

CMS

Thema: Projektdokumentation Modul 151

Dokumentinformationen

Dateiname: doku-template-short.docx
Speicherdatum: 02.11.2022

Autoreninformationen

Autor: Anton Detken
E-Mail: anton@detken.ch
Tel: +41 77 474 12 33

Inhaltsverzeichnis

1	Dokument	Error! Bookmark not defined.
---	----------------	-------------------------------------

Abbildungsverzeichnis

No table of figures entries found.

Tabellenverzeichnis

No table of figures entries found.

1 Beschreibung

Die CMS-Webapplikation «DetkenCMS» ist ein Content Management System für Firmen-Webseiten. Dabei können Mitarbeiter, welche auf der Firmenseite angezeigt werden sollen, verwandelt werden. Der Nutzen ist zur reinen Darstellung, nicht zur tatsächlichen Mitarbeiterverwaltung im Unternehmen.

2 Architektur

Die Web-Applikation besteht aus zwei Frontends und einem Spring Boot Backend.

2.1 Frontends

Für beide Frontends wird **Nuxt**, ein VUE-Framework, verwendet, wobei das öffentliche, zur Darstellung dienenden Frontend mit Nuxt 3 und das CMS Frontend mit Nuxt 2 geschrieben wurde. Der Grund dafür ist, dass es für Nuxt 3 noch keine Unterstützung für einige Module gibt, unter anderen Auth, weshalb ich das CMS Frontend downgraden musste (Das öffentliche Frontend braucht keine Authentifizierung, da es nur GET-Requests versendet).

2.2 Backend

Das Backend läuft mit einer Spring Boot REST API.

2.3 Login / Sessions

Nach dem Aufrufen des Login-Endpoints mit einem validen User-Passwort-Paar wird ein 30 Tage gültiger JWT-Token, welcher zur Authentifizierung verwendet werden kann, zurückgegeben.

Der Token wird im Local-Storage und in einem Cookie im CMS-Frontend gespeichert. Im Frontend wird Nuxt-Auth zur Speicherung des Tokens verwendet.

2.4 Domain Package

Aufbau des Domain Package ist folgender:

- appuser: Userverwaltung mit Login, JWT, etc ; wird für den CMS-Nutzer verwendet
- authority: CRUD-Authoritäten (CREATE, READ, UPDATE, DELETE) werden hier verwaltet ; sind mit Rollen verknüpft
- data: Hier werden alle Entitäten verwaltet, die zur Darstellung dienen ; in diesem Fall sind das aktuell nur Mitarbeiter, welche in data/employee sitzen ; der File Ordner besitzt IO-Utills, welche von allen Datenentitäten verwendet werden können
- exceptions: Custom-Java-Exceptions
- role: Die Rollen ADMIN, WEBMASTER und EMPLOYEE werden hier verwaltet ; sind mit dem User und Authorities verknüpft
- security: Hier liegt nur der SecurityConfig für Spring

