

IHK Abschluss-Projekt 2017

Jeremy Seipelt

4. Dezember 2017

Für den Beruf des Fachinformatiker der Anwendungsentwicklung

Agentur Website Ein Full-Stack Web-Developer Projekt

Auszubildner:

Jeremy Seipelt

Pichelsdorferstr. 72

13595 Berlin



Ausbildungsbetrieb:

zmb GmbH

Scharnhorstr. 112

11234 Berlin

Inhaltsverzeichnis

1	Vorwort	1
1.1	Beschreibung des Projektes	1
1.2	Ziel des Projektes	1
1.3	Umfeld des Projektes	1
1.4	Projektentscheidung	2
1.5	Beschränkungen des Projektes	2
2	Projekt-Planung	2
2.1	Projektverlauf	2
2.2	Ressourcenplanung	3
2.3	Entwicklungsprozess	3
3	Analysephase	3
3.1	Ist-Analyse	3
3.2	Wirtschaftlichkeit	4
3.2.1	Kosten	4
3.2.2	Amortisation	4
3.3	Anforderungen	4
3.3.1	SSR PoC	5
4	Entwurfsphase	5
4.1	Server	5
4.1.1	CMS	5
4.1.2	HTTP/HTTPS	6
4.1.3	Node	6
4.2	Single Page Application	6
5	Implementierungsphase	7
5.1	Server	7
5.1.1	Aufsetzen des Servers	7
5.1.2	NGINX	7
5.1.3	CMS	7
5.2	SPA	7
6	Abschlussphase	7
7	Dokumentation	7
7.1	Projektdokumentation	7
7.2	Entwicklerdokumentation	8

8	Endstand	8
8.1	Ist/Soll-Zustand	8
8.1.1	Geschwindigkeit	8
8.1.2	Layout	8
8.1.3	SPA	8
8.1.4	SSR	8
8.1.5	CMS	9
8.2	Quo Vadis ?	9
8.3	Fazit für zukünftige Projekte	9
A	Abbildungen	10
A.1	Tabellen	10
A.1.1	Detaillierte Übersicht der Aufgaben	10
A.2	Grafiken	11
A.2.1	Aufbau der Seite	11
A.2.2	Projekt Architektur	11
A.2.3	SSR Node Entwurf	12
A.2.4	SPA Entwurf	12
A.2.5	Website Mockup	13
A.2.6	PageSpeed Insights	14
A.2.7	Pingdom	15
A.2.8	Pingdom	16
A.3	Screenshots	17
A.3.1	Alte Website	17
A.3.2	Entwickler Dokumentation	17
A.3.3	Responsive Design	18
B	Code Beispiele	18
B.1	NGINX Konfiguration Auszug	18

SSR	Server Sided Rendering
SPA	Single Page Application
DOM	Document Object Model
B2B	Business to Business

1 Vorwort

Diese Projektdokumentation ist für die Abschlussprüfung der IHK für den Beruf des Fachinformatikers der Anwendungsentwicklung entstanden und beschreibt Vorgehensweisen des Verfassers.

Ausbildungsbetrieb ist die zmb GmbH, ein KMU mit Standort in Berlin Mitte. Die zmb vertreibt ihre eigen E-Commerce Plattform, genannt CORE, und geht Projektarbeiten im B2B Bereich für alle gängigen Shopsysteme nach. Dabei legt die zmb Wert darauf alle Wünsche des Kunden, von der Bereitstellung eigener API über Onboarding bis hin zu Zahlungsweisen, zu erfüllen.

1.1 Beschreibung des Projektes

Für die neu geplanten Agenturarbeiten der zmb, soll eine neue Internet-Präsenz aufgebaut werden.

Die zmb wird zurzeit nur durch eine Website aus Seiten, die durch das interne CMS System gepflegt werden, präsentiert, sowie auf Partnerseiten.

Die neue Seite wird eine SPA um die Seite für den Kunden modern und schnell zu machen. Der Server muss dafür die SPA bereits kompiliert versenden, da Suchmaschinen sich nur die gesendete HTML Datei ansehen, dafür benötigen wird ein SSR-Setup auf dem Server. Dafür wird im Verlauf des Projekts ein Server gemietet sowie aufgesetzt. Und eine Website erstellt auf der potentielle Kunden mit der zmb in Kontakt treten können.

1.2 Ziel des Projektes

Am Ende des Projekts soll eine SPA entstehen die mithilfe von SSR auch ein Vernünftiges SEO Ranking hinkommt. Dabei sollen namenhafte Geschwindigkeitsindexe die Seite mit maximalen Punkten bewerten. Die Inhalte der SPA werden über ein selbst auf dem Server installiertes CMS-System eingetragen. Am Ende entsteht somit eine Website die auch ohne Hilfe von HTML sowie JS und CSS gepflegt werden kann.

1.3 Umfeld des Projektes

Das Projekt wurde vom COO Simeon Gerodetti beantragt.

Herr Gerodetti ist in der zmb für das operative Geschäft sowie für das Marketing, den Online-Auftritt der zmb zuständig. Des weiteren kümmert er sich um die Partner im B2B Bereich und sorgt für Kundenaufträge und das Rechnungswesen des Unternehmens.

Jovan Gerodetti ist der Softwareentwickler für Frontend sowie der Spezialist in Angular. Er baut neue UIs sowie Schnittstellen für CORE und stellt eine Beratende Funktion für das Projekt dar.

