

Ausbaustufe 3

(Abgabe: 24. 1. 2021, 24:00 Uhr)

(A3.1) *Apollo.Core*: Führen Sie die notwendigen Erweiterungen an *Apollo.Core* durch. Überprüfen Sie nochmals Ihr Design, vor allem hinsichtlich einer sauberen Trennung der verschiedenen Schichten und überarbeiten Sie gegebenenfalls Ihre Implementierung (Refactoring).

(A3.2) *Apollo.Web*: Die Präsentationsschicht ist in Form einer Single-Page Applikation (SPA) mit Angular 10 zu realisieren. Versuchen Sie, in Ihrer Lösung soweit wie möglich zusätzliche Steuerelemente (z. B.: *Bootstrap*, *SemanticUI*, *DevExtreme*, etc.) zu verwenden. Diese Zusatzbibliotheken bieten diverse Komponenten an, die für die Erstellung des Web-Frontends hilfreich sind.

Legen Sie das Design Ihres Systems so aus, dass der Web-Klient über *Apollo.Api* auf die Geschäftslogik zugreifen kann. Konfigurieren Sie die Anwendung so, dass sie leicht an die Anforderungen eines Kunden angepasst werden kann (z. B. Host und Port des Web-Service). Achten Sie außerdem auch auf eine ausreichende client- und serverseitige Validierung.

Setzen Sie eine geeignete Authentifizierung mit OAuth2 und OpenID Connect ein, sodass nur jene Personen Daten modifizieren können, die dazu befugt sind. Es reicht aus, wenn die Web-Oberfläche authentifiziert wird. Das REST-Service muss das nicht leisten!

Fügen Sie geeignete Funktionalität zur Präsentationsschicht hinzu, sodass Benutzer von der besonderen Leistungsfähigkeit Ihrer Anwendung beeindruckt sind. Die Benutzerschnittstelle soll dabei den aktuellen Stand in der Gestaltung von Web-Anwendungen wiedergeben.

Da Sie den Umgang mit den Techniken von Angular üben sollen, ist es Ihnen **nicht** erlaubt, die Services aus z.B. OpenAPI zu generieren. Sie können sich jedoch gerne Umsetzungskonzepte davon anschauen und diese selbstständig einbauen (Anwendung von Konzepten, kein copy-paste). Beschreiben Sie diese dann auch entsprechend in der Dokumentation.

Für eine Benotung mit Sehr Gut ist ein erhöhter Einsatz gefordert: Beschäftigen Sie sich auch mit Techniken, die nicht in der Übung besprochen worden sind und wo ein Einsatz im Projekt sinnvoll erscheint, z.B. Pipes, Internationalisierung, Unit Testing, Validierung und Security, asynchrones Laden von Modulen, etc. Die Anwendung dieser Techniken sollte über bloßes Kopieren aus Tutorials, bzw. Anwenden von vorhandenen Bausteinen hinausgehen (z.B. erstellen Sie eigene Pipes, eigene Validatoren, etc.) und Sie sollten in der Lage sein, bei der Übungspräsentation die Funktionsweise der von Ihnen verwendeten Konzepte im Detail zu erläutern.

(A3.3) *Apollo.Api*: Diese Systemkomponente ist mit ASP.NET Core MVC als REST-basiertes Web-Service zu realisieren. Die Anforderungen an die zu exportierende Funktionalität des Web-Service ergeben sich aus der Funktionalität von *Apollo.Web*. Das REST-API ist nicht mit OAuth2 abzusichern.

(A3.4) *Tests*: Überprüfen Sie, ob auch nach Abschluss aller Entwicklungsarbeiten alle Unit- und Integrationstests erfolgreich ausgeführt werden können.

(A3.5) *Dokumentation*: Führen Sie die notwendigen Ergänzungen an der Dokumentation durch. Dokumentieren Sie auch den WEA5-Teil ausführlich (Architektur, UML-Diagramme, kommentierte Screenshots, Installationsanleitung sowie Benutzerschnittstellen-Beschreibung) und erstellen Sie dafür ein eigenständiges Dokument. Achten Sie darauf, dass auch der WEA5-Teil adäquat dargestellt wird.