Demoprojekt

Anforderungen

* Alles drin, was Kurs darstellt
* Vor- und zurückstellbar
* Anpassbar

Inhalt

WebArc-Projekt

* Schichten-Modell
* Contracts portable
* Datenmodell flexibel zwischen portabel und Full.Net
  + Codecentric Reversepoco
  + Configuration-Klassen
  + Annotations
* DAL wahlweise mit Full.Net und EF7 (.Net Core)
* Callback-Konzept (SignalR vs Push-Notification)
* Api (SL) als .Net.Exe oder .Net Core wahlweise
  + KundenDB mit Migration
* Webapp als Mvc, Mvc.Core, mit Angular-App
  + Bootstrap mit schönem Layout
  + Portal-Seite mit Anmeldung/Registrierung/Kunden-DB
  + Seite mit Standard-CRUD (Liste, Details, Neu, Edit, Delete) und Validierung
  + Seite mit Callback-Mechanismus (z.B. Auftrags-Status)
  + Async-Seite (z.B. Details aus eh. Kurs)
  + DI

Vorgehens-Alternativen:

Client-centric:

* WebClient + Dummy im Proj
* WebApi mit Dummy aus WebClient (eigene Model-Lib, und Interface-Lib)
  + WebClient lädt aus WebApi
* NetCore-Client / Angular-Client
  + lädt aus WebApi
* Windows / Mobile Client
  + lädt aus WebApi
* WebApi auf Basis von NetCore / Docker
* Erweiterungen WebApi
  + Authentication / CORS
  + SignalR
  + Push
* BL
  + in WebApi (und Core WebApi) einziehen
  + DI (BL in WebApi)
  + Unittest dazu
* DAL
  + Reverse-Poco
  + Integrationstest
  + DI (Context)
  + DI (DAL in BL)
  + in BL-Prozess
  + in eig. WebApi
    - Mit Auth (innerhalb VPN)

Data-centric:

* Model-Lib und Interface-Lib
* DAL
  + Reverse-Poco
  + Integrationstest
  + DI (Context)
* BL
  + Unittest
  + DI (DAL in BL)