Für das Projekt ist S. Gerodetti für Entscheidungen im Design und J. Gerodetti bei Fragen zur Implementierung zuständig.

Die Abnahme des Projektes wird von S. Gerodetti vollzogen.

1.4 Projektentscheidung

Das Hauptgeschäft besteht aus dem Verkauf und der Entwicklung der Inhouse E-Commerce Lösung 'CORE', welche von 3 Personen gewartet wird. 'Core' wird bisher nur als SaaS Lösung eingesetzt und deshalb liegt der administrative Aufwand bei der zmb. Deswegen wird geplant von der eigenen Lösung wegzugehen und den Kunden, für bereits vorhandene Shopsysteme, Lösungen anzubieten.

Den ersten Schritt in die Richtung der Agenturarbeit geht die zmb mit der zmb.agency Website, dessen Unterbau durch dieses Projekt entsteht.

1.5 Beschränkungen des Projektes

Das Projekt beschränkt sich auf das Aufsetzen sowie implementieren einer kleineren Version der Agentur Website die schon alle technischen Voraussetzungen und ein Bruchteil des Inhaltes enthält. Deswegen wird der Großteil der Layout und CSS Arbeiten die im größeren Umfang ein Teil sind, sowie das Anlegen von Inhalten für die Seite, nicht in die Projektzeit eingerechnet.

Probleme die zur Verzögerung des ganzen Projekts geführt hätten wurden aus der Projektplanung ausgenommen. (Ausführen)

2 Projekt-Planung

Das Projekt wurde auf 70h für die Anforderungen der IHK angesetzt.

2.1 Projektverlauf

Ein Übersicht der Hauptphasen sind der Tabelle 1 zu entnehmen. Eine detaillierte Ansicht der Zeitplanung ist in der A.1.1 auf Seite 10 hinterlegt.

Phase	Dauer in Stunden
Analyse	8
Entwurf	7
Implementierung	23
Testen	13
Dokumentation	8
Gesamt	64

Tabelle 1: Übersicht der Zeitplanung

2.2 Ressourcenplanung

Bei der Auswahl der Ressourcen wurde Wert auf Open Source Lösungen wert gelegt, um Projektkosten möglichst gering zu halten. Die Liste aller Benutzter Software ist unter zu finden.

2.3 Entwicklungsprozess

Die Entwicklung der Website wird per Wasserfall-Modell abgewickelt. Dabei werden die einzelnen Phasen des Projektverlaufs nacheinander durchlaufen. Nach jeder Phaase wird mit S. Gerodetti Rücksprache über den Stand der Dinge gehalten , J. Gerodetti übernimmt dessen Rolle in der Implementierung.

Das Wasserfall-Modell bietet sich für kleine Software-Projekte, die einen klaren Endzustand haben, an. Da die Anforderungen schon niedergeschrieben sind, können diese in der Entwurfsphase mit einbezogen werden.

Eine Agile Entwicklung ist somit nicht Sinnvoll, und würde nur den Aufwand erhöhen.

3 Analysephase

3.1 Ist-Analyse

Die zmb ist nur als Firma für den Vertrieb und Wartung von Shop Systemen bekannt , allen voran ihr eigenes 'CORE'. Die Seite informiert nicht über die Agenturarbeit die die zmb anbietet. Zu Beginn des Projektes steht nur ein gemieteter AWS Server zur Verfügung der eine minimale Version der Linux Distribution Ubuntu als Betriebssystem benutzt.

3.2 Wirtschaftlichkeit

Aufgrund des Umstieges auf Agenturarbeit, muss auch eben diese im Internet beworben werden. Momentan kommen Aufträge nur aus dem Partnernetzwerk und nicht auf dem direkten Weg zur zmb, deswegen ist die neue Internetpräsenz zwangsweise erforderlich.

3.2.1 Kosten

Die Ressourcenplanung besteht aus Personal und Sachkosten.

Für das Projekt werden 3h für Beratung durch einen Senior Developer eingerechnet.

Für die Entwicklung wird auf Open Source Lösungen gesetzt. Eine genau Liste der verwendeten Software auf dem Server wird in Tabelle

Personalkosten		Sachkosten	
Entwickler	$70 \cdot 10\text{€} = 700\text{€}$	AWS Lightsail Server	5€/mtl
COO	$2 \cdot 30\text{€} = 60\text{€}$	Domain	20€/jahr
Beratung durch Seniorentwickler	$3 \cdot 30\text{€} = 90\text{€}$	Software Lizenzen	0€
Gesamt Fix	850€	Gesamt Monatlich	~7€

Tabelle 2: Kostenplanung

3.2.2 Amortisation

Die Seite amortisiert sich durch die Anzahl an Stunden Agenturarbeit die durch sie vergeben wurde. Zur Berechnung der Stunden werden die Kosten aus der Ressourcenplanung genommen. Für Agenturarbeit wird eine Pauschale von 100 €/h genommen.

$$x = 850/100 = 8.5$$

Somit Rechnet sich die Seite bereits nach 9h Agenturarbeit. Das entspricht 9 Kunden im Worst-Case-Szenario die durch die Seite Agenturarbeit buchen.

3.3 Anforderungen

Angefordert wurde eine neue Website.

Diese Website muss höchste Punktzahl in verschiedenen Tempo Benchmarks bringen, um den Besuche zu zeigen das die zmb modernste Technologien beherrscht.

