akteuren
- Entities beinhalten Merkmale (Attribute)
- <https://de.wikipedia.org/wiki/Entity-Relationship-Modell>


Framework

- Allgemein
 - Dient als Programmiervorlage
 - Findet insbesondere Anwendung in
 - Objektorientierte Softwareentwicklung
 - Komponentenbasierte Entwicklungsansätze
 - <https://de.wikipedia.org/wiki/Framework>
- Testframework
 - Bestandteil zur Testautomatisierung
 - Erstellen einer temporären Testumgebung
 - Liefert Testergebnisse
 - Löscht danach Testumgebung
 - Continuous Delivery Pipeline für Testautomatisierung
 - <https://de.wikipedia.org/wiki/Testautomatisierung>
- Webframework
 - Software für die Entwicklung von
 - Dynamischen Websites
 - Web Apps
 - Webservices
 - <https://de.wikipedia.org/wiki/Webframework>

DotNet (.Net)

- Beinhaltet Net.Core 5.0
 - Entwicklung und Ausführung von Anwendungsprogrammen
- Grundlage für Web API „ASP.NET“
- https://de.wikipedia.org/wiki/.NET_5

		Seite VI
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

NAS Server

- Speicher der für ein Netzwerk auf einen Server zur Verfügung gestellt wird
- https://de.wikipedia.org/wiki/Network_Attached_Storage

Objektorientiertes Programmieren (OOP)

- Programmieren von Klassen als Vorlage von Objekten
 - Generierung von mehreren Objekten, welche unterschiedlich agieren
- Interfaces
 - Klassen mit leeren Methoden und Attributen als Schablone
- Templates
 - Universelles Attribut für den Einsatz beliebiger Datentypen
- Aufbau der Klassen in
 - Methoden
 - Attributen
- Vererbung von Klassen
 - Vererbung der Methoden
 - Vererbung der Attribute
- Sicherheitsabstufungen
 - Public
 - Zugriff von überall
 - Protected
 - Zugriff nur über eigene Klasse sowie vererbte Klassen
 - Private
 - Zugriff nur über eigene Klasse
- https://de.wikipedia.org/wiki/Objektorientierte_Programmierung


Programmablaufplan (PAP)

- Ablaufdiagramm für eine Anwendung
- <https://de.wikipedia.org/wiki/Programmablaufplan>

Progressive Web Application (PWA) vs. Native Application (NA)

- <https://topflightapps.com/ideas/native-vs-progressive-web-app/>
- PWA
 - Hybrid aus Website mit Responsive Design und App
 - Besitzt Elemente einer Native Application
 - https://de.wikipedia.org/wiki/Progressive_Web_App
- NA
 - Mobile Application
 - An einer bestimmten Plattform angepasst
 - https://de.wikipedia.org/wiki/Mobile_App

		Seite VII
	Freigegeben	von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

Proof of Concept

- Als Meilenstein zu betrachten
- Beleg für die Durchführbarkeit eines Vorhabens
- https://de.wikipedia.org/wiki/Proof_of_Concept

Remote Desktop Protocol (RDP)

- Ermöglicht Remote Control
- https://de.wikipedia.org/wiki/Remote_Desktop_Protocol

Repository

- Speicher für Softwarepakete
- Tabelle mit Inhalten in Repository mit Metadaten gespeichert
- https://en.wikipedia.org/wiki/Software_repository

Scaffolding

- Gerüst für eine vollständige Orientierungsgrundlage in Form von
 - Anleitungen
 - Denkanstößen
 - und weitere Hilfestellungen
- <https://de.wikipedia.org/wiki/Scaffolding>


SQL vs. NOSQL

- SQL (Structured Query Language)
 - Scriptsprache für Anfragen in relationalen Datenbanken
 - <https://de.wikipedia.org/wiki/SQL>
- NOSQL (Not only SQL)
 - Aufbrechen der relationen Datenbankstruktur
 - Keine festgelegten DB-Schemata mehr notwendig
 - Kann Joins vermeiden
 - Performancesteigerung
 - <https://de.wikipedia.org/wiki/NoSQL>

