

WONDER GUI

Danny Koppenhagen

Hochschule für Telekommunikation Leipzig

Danny.Koppenhagen@hft-leipzig.de

16. Februar 2015

Übersicht

Anforderungen

Umsetzung

Verwendete Frameworks und Libraries

Modulares Konzept

Kontaktmanagement

Vorführung

Probleme

Fazit/Ausblick

Anforderungen

Aufgabe

Erstellung eines Frontends basierend auf dem WONDER API als eine Alternative für das bereits existierende Frontend.

Anforderungen

GUI Elemente und Funktionalitäten für:

- ▶ Login/Logout an den Domains (NodeJS/IMS) ✓

Anforderungen

GUI Elemente und Funktionalitäten für:

- ▶ Login/Logout an den Domains (NodeJS/IMS) ✓
- ▶ Call Funktionen
 - ▶ Call zu einer beliebigen URL ✓
 - ▶ Annahme eines eingehenden Calls ✓

Anforderungen

GUI Elemente und Funktionalitäten für:

- ▶ Login/Logout an den Domains (NodeJS/IMS) ✓
- ▶ Call Funktionen
 - ▶ Call zu einer beliebigen URL ✓
 - ▶ Annahme eines eingehenden Calls ✓
- ▶ Chat ✓

Anforderungen

GUI Elemente und Funktionalitäten für:

- ▶ Login/Logout an den Domains (NodeJS/IMS) ✓
- ▶ Call Funktionen
 - ▶ Call zu einer beliebigen URL ✓
 - ▶ Annahme eines eingehenden Calls ✓
- ▶ Chat ✓
- ▶ File-Transfer 1/2

Anforderungen

GUI Elemente und Funktionalitäten für:

- ▶ Login/Logout an den Domains (NodeJS/IMS) ✓
- ▶ Call Funktionen
 - ▶ Call zu einer beliebigen URL ✓
 - ▶ Annahme eines eingehenden Calls ✓
- ▶ Chat ✓
- ▶ File-Transfer 1/2
- ▶ Kontaktliste

Anforderungen

GUI Elemente und Funktionalitäten für:

- ▶ Login/Logout an den Domains (NodeJS/IMS) ✓
- ▶ Call Funktionen
 - ▶ Call zu einer beliebigen URL ✓
 - ▶ Annahme eines eingehenden Calls ✓
- ▶ Chat ✓
- ▶ File-Transfer 1/2
- ▶ Kontaktliste
 - ▶ Add ✓, Modify ✗, Remove ✗

Anforderungen

GUI Elemente und Funktionalitäten für:

- ▶ Login/Logout an den Domains (NodeJS/IMS) ✓
- ▶ Call Funktionen
 - ▶ Call zu einer beliebigen URL ✓
 - ▶ Annahme eines eingehenden Calls ✓
- ▶ Chat ✓
- ▶ File-Transfer 1/2
- ▶ Kontaktliste
 - ▶ Add ✓, Modify ✗, Remove ✗
 - ▶ Direktes Wählen aus der Kontaktliste ✓

Anforderungen

GUI Elemente und Funktionalitäten für:

- ▶ Login/Logout an den Domains (NodeJS/IMS) ✓
- ▶ Call Funktionen
 - ▶ Call zu einer beliebigen URL ✓
 - ▶ Annahme eines eingehenden Calls ✓
- ▶ Chat ✓
- ▶ File-Transfer 1/2
- ▶ Kontaktliste
 - ▶ Add ✓, Modify ✗, Remove ✗
 - ▶ Direktes Wählen aus der Kontaktliste ✓
 - ▶ Presence der Kontakte (online/offline) 1/2

Verwendete Frameworks und Libraries

- ▶ jQuery[1]
 - ▶ Javascript Bibliothek

Verwendete Frameworks und Libraries

- ▶ jQuery[1]
 - ▶ Javascript Bibliothek
- ▶ Bootstrap[2]
 - ▶ weit verbreitetes HTML/CSS/Javascript Framework
 - ▶ optimiert für Responsive Layouts

