

1 Abkürzungsverzeichnis

JWT Jason Web Token

URI Uniform Resource Identifier

HTTP Hypertext Transfer Protocol

JSON JavaScript Object Notation

MVC Model View Controller

HTML Hypertext Markup Language

CSS Cascading Style Sheets

DOM Document Objekt Model

SPA Single Page Application

CLI Command Line Interface

API Application Programming Interface

2 Einführung

Derzeit ist auf den meisten modernen Webseiten mit Benutzerinteraktion eine Anmelde und Abmeldemöglichkeit implementiert. Zu dem wird in den meisten Fällen zwischen den Nutzern über bestimmte Benutzerrollen und Benutzergruppen unterschieden. Dabei geht es darum, Nutzern individuell nach Berechtigung der vorgegebenen Benutzerrolle und/oder Benutzergruppe ihren zugriffsmöglichen Inhalt zur Verfügung zu stellen.

Aktuell ist in Blattwerkzeug keine Benutzer Authentisierung, Authentifizierung und Autorisierung implementiert. Dies hat zur Folge, dass bisher keine individuelle Benutzerinteraktion stattfinden kann. Im Rahmen dieser Thesis soll genau dieses Problem gelöst werden. Nach Behandlung der Thesis soll es möglich sein, sich mit einer standardisierten Registrierung bei Blattwerkzeug anzumelden. Außerdem soll es ebenfalls

möglich sein, sich über externe Anbieter anzumelden. Zusätzlich soll je nach Benutzerrolle und Benutzergruppe des angemeldeten Nutzers unterschiedlicher Inhalt dargestellt werden.

3 Technologien

3.1 Blattwerkzeug

Blattwerkzeug ist ein quelloffenes Projekt, dass Informatik-Interessierten das Programmieren von HTML Grundgerüsten und SQL Statements per 'drag and drop' näher bringen kann. Dabei versteckt Blattwerkzeug den Syntax nicht vor dem Nutzer, sondern gibt ihm die Möglichkeit diesen gleich mit ein zu sehen. Dennoch ist es dem Nutzer einfach gemacht, mit visuellen Elementen teile der Informatik kennen zu lernen.

Dabei hat es sich Blattwerkzeug vor allem als Aufgabe gemacht an Schulen aufzutreten. Mit Blattwerkzeug wird Lehrern ein Werkzeug in die Hand gelegt, mit dem einfacher und informativer Informatik Unterricht gestaltet werden kann. Somit kann der veraltete und doch sehr Office-lastige Informatik Unterricht komplett erneuert und interessanter gestaltet werden.

Zu dem aktuellen Zeitpunkt ist in Blattwerkzeug keine Benutzer Authentisierung, Authentifizierung und Autorisierung implementiert. Dies hat zur Folge, dass aktuelle Nutzer ihre erstellten Projekte nicht speichern können. Darüber hinaus hat jeder Nutzer Zugriff auf Inhalte, für die er nicht autorisiert ist. Das Adminpanel ist bisher von jedem Nutzer öffentlich zugänglich. Demzufolge kann jeder Benutzer von Blattwerkzeug.de, jegliche Art von Bearbeitung im Adminpanel vornehmen.



3.2 Passwort Hashing

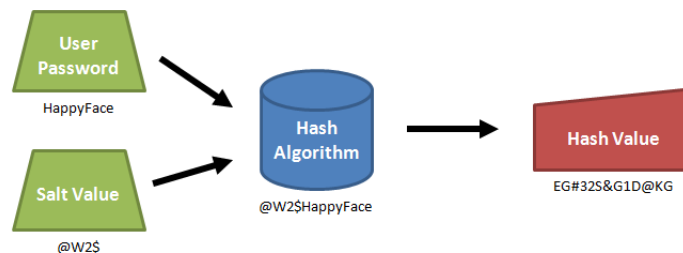
Das Problem was sich ergibt wenn eine Software mit Nutzerdaten administriert wird ist das speichern der Passwörter. Denn sollten die Daten der Nutzer im Klartext in der Datenbank gespeichert werden und ein Angreifer erlangt Zugriff auf die Datenbank, so ist es für ihn ein leichtes weitere Konten der Nutzer zu infiltrieren.