Toolchain

- Betrachtet den Funktionszusammenhang von Tools/Frameworks + Hardware
 - Für die Erfüllung einer Aufgabe
- <https://de.wikipedia.org/wiki/Toolchain>

		Seite VIII
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

Web API

- Konzept für die Entwicklung von Web-Apps
- Web Application Programming Interface (Web API)
 - Web server
 - Web browser
- https://en.wikipedia.org/wiki/Web_API


Workflow

- Beschreibt einen Arbeitsablauf
- Räumliche und zeitliche Reihenfolge von zusammengehörenden Arbeitsvorgängen
 - funktional
 - physikalisch
 - technisch
- <https://de.wikipedia.org/wiki/Arbeitsablauf>

Virtuelle Umgebung

- Simulation eines Rechnersystems
- Bildet Hardware eines hypothetischen Rechners nach
- https://de.wikipedia.org/wiki/Virtuelle_Maschine

		Seite IX
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

B. Linksammlung

JavaScript (Programmiersprache)

- <https://developer.mozilla.org/de>

Tutorial: The Morpheus Tutorials – Video (Deutsch)

- <https://www.youtube.com/watch?v=UeZi8a99iS0&list=PLNmsVeXQZj7qOfMI2ZNk-LXUAIxKrwDli>

Installationsanleitung: Microsoft – Testframework (Englisch)

- <https://docs.microsoft.com/en-us/visualstudio/javascript/unit-testing-javascript-with-visual-studio?view=vs-2019&tabs=mocha>

TypeScript (Programmiersprache)

- <https://www.typescriptlang.org>

Tutorial: Handbuch (Englisch)

- <https://www.typescriptlang.org/docs/handbook/>

Tutorial: The Morpheus Tutorials – Video (Deutsch)

- <https://www.youtube.com/watch?v=uo4PWU-R96k&list=PLNmsVeXQZj7pwHqtQSBXGBUNkyGGOJQXf>

Bootstrap

- <https://www.typescriptlang.org/docs/bootstrap>

React (UI Creating Tool)

- <https://reactjs.org>

Tutorial: The Morpheus Tutorials – Video (Deutsch)

- https://www.youtube.com/watch?v=HCJCFV-OGnY&list=PLNmsVeXQZj7oi_Q4whC28Yp12l1I-hauk


React Native (UI Creating Tool)

- <https://reactnative.dev>

Tutorial: The Net Ninja – Video (Englisch)

- <https://www.youtube.com/watch?v=ur6l5m2nTvk&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ>

		Seite X
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

Web API

Wiki

- https://en.wikipedia.org/wiki/Web_API

Postman (Testframework)

- <https://www.postman.com>

Postman - Download

- <https://www.postman.com/downloads/>

Web API (ASP.NET)

Tutorial: Tutorialsteacher – Web API

- <https://www.tutorialsteacher.com/webapi/web-api-tutorials>
- <https://www.tutorialsteacher.com/webapi/what-is-web-api>

Tutorial: Coding mi Jannick – Video (Deutsch)

- <https://www.youtube.com/watch?v=ay94twd-c0E>

Weitere Interpretationsmöglichkeiten

- <https://hub.packtpub.com/what-are-best-programming-languages-building-apis/>

Net.Core

- <https://versionsof.net/core/5.0/5.0.0/>

Wiki

- https://de.wikipedia.org/wiki/.NET_5

C# (Programmiersprache)

- <https://docs.microsoft.com/en-us/dotnet/csharp/>

Wiki (Englisch)

- [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))


Tutorial: Programmieren Starten – Video (Deutsch)

- <https://www.youtube.com/watch?v=l0aKBziWBH8&list=RDCMUVCVdfgrCLfJQfO5EgPIzaYaq&index=2>

Installationsanleitung: Microsoft – C# Testframework (Englisch)

- <https://docs.microsoft.com/en-us/visualstudio/test/walkthrough-creating-and-running-unit-tests-for-managed-code?view=vs-2019>

		Seite XI
	Freigegeben	von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

Visual Studio (IDE)