Dabei soll dem Besucher SPA entstehen damit der Benutzer wenn er auf der Seite navigiert nicht den DOM bei jedem Request neu berechnen muss. Zur Erstellung der SPA soll Angular verwendet werden, da es sich hierbei um eine bereits im Unternehmen etabliertes Framework handelt.

Damit Crawler etc unsere Seite vernünftig indexieren können muss die Seite bereits auf dem Server vorgerechnet werden um einen fertigen DOM an den Crawler zu senden. Der Prototyp soll dem Mockup von S. Gerodetti ähneln, das Mockup ist auf Seite 3 einzusehen.

Für den Anfang wird nur das Hero Image, sowie das Kontaktformular auf der Landingpage benötigt.

3.3.1 SSR PoC

Zur Analyse gehörte auch der Beweis das der SSR Prozess funktioniert. Die von Angular bereitgestellte Projekt Repo¹ konnte

4 Entwurfsphase

Die Anforderungen verlangen eine Website, dadurch ist die Wahl der Zielplattform der Webbrowser². Dabei verwenden wir soviel es geht TypeScript³, da es sich hierbei um eine in der Firma etablierte Sprache handelt. TypeScript wird deswegen sowohl auf Client als auch Server Ebene verwendet.

4.1 Server

Der Entwurf der Server Komponenten sind auf Seite x referenziert. Der Server ist ein, von Amazon gebuchter, Lightsail⁴ Server der über ein Linux OS mit Ubuntu als Distribution verfügt.

4.1.1 CMS

Geplant ist ein CMS System auf dem Server (Drupal) mit Datenbank(MYSQL). Dabei soll für jede Komponente der Website, die auf Seite referenziert werden, ein Inhaltstyp erstellt werden.

Die Inhalte sollen über ein REST-Modul in Drupal dann per REST API für die SPA erreichbar sein. Geplant sind vorerst folgende Rest Schnittstellen:

¹<https://github.com/angular/universal-starter>

²Alle Webbrowser ab IE9

³<https://www.typescriptlang.org/>

⁴•

Schnittstelle	Endpunkt
Inhalte einer bestimmten Seite	export/pages/SEITENNAME
Inhalte aller Seiten	export/pages/all
Inhalte einer bestimmten Sektion	/export/paragraphs/ID
Inhalte aller Sektionen	export/paragraphs/all
Inhalte der Kopfzeile	export/headbar

Tabelle 3: REST-API

4.1.2 HTTP/HTTPS

Der Server soll über HTTP/HTTPS erreichbar sein. Der Server muss unterscheiden, ob der Benutzer die Agentur SPA anfragt oder das CMS System. In Anbetracht der weiteren Entwicklung wurde eine zweite Instanz der Webseite erstellt, um Produktions Website von der Entwicklung Website zu trennen. Deswegen werden auf dem Server folgende Routen eingestellt:

Anwendung	Route
Drupal	/drupal
SPA/Agentur Website	/
SPA/Entwicklung	/dev

Tabelle 4: Routen

4.1.3 Node

Um Javascript auf einem Server ausführen zu können, muss ein Node Server vorhanden sein. Der Node Server muss jede auf ihm zeigende Route abfangen und die zugehörige Route der SPA bereitstellen. Hier findet der eigentliche SSR Prozess statt, dafür muss dem Node Server die ganze SPA bereitgestellt werden. Dafür wird das Express Framework verwendet welches HTTP anfragen an den Node Server verarbeitet. Mit

4.2 Single Page Application

Das Framework welches für die SPA verwendet wird ist Angular⁵, welches die zmb seit Jahren produktiv einsetzt. Bei Angular wird der Fokus auf wiederverwendbare Komponenten gesetzt die, wie native DOM-Element, in das HTML eingebunden werden.

Bei der SPA muss dem Layout aus den Anforderungen, auf Seite , entsprechen. Dafür wurde der generelle Seiten-Aufbau in der Grafik ,auf Seite festgelegt. Deswegen werden

⁵<https://angular.io>

Komponenten entworfen, die für einzelne Passage der Website stehen. Um die Rest-API vom CMS konsumieren zu können, benötigen wir noch ein Service⁶. Der Entwurfsplan der SPA ist auf der Seite hinterlegt.

5 Implementierungsphase

5.1 Server

5.1.1 Aufsetzen des Servers

Der gebuchte Lightsail Server wird nur mit einer minimalen Installation ausgeliefert. Als erstes werden die benötigten Programme installiert. Dabei werden die unter Server aufgeführten Programme über den Ubuntu Package Manager apt installiert. Für die einfache Arbeit mit dem Server wurden symbolische Links, auf wichtige Serverressourcen und Programme, in das Home Verzeichnis des Users angelegt, wie auf dem Screenshot auf Seite

5.1.2 NGINX

Der HTTPS Teil des Servers wird von NGINX gesteuert. Nach der Installation des NGINX, muss der NGINX Service gestartet werden. Sobald der NGINX läuft kann der Server konfiguriert werden. Da Drupal und Node auf dem selben Server liegen müssen für beide unterschiedliche Routen in der Konfiguration angelegt werden wie auf Seite 18.

5.1.3 CMS

Drupal ist das einzige Programm welches nicht über Package Manager installiert wurde.

5.2 SPA

6 Abschlussphase

7 Dokumentation

Die Dokumentation besteht aus zwei Teilen einer Projektdokumentation und einer Entwickler Dokumentation.