Verwendete Frameworks und Libraries

- ▶ jQuery[1]
 - ▶ Javascript Bibliothek
- ▶ Bootstrap[2]
 - ▶ weit verbreitetes HTML/CSS/Javascript Framework
 - ▶ optimiert für Responsive Layouts
- ▶ Summernote[3]
 - ▶ grafischer Editor für Textfelder

Verwendete Frameworks und Libraries

- ▶ jQuery[1]
 - ▶ Javascript Bibliothek
- ▶ Bootstrap[2]
 - ▶ weit verbreitetes HTML/CSS/Javascript Framework
 - ▶ optimiert für Responsive Layouts
- ▶ Summernote[3]
 - ▶ grafischer Editor für Textfelder
- ▶ Bootstrap video player[4]
 - ▶ HTML5 Video Player Plugin für Bootstrap/jQuery
 - ▶ Optische Anpassung der HTML5 A/V Elemente

Modulares Konzept

Modulares Konzept

Ziel: GUI Elemente können mit Modulen je nach Status ein- und ausgeblendet werden.

Modulares Konzept

Modulares Konzept

Ziel: GUI Elemente können mit Modulen ja nach Status ein- und ausgeblendet werden.

- ▶ Module
 - ▶ Start
 - ▶ Login
 - ▶ Audio
 - ▶ Video
 - ▶ Chat

Modulares Konzept

vorher (main.js)

```
1 document.getElementById('videoContainer').style.↵  
    visibility = 'visible';  
2 document.getElementById('call').style.visibility = ↵  
    'visible';  
3 document.getElementById('chat').style.visibility = ↵  
    'visible';
```

nachher (main.js)

```
1 showModule.av();  
2 showModule.chat();
```

Kontaktmanagement

- ▶ zunächst Speicherung im Local Storage
- ▶ Erzeugung eindeutiger Kontakt-IDs

Kontaktmanagement

- ▶ zunächst Speicherung im Local Storage
- ▶ Erzeugung eindeutiger Kontakt-IDs

Kontaktformat

```
1  "contacts": {  
2    "U1423592366348": {  
3      "name": "Sanders",  
4      "surname": "Bob",  
5      "logindata": "bob@nodejs.wonder"  
6    },  
7    ...  
8  }
```

Vorführung

Öffnen

Probleme

- ▶ Sehr viel Code, der oft nicht verwendet wird
- ▶ viele unkommentierte Callback-Funktionen
- ▶ Synchronisation der eintretenden Ereignisse

Ausblick

- ▶ Kontaktmanagement mit Datenbank verknüpfen

Ausblick

- ▶ Kontaktmanagement mit Datenbank verknüpfen
- ▶ Presence Anzeige
 - ▶ GUI Elemente bereits vorhanden

Ausblick

- ▶ Kontaktmanagement mit Datenbank verknüpfen
- ▶ Presence Anzeige
 - ▶ GUI Elemente bereits vorhanden
- ▶ File-Transfer Integration
 - ▶ GUI bereits vorbereitet

Ausblick

- ▶ Kontaktmanagement mit Datenbank verknüpfen
- ▶ Presence Anzeige
 - ▶ GUI Elemente bereits vorhanden
- ▶ File-Transfer Integration
 - ▶ GUI bereits vorbereitet
- ▶ Auslagerung des Multiparty Codes

References



jquery.com/ (Letzter Aufruf: 09.02.2014)

Javascript Bibliothek

<http://jquery.com/>



getbootstrap.com (Letzter Aufruf: 09.02.2014)

HTML/CSS/Javascript Framework

<http://getbootstrap.com/>



summernote.org (Letzter Aufruf: 09.02.2014)

WYSIWYG Editor

<http://summernote.org/>



html5-ninja.com (Letzter Aufruf: 09.02.2014)

Bootstrap video player

<http://html5-ninja.com/#/item/Bootstrap-video-player-jQuery-plugin/5>

Vielen Dank
für ihre Aufmerksamkeit!

Fragen?