YISU Travel

Dokumentation

Üzeyir Yilmaz

16.10.2024

**Bedingungen für den Source Code:**

Angular CLI: 18.2.6

Node: 20.17.0

Package Manager: npm 10.8.2

OS: win32 x64

Package Version

------------------------------------------------------

@angular-devkit/architect 0.1802.6 (cli-only)

@angular-devkit/core 18.2.6 (cli-only)

@angular-devkit/schematics 18.2.6 (cli-only)

@schematics/angular 18.2.6 (cli-only)

Installationen:

ng add @angular/material

* Version: @angular/material@18.2.6

ng add @angular/cdk

* @angular/cdk": "^18.2.6

npm install angular-bootstrap-md

* Version: "bootstrap": "^5.3.3",

npm i ngx-mat-timepicker

* Version: [ngx-mat-timepicker@18.0.0](mailto:ngx-mat-timepicker@18.0.0)

npm install core-js whatwg-fetch

ng generate config browserslist

das einfügen: (Safari fix)  
last 2 Chrome versions  
last 1 Firefox version  
last 2 Edge major versions  
last 2 Safari major versions  
Safari >= 15  
last 2 iOS major versions  
last 2 Android major versions  
Firefox ESR

npm i swiper

"swiper": "^11.1.15",

SSR und SSG aktivieren:

ng add @angular/ssr (Wichtig für SEO, da HTML Dateien statisch geladen werden)

* ng build (alles im Ordner dist/browser hochladen)
* => FALLS EINE index.csr.html generiert wurde nach ng build im dist/browser Ordner, BENNENE ES UM IN index.html und lösche die vorherige index.html.
  + Falls beide (index.csr.html, index.html) generiert werden, dann nur den index.html deployen und den anderen löschen
* **outputHashing** auf **none** und **optimization** auf **true** stellen in der angular.json

Ein Bild, das Text, Screenshot, Software, Design enthält.

Automatisch generierte Beschreibung

Cache Problem verhindern:

KEIN: ng add @angular/pwa 18.2.10 Version verwenden!!!

* Sobald SSR aktiviert ist, werden die Sachen schon statisch geladen. Wir müssen nur noch den .htaccess so schreiben, dass er niemals cache speichert, damit der Browser immer aktuell ist. Cache ist sehr gefährlich. Wie machen wir das? Siehe .htaccess Einstellung.

Weitere Versionen (Bootstrap usw. in der package.json)

Ein Bild, das Text, Screenshot, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Font awesome runterladen (im lokalen Ordner wegen dsgvo und im /public/fontawesome/ packen):

* Version fontawesome-free-6.0.0-beta3-web

**Herunterladen:**

https://use.fontawesome.com/releases/v6.0.0-beta3/fontawesome-free-6.0.0-beta3-web.zip

Wichtige Anmeldedaten:

IONOS: Webhosting

<https://login.ionos.de/?shoplogin=enabled>

264653739

YiSu2016Travel

IONOS: VPS

923432449

Stayhappy1#

FTP FileZilla:

-sftp://u75950685@home511037246.1and1-data.host

u75950685

Y1sU\_Tr4v3l.!79-X

Database:

Y1sU\_Tr4v3l.Datab4se\_87!\_.&

Livechat Integration:

Login: https://1148.3cx.cloud/#/login?next=%2Foffice

Anmeldename & Passwort

info@yisu-travel.de

Y1sU\_Tr4v3l.!79-X

Fonts:

1) Herunterladen z.B.

https://fontawesome.com/download

https://fonts.google.com/selection

2) in /public/fonts google fonts reintun und von fontawesome auch in ein public/fontawesome ordner und all.min.css reintun

3) in styles.css alles rein (chatgpt + dir /b alles laden und dann adden lassen)

/\* Roboto Font Faces \*/

@font-face {

font-family: 'Roboto';

font-weight: 100; /\* Thin \*/

font-style: normal;

src: url('/fonts/Roboto-Thin.ttf') format('truetype'),

url('/fonts/Roboto-ThinItalic.ttf') format('truetype');

}

@font-face {

font-family: 'Roboto';

font-weight: 300; /\* Light \*/

font-style: normal;

src: url('/fonts/Roboto-Light.ttf') format('truetype'),

url('/fonts/Roboto-LightItalic.ttf') format('truetype');

}

@font-face {

font-family: 'Roboto';

font-weight: 400; /\* Regular \*/

font-style: normal;

src: url('/fonts/Roboto-Regular.ttf') format('truetype'),

url('/fonts/Roboto-Italic.ttf') format('truetype');

}

@font-face {

font-family: 'Roboto';

font-weight: 500; /\* Medium \*/

font-style: normal;

src: url('/fonts/Roboto-Medium.ttf') format('truetype'),

url('/fonts/Roboto-MediumItalic.ttf') format('truetype');

}

@font-face {

font-family: 'Roboto';

font-weight: 700; /\* Bold \*/

font-style: normal;

src: url('/fonts/Roboto-Bold.ttf') format('truetype'),

url('/fonts/Roboto-BoldItalic.ttf') format('truetype');