An diesem Punkt kommt das Hashen von Passwörtern zum Einsatz. Passwort Hashing soll dem Nutzer Sicherheit gewährleisten und es einem Angreifer nicht möglich machen mit erlangten Daten weitere Konten der Nutzer zu infiltrieren. Dabei wird aus einem Passwort ein Hash generiert, dieser Hash macht es einem unmöglich, das Passwort wiederherzustellen. Um ein gehashtes Passwort zu erhalten, muss ein Hashing Algorithmus auf das jeweilige Klartext Passwort angewendet werden.

Mittlerweile gibt es verschiedene Hash-Funktionen, von denen manche als nicht mehr sicher gelten. Bestimmte Menschen haben es sich zur Aufgabe gemacht sogenannte Rainbowtables zu erstellen, in denen Hashes mit dazugehörigem Klartext Passwort stehen. Weshalb MD5 und SHA zwei der bekanntesten Hash-Funktionen, seit geraumer Zeit nicht mehr zum Passwort hashen verwendet werden.



Aus diesem Grund werden sogenannte Salts, zufällig generierte Zeichenketten, an das Passwort angehängt und darauf folgend die Hashfunktion angewandt.



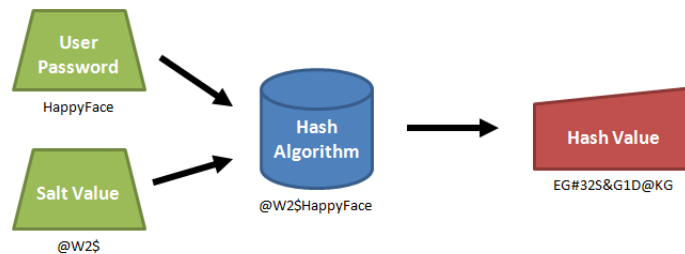
3.3 Sessions

HTTP ein zustandsloses Protokoll, dass sich keine Informationen der jeweiligen Aufrufe zwischenspeichert. Dies ist unpraktisch, da so keine Informationen des Nutzers kurzzeitig gespeichert werden können. Weshalb man auf die Session zurückgreifen kann.

Die Session ist eine serverseitige Daten speicher Möglichkeit. Dabei wird bei der Anfrage von einem Client an den Server ohne Session-ID eine

Session und Session-ID erstellt. Diese Session-ID wird bei der Antwort des Servers mit an den Client ausgeliefert. Ab diesem Punkt wird bei jeder Anfrage vom Client an den Server die Session-ID mit geschickt. Dies kann über einen Cookie oder über die URI erfolgen. Aufgrund dessen kann der Server dem Client Daten aus der jeweiligen Session zur Verfügung stellen.

will be replaced by useful graphic



3.4 JSON Web Token

"JSON Web Token sind auf JSON basierende RFC 7519 genormte Access-Token." - Zitat Wikipedia, muss noch überarbeitet werden

Diese Tokens werden zur eindeutigen Identifizierung von Nutzern verwendet und können die Session ersetzen. Dabei ist es bei einem JSON Web Token nicht von nöten die Daten auf dem Server zu speichern. Dies hat zur Folge, dass sich nicht um den Speicher gekümmert werden muss.

Ein JSON Web Token besteht aus Header, Payload und Signatur. Dabei ist der Header und die Payload jeweils ein JSON Objekt.

3.4.1 Header

typ Der typ Claim beschreibt den Mediatypen des JWT, dieser wiederum teilt dem Client oder Server mit um welche Art von Medium an Daten es sich handelt.

alg Der alg Claim beschreibt die Verschlüsselungsmethode

cty Der cty Claim wird benötigt wenn der Payload des JSON Web Token ebenfalls ein JWT ist

IMAGE

3.4.2 Payload

Die Payload beinhalteten Schlüssel-Wert Paare werden Claims genannt. Dabei handelt es sich um ein JSON Objekt, bei dem bestimmte Schlüssel des Objektes bereits reserviert sind. Diese nennen sich registrierte Claims. Außerdem gibt es öffentliche und private Claims. Hierbei wird zwischen öffentlichen und privaten differenziert.