7.1 Projektdokumentation

Die Projektdokumentation besteht aus einer genauen Beschreibung der für die Ausführung des Projektes benötigten Phasen, sowie Information zur in der Phase benutzten

⁶<https://angular.io/guide/dependency-injection>

Technologie. Die Dokumentation wird mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ geschrieben um ein einheitliches Layout zu gewähren.

7.2 Entwicklerdokumentation

Die Entwicklerdokumentation besteht aus einer automatisch generierten Doku unter Verwendung von Compodoc⁷, welcher alle Informationen aus dem Sourcecode bezieht und diese für Entwickler aufbereitet. Diese Dokumentation ist aus dem Sourcecode für alle Entwickler kompilierbar. Screenshot im Anhang A.3.2 auf Seite 17.

8 Endstand

Das Projekt wurde mit einer funktionstüchtigen Website abgeschlossen.

8.1 Ist/Soll-Zustand

Der Ist-Zustand wird in den folgenden Abschnitten mit den Anforderungen(Soll-Zustand) verglichen.

8.1.1 Geschwindigkeit

Die Website schneidet mit höchsten Punktzahlen bei den Ausgewählten Benchmarks ab. Die Berichte für Pingdom sowie PageSpeed Insight sind auf Seite 14/16. Die Google Audits für Performance konnten nicht im Zeitrahmen des Projekts nicht in den grünen Bereich gebracht werden. Übersicht des Audits ist auf Seite zu sehen.

8.1.2 Layout

Das fertige Layout ist auf 3 verschiedenen Endgeräten auf Seite zu sehen. Die Navigation in der Kopfzeile fehlt, da es noch keine Navigation auf der Seite gibt. Problematisch ist die weiße Überschrift, welche wenn sie die Erde berührt nur schwer zu Lesen ist.

8.1.3 SPA

Die geschriebene SPA ist in der Lage die Seite in gewünschter Form anzuzeigen. Die Screenshots aus dem Layout Teil sind direkt in der SPA entstanden.

8.1.4 SSR

Die Seite wird, wie auf den Screenshots auf Seite X und Y gezeigt, bereits auf dem Server gerendert.

⁷<https://compodoc.github.io/compodoc/>

8.1.5 CMS

Das **CMS!** Drupal ist auf dem Server installiert und ist über eine eigene URL auf der Domain der Agenturseite freigeschaltet. Inhalte werden bereits für die Seite erstellt und gepflegt, wie auf Seite zu sehen ist.

8.2 Quo Vadis ?

Nach Abschluss ist noch eine Menge für die Seite zu tun. Das CMS ist zwar vorhanden besitzt aber kaum Einträge oder Übersetzungen in andere Sprachen. Somit muss noch ein Übersetzung Service in Angular gebaut und Inhalte erstellt werden. Das Bootstrapping der SPA verwirft den DOM zurzeit, deswegen ist der Einbau des PrebootModuls grade an oberster Priorität. Geplant ist auch das verwenden von Service Workers⁸ wie Google Amp⁹.

