

Modulidentifikation

Modulnummer	292
Titel	Applikation mit Webtechnologien clientseitig implementieren
Kompetenz	Applikationen mit Webtechnologien für unterschiedliche Geräte- und Systemklassen umsetzen, dabei mit anderen Diensten kommunizieren, sowie diese Applikationen testen und publizieren.
Handlungen	<ol style="list-style-type: none"> 1. Kommuniziert mit anderen Web-Diensten über definierte Schnittstellen. 2. Bewerkstelligt Datenverarbeitung und -haltung clientseitig. 3. Testet und publiziert Applikationen mit Webtechnologien und verwaltet Abhängigkeiten. 4. Entwickelt clientseitige Webapplikationen mit Hilfe von Frameworks. 5. Wendet Werkzeuge für die Entwicklung, Wartung und das Hosting von Applikationen mit Webtechnologien an.
Persönliche Kompetenzen	
Kompetenzfeld	Web engineering, Application engineering
Objekt	Applikation mit Webtechnologien, welche über Schnittstellen mit bestehenden Backends kommuniziert, notwendige Daten clientseitig und offline verwaltet, sowie den Benutzer authentifiziert, erstellen, diese testen und publizieren.
Nachweis	
Niveau e-CF	4
Voraussetzungen	291 - Benutzeroberfläche für Webapplikation clientseitig implementieren
Arbeitsaufwand ca. h	40
Anerkennung	Eidg. Fähigkeitszeugnis
Modulversion	1.1
MBK Release	

Handlungsnotwendige Kenntnisse

Handlungsnotwendige Kenntnisse beschreiben Wissen, das die kompetente Ausführung der Handlungen eines Moduls unterstützt. Diese Kenntnisse dienen der Orientierung und sind nicht abschliessend definiert. Die daraus folgende Konkretisierung der Lernziele und das Festlegen des Lernwegs für den Kompetenzerwerb sind Sache der Bildungsanbieter.

Modulnummer			
Titel		Applikation mit Webtechnologien clientseitig implementieren	
Kompetenzfeld		Web engineering, Application engineering	
Modulversion		1.1	
MBK Release			
Handlungsziele und handlungsnotwendige Kenntnisse	1	1.1	Kennt Konzepte der asynchronen Programmierung (z. B. Promises, Async / Await, ...) mit Webtechnologien und deren Anwendungszweck.
		1.2	Kennt die Funktionsweise moderner Web-APIs (z. B. REST, GraphQL).
		1.3	Kennt Möglichkeiten sich gegen eine API zu authentifizieren (z. B. JWT, OAuth2).
	2	2.1	Kennt Möglichkeiten zur clientseitiger Datenhaltung (z. B. IndexedDB, LocalStorage).
		2.2	Kennt die Konzepte um Daten clientseitig zu Verwalten und auf Veränderungen zu reagieren (one-way binding z. B. mit Redux/Flux stores, two-way binding z. B. mit MobX).
		2.3	Kennt Hilfsmittel um rechenintensive Verarbeitung zu bewerkstelligen (background processing).
		2.4	Kennt die notwendigen Schritte, um eine Applikation mit Webtechnologien offline verfügbar zu machen (z. B. ServiceWorkers, Cache API, ...).
	3	3.1	Kennt Konzepte um Abhängigkeiten von JavaScript-Applikationen mit einer Paketverwaltung zu definieren, installieren und aktualisieren (z. B. npm, yarn).
		3.2	Kennt Konzepte um Abhängigkeiten von JavaScript-Applikationen mit einer Paketverwaltung zu definieren, installieren und aktualisieren (z. B. npm, yarn).
		3.3	Kennt eine Vorgehensweise, wie entfernte APIs für Testfälle imitiert werden können (Mocking).
		3.4	Kennt die Unterschiede zwischen den verschiedenen Versionen von JavaScript, sowie die Möglichkeiten eine neuere Version für ältere Browser zugänglich zu machen (z.B. mit Polyfills oder transpiling).
	4	4.1	Kennt die Funktionsweise von JavaScript-Frameworks (z. B. Angular, React, Vue, ...) und kann diese gezielt einsetzen.
		4.2	Kennt Hilfsmittel um clientseitige Applikationen mit Hilfe eines Frameworks zu erstellen, optimieren und publizieren.
	5	5.1	Kennt Editoren und Entwicklungsumgebungen um clientseitige Webapplikationen zu entwickeln.
		5.2	Kennt Möglichkeiten die Applikation während der Laufzeit zu untersuchen (z. B. mit Entwicklertools von Browsern, SourceMaps).
	5.3	Kennt die Unterschiede zwischen einer Entwicklungsumgebung und einer produktiven Umgebung und kann damit umgehen.	