Beispiel registrierter Claims

- iss** Der iss Claim steht für den Austeller des Tokens, beispielsweise eine Domain.
- sub** Der sub Claim definiert für wen oder was diese Claims getätigt werden sollen.
- aud** Der aud Claim wird genutzt um den Zugriff auf das Token auf eine bestimmte Domäne zu beschränken.
- exp** Der exp Claim kennzeichnet den JWT mit einem Ablaufdatum.
- nbf** Der nbf Claim bestimmt ab welchem Datum der JWT gültig ist.
- iat** Der iat Claim sagt aus wann dieser Claim erstellt wurde.

Beispiel öffentlicher Claims

IMAGE

Beispiel privater Claims

IMAGE

3.4.3 Signatur

Um die Signatur zu erhalten muss die Payload und der Header Base64 kodiert werden. Außerdem müssen diese beiden kodierten Zeichenfolgen mit einem Punkt als Trennzeichen verknüpft werden. Darauf folgend wird eine Hashfunktion auf das jeweilige Ergebnis mit zusätzlich sicherer Zeichenfolge als Parameter angewandt. An dieser Stelle kann festgestellt werden ob der JWT verändert wurde.

3.4.4 Zusammengesetztes Token

Schlussendlich ergibt sich der JWT aus kodierten Header, kodierten Payload und der Signatur. Dabei steht der Header am Anfang. Darauf folgend mit einem Punkt getrennt die Payload und zum Schluss die Signatur, ebenfalls mit einem Punkt getrennt.

IMAGE

3.5 Ruby on Rails

Ruby on Rails ein quelloffenes Webframework für die Programmiersprache Ruby. Das Webframework nutzt das MVC Muster und stellt bereits ein sehr umfangreiches CLI zur Verfügung. Mittels des generate Werkzeugs kann beispielsweise Model, View und Controller erstellt werden. Jeder dieser Komponenten wird automatisch in die erstellte Rails Anwendung eingebunden. Außerdem stellt Rails eine umfangreiche Test-Architektur und einen Service zum versenden von Mails. Dabei kann der Inhalt der E-Mail im Textformat oder als HTML versendet werden.

3.5.1 Controller

Der Controller dient hierbei zur Kapselung von bestimmten Prozessen. Jede Route verweist in irgendeiner Weise auf eine Controller Funktion. In der der jeweiligen Controller Funktion wird dann meistens mit einem Model interagiert. Es wird beispielsweise eine Benutzerberechtigung abgefragt und individuell auf die Berechtigung reagiert. Um auf die jeweilige Berechtigung zu reagieren gibt es mehrere Möglichkeiten. Eine der Möglichkeiten wäre, direkt ein View Template auf dem Server zu rendern und an den Client auszuliefern. Eine andere Möglichkeit wäre ein JSON Objekt zurück zu geben und darauf mit dem Client zu agieren.

3.5.2 Model

Das Model spielt während dieser Thesis eine große Rolle. Es stellt jeweils eine Datenbanktabelle dar. Die Attribute des Models sind jeweilige Spalten der Datenbanktabelle. Jedes Model kann zusätzliche Funktionen beinhalten, die direkt auf den jeweiligen Datenbankeintrag angewandt werden kann. Außerdem bietet Rails die Möglichkeit die Beziehungen zwischen Datenbanktabellen direkt in den Modellen festzulegen.

3.5.3 View

Die View stellt in Rails die Möglichkeit HTML Template auf dem Server zu rendern. Dabei kann beim rendern das HTML Template dynamisch verändert werden. Da diese Komponente während dieser Thesis keine Rolle gespielt hat, wird diese nicht weiter erläutert.

