

web campus

Dokumentation zu Jahntistik-Website Sommer 2016 – Kurs: 60219 – Greiner und Pfeiffer

von: Thomas Hardieck, Sebastian Peiser

Matrikelnummer: 1676209

Semester: SS2016

Fächerkombination: Informationswissenschaft/Musikwissenschaft,

Informationswissenschaft/Medieninformatik

Modul: RZ M03

Fachsemester: 7

Abgabedatum: 14.10.2016

Inhaltsverzeichnis

1 Konzeption 3

2 Realisierung 3

2.1 Home 3

2.2 Überblick 3

2.3 Statistik 4

2.4 Kader 4

2.5 Stadion 4

# 1 Konzeption

## 1.1 Motivation

## 1.2 Design

Jahnrot von <http://html-color-codes.info/colors-from-image/>

Jahnfarben verwendet: rot und weiß, schwarz als gute Ergänzung

Kein Framework, Flexboxen

Eine mainStyle.css für Header und allgemeingültige Regeln, mobileStyle.css für Mobilegeräte, sonst hat jede Seite eigene style.css.

# 2 Realisierung

## 2.1 Home

## 2.2 Überblick

Die Seite „Überblick“ zeigt alle Drittligamannschaften gegen die der SSV Jahn Regensburg in der aktuellen Saison spielen wird. Die Mannschaften werden zum einen in einer Übersichtskarte und zum andern daneben in einer Liste mit zusätzlichen Informationen zu jedem Team dargestellt. Durch einen Klick auf ein Listenelement zoomt die Karte auf das entsprechende Stadion. Möchte der Nutzer wieder auf den ursprünglichen Zoomfaktor zurückwechseln, erreicht er das mit einem Klick auf das Deutschland-Icon in der Karte. Für jede Mannschaft werden die Ergebnisse aus den Spielen gegen den SSV Jahn aus der aktuellen Saison bereitgestellt. Darüber hinaus können auch die Resultate aus den letzten maximal fünf Spielen über den Pfeilbutton eingesehen werden.

Für die Kartendarstellung wurde Leaflet verwendet. Die Informationen über die Mannschaften und Spielstände stammen zum einen aus einer händisch angelegten JSON-Objekt, welches aus der *overview.html* geladen wird, und zum anderen von dem Onlineservice http://www.openligadb.de/. Die Daten werden beim Öffnen der Seite durch die *JahntickerApp.js* geladen und per Team-Id zusammengeführt. Das daraus entstandene Array aus Objekten wird in die *OpponentsListView.js* übergeben und dort mit Hilfe der Templating-Funktion der *underscore.js*-Bibliothek in Listeneinträge verpackt und eingeblendet.

## 2.3 Statistik

Auf der Seite Statistik.html soll es dem User ermöglicht werden die jeweiligen *Fieberkurven*, d.h. den Saisonverlauf, der letzten fünf Saisons zu vergleichen. Hierzu wurden mit Hilfe von Chart.js zwei Charts realisiert. Zum einen ein Line-Chart, mit dem es möglich ist die Punkteausbeute der ausgewählten Saisons pro Spieltag zu vergleichen, zum anderen mehrere Pie-Charts, die je nach im Line-Chart ausgewählter Saison einsehbar sind und das Verhältnis von Siegen, Niederlagen und Unentschieden der jeweiligen Spielzeiten visualisieren sollen.

Zur Realisierung der Seite wurde der Saisonverlauf der letzten fünf Spielzeiten des Jahn Regensburg mit Hilfe der auf der Website *kicker.de* vorhandenen Daten indexiert und in der Html-Datei gespeichert. Hierzu wurde ein script-tag benutzt, in welchem die Punkteausbeute der jeweiligen Spieltage pro Saison in Form von Arrays dokumentiert wurde.

## 2.4 Kader

## 2.5 Stadion

Die Seite „Stadion“ bietet einen Überblick über die Regensburger Continental Arena. Neben einem kurzen Einführungstext stehen dem Nutzer eine Preistabelle, ein Stadionplan und ein Link zum offiziellen Onlineshop des SSV Jahn zu Verfügung. Auf der rechten Seite befinden sich kurze Informationen zu Anfahrt. Auch hier wird vom Leaflet-Service Gebrauch gemacht, um dem Nutzer die geographische Lage des Stadions zu zeigen.

## 2.6 Modals

Mangels eines Frameworks, welches diese Funktion bot, entschieden wir uns, unsere eigenen Modals zu erstellen. Wird auf eine Seite in Modal benötigt muss sowohl die *modalStyle.css* als auch die *Modal.js* in das HTML-Dokument eingebunden werden. Das Dokument braucht danach noch zwei Komponenten: Ein Element, welches durch Anklicken das Modal öffnen soll und das Modal selbst, welches ein zunächst verstecktes div-Element mit der Klasse „modal“ ist. Dieses Element verfügt über ein Kind-div-Element mit der Klasse „modalContainer“, unter dem der Inhalt des Modals eingegeben werden kann. Das anzuklickende Element benötigt außerdem ein Attribut „modal“, welches als Wert eine Referenz auf das Modal setzt, indem das Modal eine zweite Klasse mit genau diesem Wert besitzt. In der *Modal.js* muss nun lediglich das *modalButtons*-Array mit den entsprechenden anklickbaren Elementen befüllt werden.