

TECHNISCHE UNIVERSITÄT BERLIN

PROJEKT AAL

APPLIKATIONSGRUPPE

DOKUMENTATION

Tom Nick
Jonathan Seilkopf
Niklas Gebauer
Maximilian Bachl
Tom Lehmann

5. März 2014



Inhaltsverzeichnis

1	Entwicklerhandbuch	2
1.1	Installation	2
1.2	Projektstruktur	2
1.2.1	Allgemeiner Aufbau	2
1.2.2	Frontend	2
1.3	Erweiterung	3
2	Nutzeranleitung	4
3	Projektbericht	5

Kapitel 1

Entwicklerhandbuch

1.1 Installation

Für die vollständige Lauffähigkeit unserer finalen Abgabe, müssen folgende Programme installiert sein:

- Play
- node.js
- Java 1.7
- Google Chrome

Zum Starten des Projekts muss zunächst das Play-Backend gestartet werden. Das kann getan werden, indem aus dem Projektverzeichnis heraus Play mit `play run` via Konsole gestartet wird. Nachdem der Server fertig geladen hat, muss die Seite einmalig über die Adresse `http://localhost:9000` gestartet werden. Bei diesem Aufruf werden die Jiac-Agenten initialisiert und gestartet. Wird die Seite mehrmals über diese URL geladen, werden die Jiac-Agenten mehrfach gestartet. Das kann zu undefiniertem Verhalten führen und sollte deshalb vermieden werden. Nach dem ersten Aufruf, wechselt der Status zu `http://localhost:9000/index.html#/nouser`. Ab sofort reagiert die Wall auf einkommende Nachrichten und ändert ihren Status selbstständig.

1.2 Projektstruktur

1.2.1 Allgemeiner Aufbau

Aus diversen Gründen haben wir uns dazu entschieden das Frontend mit dem, von Google entwickelten, Javascript-Framework AngularJS¹ zu entwickeln. Die komplette Frontendimplementierung befindet sich im Unterordner `public/angular/app`. Die Widgets haben wir als Angular-Directives implementiert und diese befinden sich im `scripts/directives`-Ordner. Allgemeine Funktionen, welche die gesamte Applikation beziehungsweise den gerade relevanten Teil der Applikation betreffen, werden in den Controllern realisiert. Für häufig genutzte und ausgliederbare Funktionalität, benutzen wir die Services. Ein weiterer zentraler Bestandteil unserer Applikation ist der AngularUI Router² welcher für die Anzeige und den Wechsel der einzelnen Zustände zuständig ist. Sämtliche visuell relevanten Codeteile befinden sich in dem Unterordner `views`.

Unser Backend-Code ist im `app`-Verzeichnis abgelegt. Die Aufgabe des Backends besteht im wesentlichen darin, sich um die Kommunikation mit anderen Gruppen des Projekts via Jiac zu kümmern und das Frontend mit Daten zu versorgen. Weiterhin stellt es der Wallapplikation sowie den Mobilgeräten, welche zur Bedienung ebenjener verwendet werden die Websockets als Kommunikationskanal zur Verfügung.

1.2.2 Frontend

Controllers

AuthCtrl Der Auth-Controller ist aktiv, wenn sich die Applikation in einem für die Nutzererkennung relevanten Zustand befindet. Also wenn sie anzeigt, dass gerade eine Erkennung durchgeführt wird oder ein Nutzer als bekannt oder unbekannt identifiziert wurde. Er stellt unter Anderem die Funktionen `startTraining` sowie `recognizeAgain` bereit, welche über Buttonklicks aufgerufen werden können und die Einleitung eines Trainings-

¹<http://angularjs.org/>

²<http://github.com/angular-ui/ui-router>

oder Erkennungsprozesses durch das Backend initiieren. Weiterhin stellt er die, für die Anzeige des QR-Codes, relevanten Daten zur Verfügung.

MainCtrl Der Main-Controller ist in unserer Applikation praktisch der root-Controller. Er wird beim Start als erstes geladen, empfängt **ADD_USER** und **REMOVE_USER** Nachrichten unseres Backends und kümmert sich um die korrekte Zustandsänderung der Applikation.

MobileCtrl Der Mobile-Controller ist auf allen, sich auf der Mobilseite **http://localhost:9000/index.html#/mobile** befindenden, Geräten aktiv. Er wartet auf Nachrichten von der Wall, mit der er gepaired ist und initiiert Zustandsänderungen auf der Angularinstanz, welche auf dem Mobilgerät aktiv ist. Weiterhin sendet er, sofern das **modal**-Objekt verändert wurde, das neue Objekt an die Wall, sodass sich die Nutzereingaben auf dem Smartphone direkt im, auf der Wall eingeblendeten, Modal-Fenster verfolgen lassen. Initial befindet das Mobilgerät im Zustand **wrapper.mobile.navigation**. Das bedeutet zum Einen, dass sowohl der Main- als auch der Mobil-Controller aktiv sind und zum Anderen, dass in der Subview der **views/mobile.html** die **views/-widgets/mobile/mobile.navigation.html** geladen wird. Diese bietet direkten Zugriff auf die Core-Features der Applikation vom Mobilgerät aus. Das heißt, man kann ohne auf der Wall navigieren zu müssen, Funktionen wie „Facebook-Login“ oder „Add Calendar Entry“ ausführen.

ModalCtrls Die Modal-Controller sind jeweils aktiv, sobald auf der Wall das entsprechende Eingabemodal für die angeforderte Funktionalität angezeigt wird. Sie kommunizieren mit dem Mobilgerät über Websockets und initiieren dadurch gewünschte Zustandsänderungen im **MobileCtrl** auf dem Mobilgerät. Die gesamte Funktionalität des **ModalSocialCtrl**, ist in der finalen Abgabe jedoch nicht erreichbar, da das Posten von anderen Gruppen nicht unterstützt wurde.

SettingsCtrl

1.3 Erweiterung

Kapitel 2

Nutzeranleitung

Kapitel 3

Projektbericht