8.3 Fazit für zukünftige Projekte

⁸<https://developers.google.com/web/fundamentals/primers/service-workers/>

⁹<https://www.ampproject.org/>

A Abbildungen

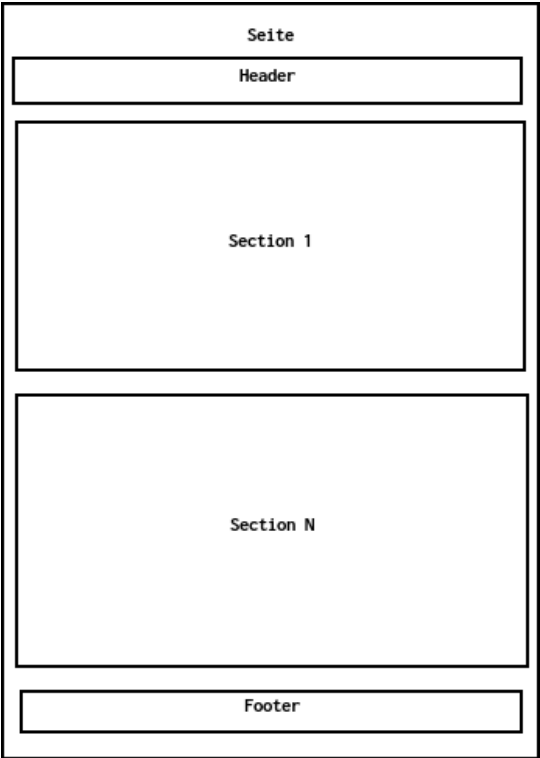
A.1 Tabellen

A.1.1 Detaillierte Übersicht der Aufgaben

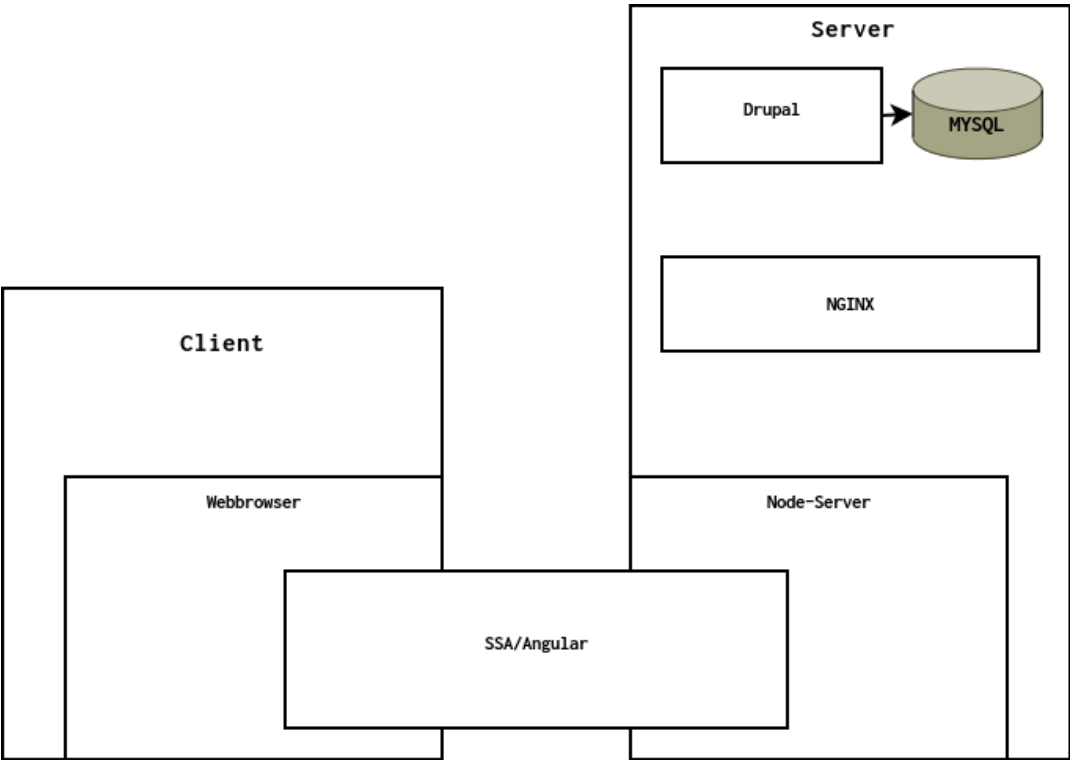
Teilaufgabe	Zeit in Stunden
Analyse	8
Serverseitiges Rendern	4
Belesung	1
PoC anhand einer Implementierung in Angular mit Typescript	2
Anforderungen an den Server	1
Drupal	4
Einarbeitung ins System	2
Benötigte Module	1
Benötigte Inhaltstypen	1
Entwurf	7
SSA	4
Entwurf der Klassen	2
Entwurf der Componenten Templates	2
Entwurf für Drupal	3
Inhaltstypen	1
REST-API	2
Implementierung	28
Webserver	8
Webserver buchen + hochfahren + konfigurieren	1
Nginx installieren + konfigurieren	2
Mysql installieren + konfigurieren	1
PHP/FastCGI installieren + konfigurieren	1
Drupal installieren + konfigurieren	2
Node/PM2 für SSA installieren	1
Drupal	7
Inhaltstypen implementieren	3
REST-API implementieren	2
Content erstellen	2
SSA in Angular mit Typescript	13
Umschreiben der PoC Implementierung als SSA(aus der Analyse)	1
Implementierung der geplanten Klassen	2
Anpassung der Componenten Templates + SASS	5
Main.js für den Node erstellen	2
Compiler einstellen(NGC+TSC+SASS)	1
Taskrunner(Gulp) einstellen	2
Testen	13
Pagespeed testen*	10
Responsiveness testen	3
Dokumentation	8
Projektdokumentation	6
Entwicklerdokumentation	2
Gesamte geplante Projektdauer	64
Maximale Projektdauer	70
Puffer	6
*Der Pagespeed wird sobald die Website auf dem Server ist getestet. Hier werden ggf. Pagespeed Optimierungen implementiert, wenn sie erforderlich sind. (Caching,Minify,Uglyfy,Compression etc)	