- <https://visualstudio.microsoft.com/de/>

Download

- <https://visualstudio.microsoft.com/de/downloads/>

Installationsanleitung: Microsoft – Visual Studio (Deutsch)

- <https://docs.microsoft.com/de-de/visualstudio/install/install-visual-studio?view=vs-2019>

Wiki: Xamarin – Mobile Development Plugin

- <https://de.wikipedia.org/wiki/Xamarin>

Node.js

- <https://nodejs.org/en/>

NPM (Package Manager)

- <https://www.npmjs.com>

Wiki (Deutsch)

- [https://de.wikipedia.org/wiki/Npm_\(Software\)](https://de.wikipedia.org/wiki/Npm_(Software))

Installationsanleitung: Microsoft – Integration in Visual Studio (Englisch)

- <https://docs.microsoft.com/en-us/visualstudio/javascript/npm-package-management?view=vs-2019>

NUGET (Package Manager)

- <https://www.nuget.org/>


Wiki (Deutsch)

- <https://de.wikipedia.org/wiki/NuGet>

Installationsanleitung: Microsoft – Integration in Visual Studio (Deutsch)

- <https://docs.microsoft.com/de-de/nuget/consume-packages/install-use-packages-visual-studio>

		Seite XII
		Freigegeben von: TRi
		am: 23.07.2021

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

MS IIS

- <https://www.iis.net/>

Wiki

- https://de.wikipedia.org/wiki/Microsoft_Internet_Information_Services

Download

- <https://www.iis.net/downloads>

GIT (Versionskontrolle)

- <https://git-scm.com/>

Wiki

- <http://de.wikipedia.org/wiki/Git>

Download

- <https://git-scm.com/downloads>

Progressive Web Application (PWA)

Wiki

- https://de.wikipedia.org/wiki/Progressive_Web_App

Native Application (NA)

Wiki

- https://de.wikipedia.org/wiki/Mobile_App
- [https://en.wikipedia.org/wiki/Native_\(computing\)](https://en.wikipedia.org/wiki/Native_(computing))

GNU General Public License

Wiki

- https://de.wikipedia.org/wiki/GNU_General_Public_License

		Seite XIII
		Freigegeben von: TRi
		am: 23.07.2021

Virtuelle Umgebung

Wiki

- https://de.wikipedia.org/wiki/Virtuelle_Maschine

VMware + Download

- <https://www.vmware.com>
- <https://www.vmware.com/de/download.html>

VirtualBox + Download

- <https://www.virtualbox.org>
- <https://www.virtualbox.org/wiki/Downloads>

MySQL (DB)

- <https://www.mysql.com/de/>

Wiki

- <https://de.wikipedia.org/wiki/MySQL>

Download

- <https://www.mysql.com/de/downloads/>

SQL (Programmiersprache)

Wiki (Deutsch)

- <https://de.wikipedia.org/wiki/SQL>

Tutorial: Programmieren Starten – Video (Deutsch)

- https://www.youtube.com/watch?v=fgOiWEGNJ-o&list=RDCMUVCVdfgrCLfJQfO5EgPlzaYAQ&start_radio=1&rv=fgOiWEGNJ-o&t=0


ER-Modell

Wiki (Deutsch)

- <https://de.wikipedia.org/wiki/Entity-Relationship-Modell>

Tutorial: Creately (Deutsch)

- <https://creately.com/blog/de/diagramme/entity-relationship-modell/>

	Projekt: GIS2ALM	Verfasser: TRi
	Thema: Entwicklungsumgebung für Web-Apps	Datum: 05.07.2021
		Ausgabe: V01

HTML

Wiki (Deutsch)

- [https://de.wikipedia.org/wiki/Hypertext Markup Language](https://de.wikipedia.org/wiki/Hypertext_Markup_Language)

Tutorial: The Morpheus Tutorials – Video (Deutsch)

- <https://www.youtube.com/watch?v=pM-G2sIFPkA&list=PLNmsVeXQZj7qlbKPeroqn3-BkUTWzYBT4>

		Seite XV
		Freigegeben von: TRi
		am: 23.07.2021