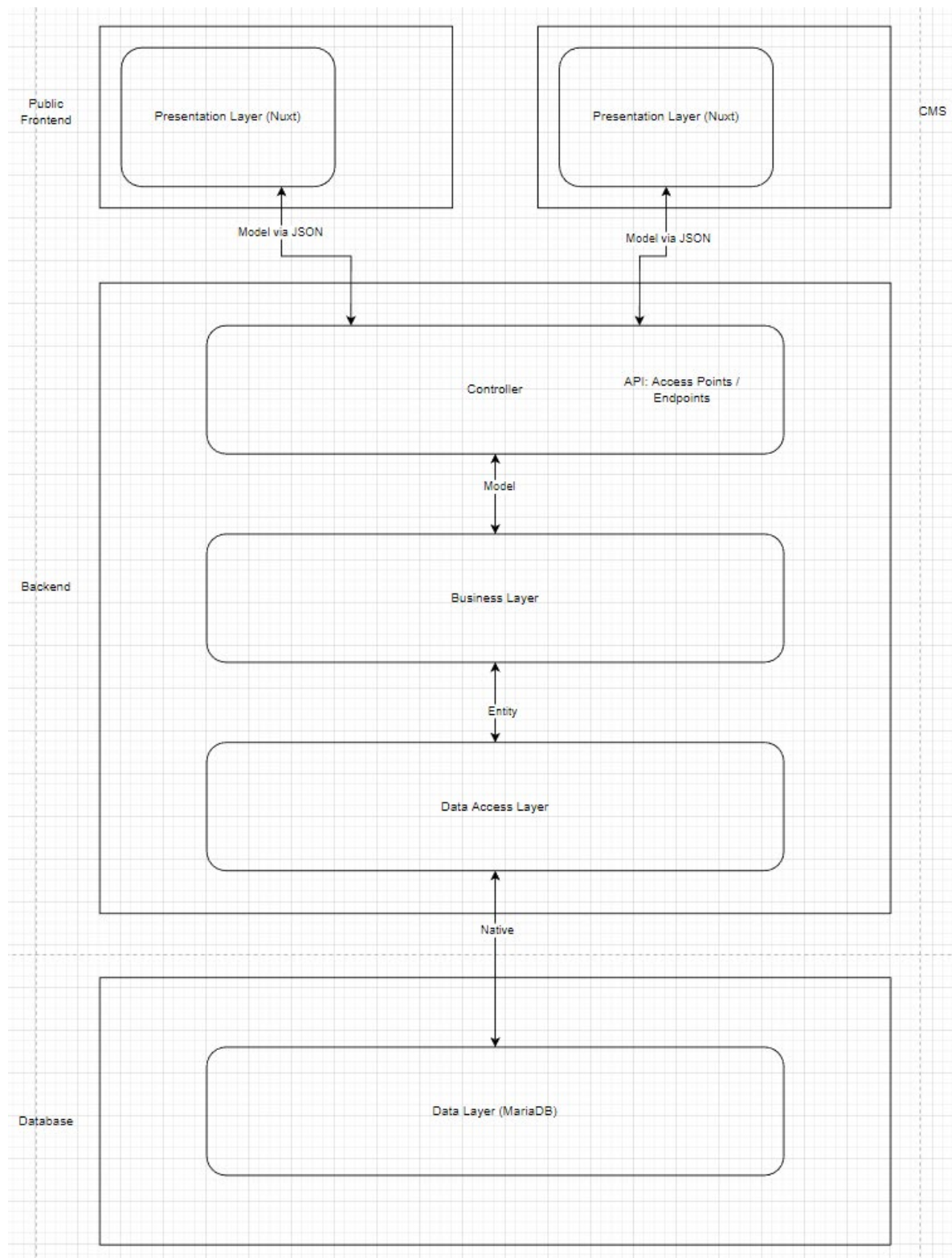


Abbildung 1 Architektur

3 Testing

3.1 Testkonzept

Tests sind unverzichtbar. Für die Tests wurde eine Vorlage für ein Testkonzept mit Erklärungen erstellt.

Testfall-Nummer		Nummer			
Bezeichnung		Kurzer Titel des Tests			
Beschreibung / Zu testende Funktionalität		Kurzbeschreibung			
Klassifikation		TP / FP / TN / FN			
Use-Case-Nummer		Use-Cases der getestet wird			
Datum		Datum der Testdurchführung (YYYY-MM-DD)			
Tester					
Voraussetzung		Umgebung, welche vorausgesetzt wird			
Testumgebung					
	Software	Beschreibung der Softwareumgebung, in welcher getestet wird (i.e: Datenbank «Buchhaltung» oder Chrome 98.0.4758.102)			
	Betriebssystem	i.e: MacOS 12.0.1			
	Hardware	Falls relevant: Bildschirm, Tastatur, Maus, ...			
Testschritte					
	Aktion	Erwartetes Ergebnis	Effektives Ergebnis	Erfüllt	Kommentar
1					
2					

Tabelle 1 Testkonzept

3.1.1 Erklärung Klassifikation

Es gibt verschiedene Testarten. Man unterscheidet zwischen True Positive, False Positive, True Negative, False Negative.

	True	False
Positive	Es wird erwartet, dass etwas <i>funktioniert</i> und es <i>funktioniert</i> schlussendlich auch.	Es wird erwartet, dass etwas <i>nicht</i> funktioniert, aber es funktioniert schlussendlich <i>doch</i> .
Negative	Es wird erwartet, dass etwas <i>nicht</i> funktioniert und es funktioniert schlussendlich auch <i>nicht</i> .	Es wird erwartet, dass etwas <i>funktioniert</i> , aber es funktioniert <i>doch nicht</i> .

Tabelle 2 Test-Klassifikationen

3.1.2 Template

Testfall-Nummer					
Bezeichnung					
Beschreibung / Zu testende Funktionalität					
Klassifikation					
Use-Case-Nummer					
Datum					
Tester					
Voraussetzung					
Testumgebung					
	Software				
	Betriebssystem				
	Hardware				
Testschritte					
	Aktion	Erwartetes Ergebnis	Effektives Ergebnis	Erfüllt	Kommentar
1					-
2					-
3					-

Tabelle 3 Testfall-Template

4 Reflexion

Obwohl ich Spring Boot und Nuxt schon zuvor verwendet hatte, hab ich viel gelernt. Die grössten Fortschritte hab ich im Bereich Authentifizierung gemacht, da ich das zuvor noch nie *richtig* implementiert hatte. Viele Dinge wie Session Tokens und JWT verstehe ich nun.

Spring Boot hatte ich lange nicht mehr verwendet. Ich konnte mein Halbwissen auffrischen und stark erweitern.

Die grössten Schwierigkeiten hatte ich tatsächlich auch mit dem Login. An diesem Punkt bin ich lange steckengeblieben, was mir aber gelehrt hat, was möglich ist und was eben nicht möglich ist. Da ich dafür so viel Zeit gebraucht hatte, konnte ich im *Frontend bei weitem nicht alles vom Backend umsetzen*. In Zukunft werde ich meine Zeit besser einplanen müssen, dass das Endprodukt nicht abgespeckt ist.

Das Projekt war alles in allem ein Erfolg und ich konnte mir viel neues Wissen aneignen.