A.2 Grafiken

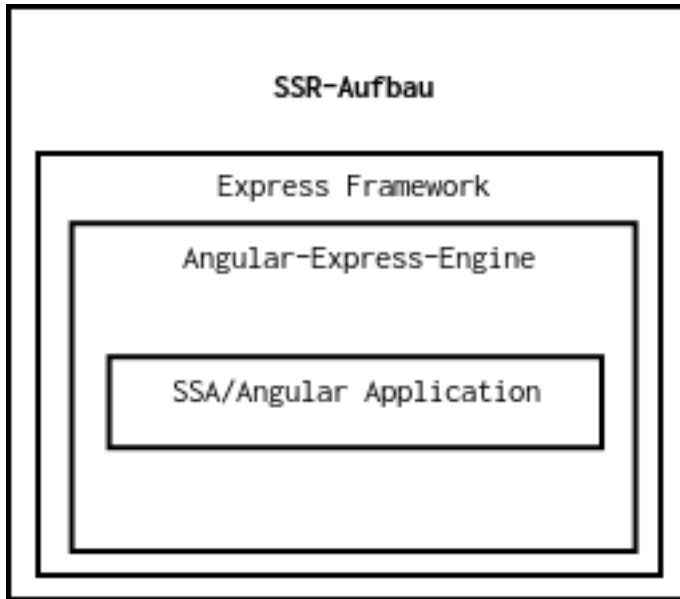
A.2.1 Aufbau der Seite



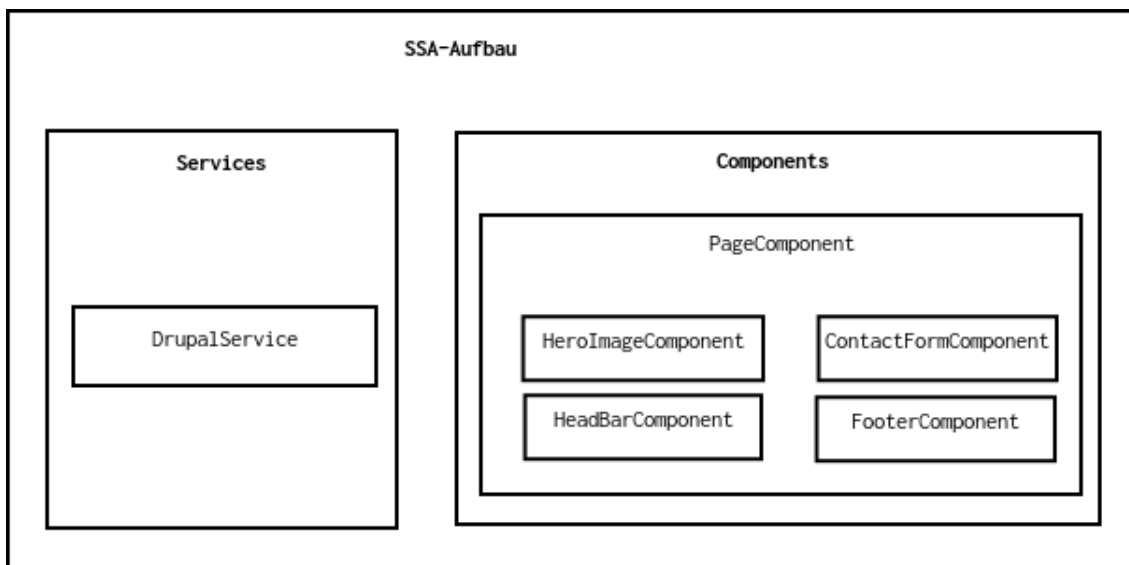
A.2.2 Projekt Architektur



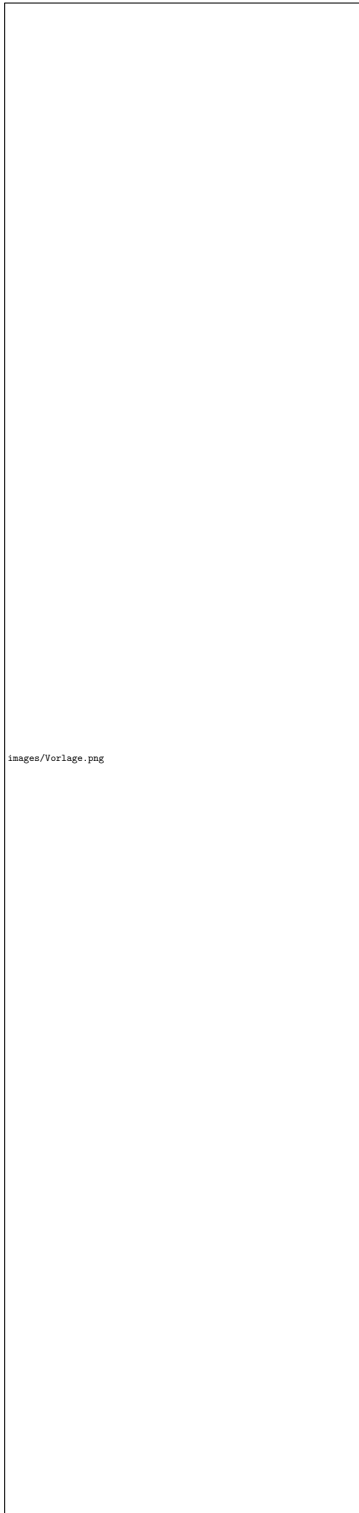
A.2.3 SSR Node Entwurf



A.2.4 SPA Entwurf



A.2.5 Website Mockup



images/Vorlage.png

A.2.6 PageSpeed Insights

PageSpeed Tools > Insights

LEITFÄDEN REFERENZ BEISPIELE SUPPORT

PageSpeed Insights

http://52.29.161.188/agency/

ANALYSIEREN

Mobil

Desktop

Good

100 / 100

Gut gemacht. Auf dieser Seite werden die meisten Best Practices zur Leistung umgesetzt, sodass eine gute Nutzererfahrung gewährleistet sein sollte.

Glückwunsch! Es wurden keine Probleme gefunden.

Antwortzeit des Servers reduzieren

Ihr Server hat schnell geantwortet. [Weitere Informationen zur Optimierung der Serverantwortzeit](#)

Bilder optimieren

Ihre Bilder wurden optimiert. [Weitere Informationen zum Optimieren von Bildern](#)

Browser-Caching nutzen

Sie haben das Browser-Caching aktiviert. [Empfehlungen für das Browser-Caching](#)

CSS reduzieren

Ihre CSS-Ressource wurde reduziert. [Weitere Informationen zum Reduzieren von CSS-Ressourcen](#)

HTML reduzieren

Ihre HTML-Ressource wurde reduziert. [Weitere Informationen zum Reduzieren von HTML-Ressourcen](#)

JavaScript reduzieren

Ihre JavaScript-Ressource wurde reduziert. [Weitere Informationen zum Reduzieren von JavaScript-Ressourcen](#)

