

Evaluierung dbh Web-Frameworks

Im Rahmen der Evaluierung eines für dbh passenden Web-UI-Frameworks sollen drei aktuelle Web-Frameworks genauer untersucht werden.

- [Evaluationsmatrix zu den Web-Frameworks \(Vue, React, Angular 8\)](#)
- [Fazit Angular-Prototyp](#)
- [Fazit React-Prototyp](#)
- [Fazit Vue-Prototyp](#)
- [Finale Evaluierung](#)

Web-Recherche

Vorab soll eine Web-Recherche durchgeführt werden um allgemeine Gesichtspunkte wie Nachhaltigkeit, Verbreitung, Popularität, ... zu beurteilen.

Eine Evaluationsmatrix befindet sich [hier](#).

Rahmenbedingungen

Client/Server Kommunikation

In jedem Fall soll die Kommunikation mit dem A3 Server über standardisierte RESTful Services laufen. Über diese weit verbreitete und etablierte Technologie können standardisierte HTML UI's versorgt werden. Diese Vorgehensweise ermöglicht eben auch den Einsatz von externen "Standard" Web-Entwicklern. Zudem können diese Schnittstellen auch von anderen Software-Komponenten konsumiert werden (SAP, ERP, ...).

Client Frameworks/Bibliotheken

Im Moment der Untersuchung sind drei große Player vorzufinden: **Angular**, **React** und **Vue**. Alle drei stellen JavaScript basierte Frameworks zur Erstellung von Single Page Applications (SPA) zur Verfügung. **Angular** ist im Vergleich zu den anderen beiden etwas umfangreicher und könnte als Framework bezeichnet werden während **React** und **Vue** eher als Bibliotheken verstanden werden.

Prototyp Umfang Client

Um einen Eindruck im praktischen Umgang mit dem jeweiligen Kandidaten zu entwickeln soll folgender Umfang mit jedem Prototyp implementiert werden.

- Ein Auflister mit Objekten vom Typ Vorgang aus dem A3 Test-Client (ohne Suchfunktion)
- Erstellen eines Objekts vom Typ Vorgang
- Aktualisieren eines Objekts vom Typ Vorgang
- Löschen eines Vorgangs

Anzahl der Spalten und Felder des Editors sind im TestClient ersichtlich. Als Komponenten Bibliothek soll die Open-Source [Element UI](#) benutzt werden, damit nicht auf den rudimentären HTML-Elementen aufgebaut werden muss.

Screenshots des Prototype in Vue + Element.

localhost:8080

Apps Bookmarks News Privat music Programming dbh Portugal localhost:8080 A3 Board - Agile Bo... Überseite [enims] dbh Logistics IT AG... Startseite >

dbh

Star Wars API [Locate Persons](#)

search

More Copy Delete

ID	First	Second	Paid	Nickname	City	Birthday
9	Reyu	Murakami	0	The Flow	Gotham	17.3.1997
1	Joshi	Rudolf	245	Rambo	Berlin	18.3.1997
2	Andy	Krietschme	490	Cruncher	New Del	19.3.1997
3	Helen	Andy	735	Delightful	Berlin	20.3.1997
4	Carlo	Helen	980	Winced	Rom	21.3.1997
5	Peter	Carlo	1225	Ordnly	Budapest	22.3.1997
6	John	Peter	1470	King	New Orleans	23.3.1997
7	Reyu	John	1715	The Flow	Gotham	24.3.1997
8	Joshi	Reyu	1960	Rambo	Berlin	25.3.1997

Id: Date of Birth:

Address:

First: Second:

City: Street: Country:

Other:

Nickname:

Comment:

More Save Send

Es muss ein entsprechender RESTful Service implementiert werden mit den CRUD Funktionen wie oben beschrieben. Hierbei soll es noch nicht um die Steuerung von Berechtigungen oder die Authentifizierung gehen. Für die Implementierung soll die neue A3 Jersey Integration verwendet werden.

- Alle drei Frameworks bieten ein Command Line Interface (CLI) welches für jeden Prototyp genutzt werden soll.
- Als Entwicklungsserver und als Build/Test Tool soll 'npm' verwendet werden.
- Es soll TypeScript verwendet werden, um einfache Programmierfehler zu vermeiden.
- Als IDE soll Webstorm, von IntelliJ verwendet werden (30 Tage Test-Lizenz, später 100€/Jahr/Entwickler). Entspricht unserer gewohnten Entwicklungsumgebung für Java.
- Die Debugging-Möglichkeiten sollen getestet werden.
- Der Editor soll ein TextField, ein numerisches Field, ein DateField, ein ComboBoxField, eine Checkbox, ein ProxyField enthalten.
- Der Editor soll die 1 zu n Beziehung zwischen Vorgang und Positionen als Liste mit Detail-Editor abbilden (ListPaneAndEditor).
- Der Editor soll mit automatischen Tests ähnlich JUnit getestet werden.
- Hierfür ist eine Evaluierung der üblichen TestTools/Frameworks notwendig.
- Für die Notation des UI soll eine Fluent-API mit Builder-Pattern entwickelt werden.
- Die Builder für Forms, Columns, Commands, sollen unabhängig von der verwendeten Komponenten lib entwickelt werden, damit sie nur einmal entwickelt werden müssen.
- Anstatt rechte Mouse Menüs sollen Buttonleisten mit Primär- und Sekundär-Commands verwendet werden.
- Die UI soll Mobile tauglich sein (Tablet+Desktop) Responsive UI soll im Prototyp vorführbar sein.
- Performance bei mindestens 500 Feldern in einer Maske muss als akzeptabel getestet werden. Hierbei ist das Binding zwischen Model und Feld als mögliche Performancebremse auszuschließen.
- Gibt es Alternativen zur Element UI lib?