

HOCHSCHULE HARZ  
HOCHSCHULE FÜR ANGEWANDTE  
WISSENSCHAFTEN (FH)

WEBPROGRAMMIERUNG MIT NODE.JS UND ANGULAR

---

Gleim.net

---

*Autoren:*

Luisa DREWS

Hagen HASENBALG

Christian SCHREBE

*Dozent:*

Prof. Jürgen SINGER,

PH.D.

31. Januar 2016

# Inhaltsverzeichnis

<b>1</b>	<b>Beschreibung</b>	<b>2</b>
<b>2</b>	<b>Entwurf</b>	<b>2</b>
2.1	Profile . . . . .	2
2.2	Freundschaften . . . . .	2
2.3	Chat . . . . .	2
2.4	Ex- und Import . . . . .	3
<b>3</b>	<b>Technologiewahl</b>	<b>3</b>
3.1	Node.js . . . . .	3
3.2	MongoDB . . . . .	3
3.3	Hapi.js . . . . .	3
3.4	Frame . . . . .	3
3.5	Angular 2 . . . . .	3
<b>4</b>	<b>Probleme</b>	<b>4</b>
4.1	Callbacks . . . . .	4
4.2	MongoDB und Treiber . . . . .	4
4.3	Beta-Version von Angular 2 . . . . .	4
<b>5</b>	<b>Anwendung Gleimnet</b>	<b>4</b>
5.1	Abhängigkeiten . . . . .	4
5.2	Installation . . . . .	5
5.3	Webanwendung . . . . .	5
5.3.1	Administration . . . . .	5
<b>6</b>	<b>API</b>	<b>5</b>
<b>7</b>	<b>Zusammenfassung</b>	<b>5</b>

# **1 Beschreibung**

Das GleimNet ist ein soziales Netzwerk, welches im Gleimhaus in Halberstadt eingesetzt werden soll, um im Rahmen von Workshops dazu genutzt werden soll, Schülern Medienkompetenz zu vermitteln. Dafür agieren die Schüler in der Rolle einer historischen Person zur Zeit von Johann Wilhelm Ludwig Gleim. Die erste Version von GleimNet basiert auf Diaspora, das sich aber in der Praxis als ungeeignet erwies.

Im GleimNet sind folgende Funktionen umgesetzt:

## **2 Entwurf**

### **2.1 Profile**

Jeder Nutzer kann sich mit einem Nutzernamen und dazugehörigem Passwort anmelden. Das Profil eines jeden Nutzers umfasst seinen Vor- und Nachnamen, sein Geburtsdatum, sein Profilbild, seine Beschreibung und seine Freundesliste. Der Nachname kann außerdem als Nutzername dienen. Auf der Profilseite des Nutzers befindet sich seine Timeline.

Die Timeline enthält Textpostings, Bilder und Links, welche der Nutzer selbst, oder seine Freunde, erstellen können. Ebenso können die Beiträge von Nutzern kommentiert werden.

### **2.2 Freundschaften**

Zwischen den Nutzern können Freundschaften geschlossen werden, in dem ein Nutzer einem anderen eine Freundschaftsanfrage stellt und derjenige, der sie erhält, diese beantwortet. Bestätigte Freunde sind in der Freundesliste zu finden.

### **2.3 Chat**

Nutzer können mit einem einzelnen anderen Nutzer oder einer Gruppe von Nutzern Nachrichten austauschen. Diese Nachrichten können aus Text, Bildern und Links bestehen.

## **2.4 Ex- und Import**

Die Mitarbeitern des Gleimhaus können den aktuellen Zustand des Netzwerks exportieren und speichern um diese Datei zu einem späteren Zeitpunkt wieder importieren können.

# **3 Technologiewahl**

## **3.1 Node.js**

## **3.2 MongoDB**

## **3.3 Hapi.js**

Hapi.js ist wie Express ein Webserver-Framework auf Basis von Node.js. Die Stärken von diesem Framework ist, es, dass es die Programmierung sehr modular erfolgt. Logische zusammenpassende Routen werden als ein Modul in Hapi.js registriert. So ist es zum Beispiel möglich durch ein Authentifizierungsmodul die An- und Abmeldung ein und auszuschalten.

## **3.4 Frame**

Frame(<https://github.com/jedireza/frame>) stellt für die Entwicklung einer API eine Gute Struktur bereit. Zudem benutzt es als Basis das Webserver Framework Hapi.js. für die Modellerung der Datenbankmodels wird die bei MongoDB meist verwendet Toolset Mongoose verwendet. In diesem Fall dass Implementierung in hapi.js.

Der Hauptgrund für die Entscheidung für Frame war das es eine funktionierende Kombination von verschiedenen node-Modulen bietet.

## **3.5 Angular 2**

Zu Projektbeginn war Angular 2 noch im Alphastadium und es war noch nicht klar ob dieses oder die Version eins zum Einsatz kommen sollte doch sprachen einige Argumente für das Nutzen von Angular 2:

- nutzen von Typescript
- damit zusammen hängende statische Typisierung
- einfacher Anwendung im Gegensatz zu Angular 1.x
- Code lässt sich einfacher strukturieren z.B. durch Module und Klassen

## **4 Probleme**

### **4.1 Callbacks**

Callbacks sind die Hölle!

### **4.2 MongoDB und Treiber**

Der offizielle MongoDB Treiber unterstützt viele sinnvolle Aktionen, wie z.B. das Ersetzen von einer Referenz auf ein Object durch das Object selbst. Leider unterstützt der Objectmapper Mongoose diese zum Teil nicht, sodass diese durch vermehrte Abfragen kombiniert mit map Verfahren und vielen verschachtelten Callbacks ersetzt werden mussten.

### **4.3 Beta-Version von Angular 2**

Die am Anfang benutzte Version von Angular2-beta0 hatte ein paar Fehler, die später gefixt wurden. Nur die Fehler zu finden, wenn diese von fehlerhafter Software entstehen, ist nicht sehr einfach.

## **5 Anwendung Gleimnet**

### **5.1 Abhängigkeiten**

Auf dem System sollen mindestens NodeJS 5.5.0 und MongoDB 3.2.0 installiert sein.

## 5.2 Installation

1. Mongo DB starten
2. npm install
3. node setup.js
4. npm start

Jetzt können entweder neue User über **node create-user.js** oder über das Admin-Panel über **/admin** im Webbrowser aufgerufen werden.

## 5.3 Webanwendung

### 5.3.1 Administration

Die Administration wird über **/admin** aufgerufen. Dort meldet sich der Administrator mittel vorher erstellten Account an. Dies wird mittel Cookies realisiert.

Im Admin-Panel können neue Gruppen aus einer Initialkonfiguration erstellen werden, die aktuelle Sitzung gespeichert werden oder vorherige Sitzungen geladen werden.

Diese Konfigurationen werden als ExtendedJSON(EJSON) gespeichert, damit die von MongoDB gehaltenen Typen wieder hergestellt werden können.

## 6 API

Die Dokumentation zur API ist im separaten pdf zu finden oder bei laufenden Gleimnet unter **/api-dokumentation/index.html** zu finden.

## 7 Zusammenfassung

## Literatur