JavaScript- und CSS-Ressourcen, die das Rendering blockieren, in Inhalten "above the fold" (ohne Scrollen sichtbar) beseitigen

Sie haben keine Ressourcen, die das Rendering blockieren. [Weitere Informationen zum Entfernen von Ressourcen, die das Rendering blockieren](#)

Komprimierung aktivieren

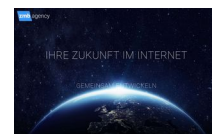
Die Komprimierung ist aktiviert. [Weitere Informationen zum Aktivieren der Komprimierung](#)

Sichtbare Inhalte priorisieren

Die Inhalte "above the fold" (ohne Scrollen sichtbar) wurden ordnungsgemäß priorisiert. [Weitere Informationen zum Priorisieren sichtbarer Inhalte](#)

Zielseiten-Weiterleitungen vermeiden


Auf Ihrer Seite sind keine Weiterleitungen vorhanden. [Weitere Informationen zum Vermeiden von Zielseiten-Weiterleitungen](#)



** Die Ergebnisse werden 30 Sekunden lang im Cache gespeichert. Wenn Sie Änderungen an Ihrer Seite vornehmen, warten Sie 30 Sekunden, bevor Sie den Test erneut durchführen.*

** Mit diesem Test wird überprüft, ob auf einer Seite gängige Best Practices zur Leistung umgesetzt werden. Eine hohe Punktzahl weist auf eine gute Nutzererfahrung hin, ist jedoch keine Garantie dafür. [Weitere Informationen](#).*

A.2.7 Pingdom



[FULL PAGE TEST](#) [DNS HEALTH](#) [LOG IN](#) [SIGN UP](#)

Pingdom Website Speed Test

Enter a URL to test the load time of that page, analyze it and find bottlenecks.

URL

Test from

http://52.29.161.188/agency/


Stockholm, Sweden

START TEST

DOWNLOAD HAR

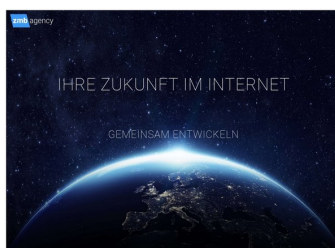
SHARE RESULT

Sign up for **free** to test your site every minute



[SIGN UP FOR FREE](#)

Summary



Performance grade

A 100

Load time

484 ms

Faster than

97 %

of tested sites

Page size

1.5 MB

Requests

6

Tested from

Stockholm

on Nov 8 at 17:54

Performance insights

GRADE	SUGGESTION
A 100	Avoid bad requests
A 100	Leverage browser caching
A 100	Minimize redirects
A 100	Minimize request size
A 100	Remove query strings from static resources
A 100	Serve static content from a cookieless domain
A 100	Specify a cache validator
A 100	Specify a Vary: Accept-Encoding header

Response codes

A.2.8 Pingdom

RESPONSE CODE	RESPONSES
200 OK	6

Content size by content type

CONTENT TYPE	PERCENT	SIZE
Image	66.0 %	1018.98 ...
Script	25.7 %	396.24 KB
Other	8.1 %	124.69 KB
HTML	0.3 %	4.31 KB
Total	100.00 %	1.51 MB

Requests by content type

CONTENT TYPE	PERCENT	REQUESTS
Image	33.3 %	2
Other	33.3 %	2
HTML	16.7 %	1
Script	16.7 %	1
Total	100.00 %	6

Content size by domain

DOMAIN	PERCENT	SIZE
52.29.161.188	100.0 %	1.51 MB
Total	100.00 %	1.51 MB

Requests by domain

DOMAIN	PERCENT	REQUESTS
52.29.161.188	100.0 %	6
Total	100.00 %	6

File requests

Sort by Filter DNS SSL Send Wait Receive Connect

FILE	SIZE	0.0s	0.1s	0.2s	0.3s	0.4s	0.5s
http://52.29.161.188/agency/	4.3 kB						
bundle.js 52.29.161.188/agency/out/	396.2 kB						
AdobeStock_112468496-min%202.jpeg 52.29.161.188/drupal/sites/default/fi...	1.0 MB						
Roboto-Thin.woff2 52.29.161.188/agency/out/fonts/roboto/	61.7 kB						
Roboto-Regular.woff2 52.29.161.188/agency/out/fonts/roboto/	63.0 kB						
logo_1.svg 52.29.161.188/drupal/sites/default/fi...	2.0 kB						
6 requests	1.5 MB	484 ms					

State Colors

The following colors are used in the bars in the waterfall chart to indicate the different stages of a request.

DNS	Web browser is looking up DNS information
SSL	Web browser is performing a SSL handshake
Connect	Web browser is connecting to the server
Send	Web browser is sending data to the server
Wait	Web browser is waiting for data from the server
Receive	Web browser is receiving data from the server

Content Types

The following icons are used to indicate different content types.

About Pingdom

Pingdom offers cost-effective and reliable uptime and performance monitoring for your websites. We use more than 70 global polling locations to test and verify our customers' sites 24/7, all year long. With Pingdom you can monitor your website's uptime, performance, and interactions for a better end-user-experience. Your customers will thank you.

Nobody Likes a Slow Website – We built this Website Speed Test to help you analyze the load speed of your websites and learn how to make them faster. It lets you identify what about a web page is fast, slow, too big, what best practices you're not following, and so on. We have tried to make it useful both to experts and novices alike.