3.6 Zusammenfassung

Schlussendlich wird über die Route auf den jeweiligen Controller zugegriffen. Dieser fragt in den meisten Fällen nach einem bestimmten Eintrag eines Models. Darauf folgend wird mit dem Ergebnis der Anfrage interagiert. Es werden Veränderungen oder abfragen bestimmter Daten getätigt. Danach wird ein Ergebnis dem Client ausgeliefert.



3.7 Angular

Angular ist ein TypeScript basiertes Front-End Webframework, dass in vielen Fällen für SPAs verwendet wird. Das Webframework bietet verschiedene Möglichkeiten dynamisch Daten und Elemente des DOMs zu manipulieren oder auszutauschen.

Hierbei bietet Angular die Möglichkeit HTML, CSS und TypeScript in Komponenten zu kapseln. Das bedeutet, dass jede Komponente unabhängig von einer anderen Komponente arbeiten kann.

Zur Kommunikation mit einem Server und/oder zum Datenaustausch zwischen unterschiedlichen Komponenten wird meistens ein Service verwendet. Diese Services werden beim Laden der Module instanziiert und dem Konstruktor der Komponente als instanziiertes Objekt übergeben.

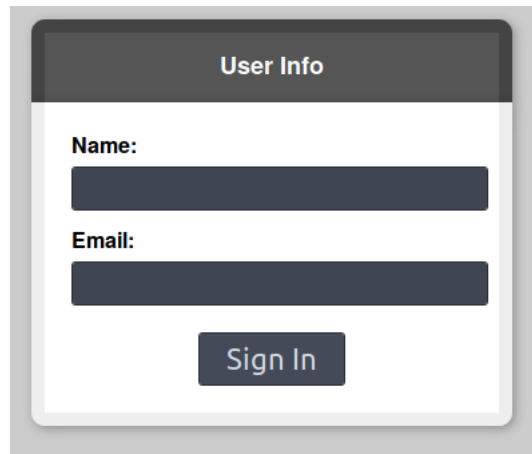
Zusätzlich bietet Angular außerdem die Möglichkeit eigene Module zu erstellen in denen dann beispielsweise Services und Komponenten zusätzlich abgekapselt werden können. Einer der Vorteile die Angular gegenüber anderen JavaScript Frameworks hat, sind die bereits von Angular mitgelieferten Module.



3.8 Omniauth

Omniauth ist eine quelloffene Library für Ruby on Rails und ermöglicht einem, eine Anmeldung mittels unterschiedlicher Anbieter. Hierbei werden bereits viele Funktionen von Omniauth selber übernommen. Sobald der Nutzer sich bei dem jeweiligen Anbieter angemeldet hat, wird die Antwort des jeweiligen Anbieters automatisch über die von Omniauth festgelegte Route verarbeitet. Jedoch muss vorher das spezifische Gem des Anbieters für Omniauth installiert werden.

Omniauth selber verfügt nur über die Developer Strategie, diese ermöglicht eine Anmeldung ohne spezifische Überprüfung der angegebenen Daten. Das hat zur Folge, dass diese Art von Anmeldung auf keinen Fall im Produktiv System vorhanden sein darf. Die Developer Strategie erzeugt automatisch ein Formular zur Anmeldung eines Nutzers.



Die Möglichkeit dieses Formular zu deaktivieren und sich ausschließlich über ein API mit der Developer Strategie anzumelden ist nicht gegeben.

3.8.1 Omniauth Identity

Omniauth Identity ist eine Library zur Erweiterung von Omniauth. Mit Omniauth Identity ist intern ein Anbieter gegeben mit dem es möglich ist, sich zusätzlich mit einem Passwort zu registrieren und anzumelden. Ebenso wie die Developer Strategy, bietet auch diese Library die Möglichkeit ein vorgefertigtes Formular für Anmeldung und Registrierung zu erstellen.

Jedoch ist es bei dieser Library optional und eine ausschließliche Kommunikation über ein API ist möglich.