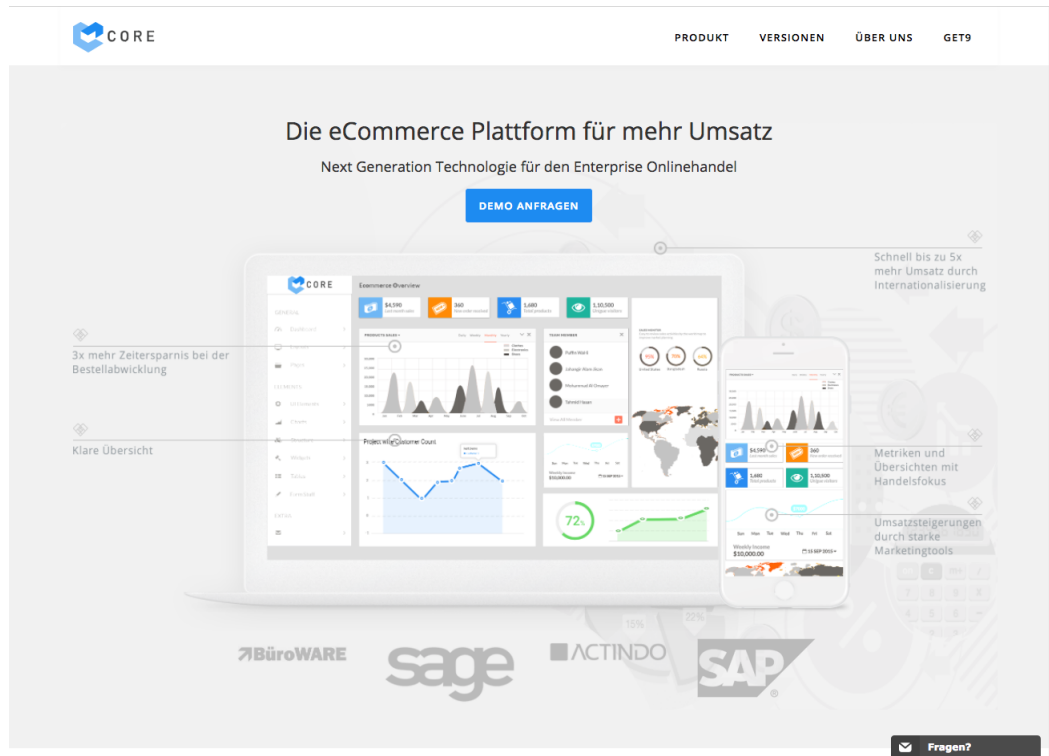
In short, we wanted it to be a easy-to-use tool to help webmasters and web developers everywhere optimize the performance of their websites.

Feature Overview

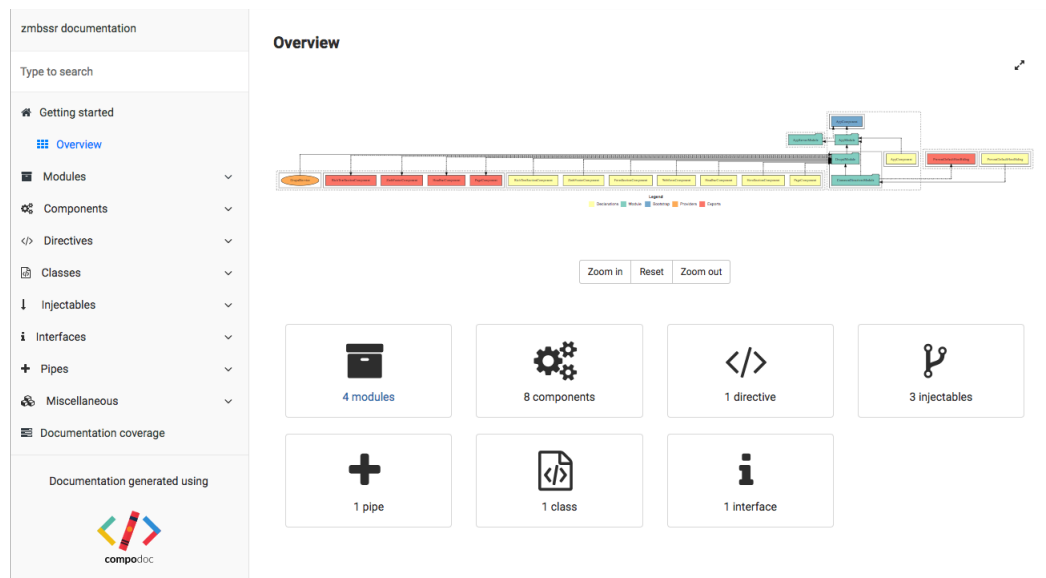
Examine all parts of a web page – View file sizes, load times, and other details about every single element of a web page (HTML,

A.3 Screenshots

A.3.1 Alte Website



A.3.2 Entwickler Dokumentation



A.3.3 Responsive Design



B Code Beispiele

B.1 NGINX Konfiguration Auszug

```
location / {  
    proxy_pass http://localhost:1337/;  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection 'upgrade';  
    proxy_set_header Host $host;  
    proxy_cache_bypass $http_upgrade;  
}  
  
location /drupal/ {  
    index index.php;  
    root /data/www/drupal;  
}
```