}

@font-face {

font-family: 'Roboto';

font-weight: 900; /\* Black \*/

font-style: normal;

src: url('/fonts/Roboto-Black.ttf') format('truetype'),

url('/fonts/Roboto-BlackItalic.ttf') format('truetype');

}

4) in angular.json fontawesome laden:

"styles": [

"@angular/material/prebuilt-themes/azure-blue.css",

"public/fontawesome/fontawesome-free-6.0.0-beta3-web/css/all.min.css",

"src/styles.css"

],

=> Copy path from content root

**Server Einstellungen und SEO Optimization**

.htaccess (erstellen im Root Ordner der Domain)

**Aktuellste Version, kein http und kein www (kein doppelter Content wegen SEO)**

<IfModule mod\_rewrite.c>

RewriteEngine On

# Umleitung von /startseite/ auf die Hauptseite /

RewriteCond %{REQUEST\_URI} ^/startseite/?$

RewriteRule ^ / [R=301,L]

# Redirect to non-www

RewriteCond %{HTTP\_HOST} ^www\.(.\*)$ [NC]

RewriteRule ^ https://%1%{REQUEST\_URI} [L,R=301]

# HTTPS redirect

RewriteCond %{HTTPS} off

RewriteRule ^ https://%{HTTP\_HOST}%{REQUEST\_URI} [L,R=301]

RewriteBase /

# Ensure index.html is not rewritten

RewriteRule ^index\.html$ - [L]

# Handle Angular routes (SSG and SSR)

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteRule . /index.html [L]

</IfModule>

<IfModule mod\_headers.c>

# Enable necessary headers for PWA and SSR

Header set Cache-Control "no-store, no-cache, must-revalidate"

Header set Pragma "no-cache"

Header set Expires 0

</IfModule>

* Erstelle in der google search console eine **sitemap.xml**, eine **google google080d84130527cbc7.html** und eine robots.txt Datei für den Crawler.

Inhalt:

1. sitemap.xml:

<?xml version="1.0" encoding="UTF-8"?>

<urlset

xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9

http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">

<!-- created with Free Online Sitemap Generator www.xml-sitemaps.com -->

<url>

<loc>https://yisu-travel.de/</loc>

<lastmod>2024-10-05T06:03:48+00:00</lastmod>

</url>

</urlset>

#bearbeitet

<?xml version="1.0" encoding="UTF-8"?>

<urlset

xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9

http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">

<!-- created with Free Online Sitemap Generator www.xml-sitemaps.com -->

<url>

<loc>https://yisu-travel.de/startseite</loc>

<priority>1.0</priority>

<lastmod>2024-10-05T06:03:48+00:00</lastmod>

</url>

<url>

<loc>https://yisu-travel.de/ueber-uns</loc>

<priority>0.8</priority>

</url>

<url>

<loc>https://yisu-travel.de/dienstleistung</loc>

<priority>0.8</priority>

</url>

<url>

<loc>https://yisu-travel.de/kontakt</loc>

<priority>0.7</priority>

</url>

<url>

<loc>https://yisu-travel.de/impressum</loc>

<priority>0.5</priority>

</url>

<url>

<loc>https://yisu-travel.de/datenschutzerklaerung</loc>

<priority>0.5</priority>

</url>

</urlset>

//AKTUELL  
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"

xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9

http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<url>

<loc>https://yisu-travel.de/startseite</loc>

<priority>1.0</priority>

<lastmod>2024-10-05T06:03:48+00:00</lastmod>

<changefreq>daily</changefreq>

</url>

<url>

<loc>https://yisu-travel.de/ueber-uns</loc>

<priority>0.8</priority>

<lastmod>2024-10-05T06:03:48+00:00</lastmod>

<changefreq>monthly</changefreq>

</url>

<url>

<loc>https://yisu-travel.de/dienstleistung</loc>

<priority>0.8</priority>

<lastmod>2024-10-05T06:03:48+00:00</lastmod>

<changefreq>monthly</changefreq>

</url>

<url>

<loc>https://yisu-travel.de/kontakt</loc>

<priority>0.7</priority>

<lastmod>2024-10-05T06:03:48+00:00</lastmod>

<changefreq>monthly</changefreq>

</url>

<url>

<loc>https://yisu-travel.de/impressum</loc>

<priority>0.5</priority>

<lastmod>2024-10-05T06:03:48+00:00</lastmod>

<changefreq>yearly</changefreq>

</url>

<url>

<loc>https://yisu-travel.de/datenschutzerklaerung</loc>

<priority>0.5</priority>

<lastmod>2024-10-05T06:03:48+00:00</lastmod>

<changefreq>yearly</changefreq>

</url>

</urlset>

1. google080d84130527cbc7.html

google-site-verification: google080d84130527cbc7.html

1. robots.txt

# Erlaubt allen Bots den Zugang zur gesamten Website

User-agent: \*

Allow: /

# Blockiere administrative und irrelevante Bereiche

Disallow: /admin/

Disallow: /wp-admin/

Disallow: /cgi-bin/

Disallow: /login

Disallow: /search

# Gebe die Sitemap der Website an, damit Bots die Struktur verstehen

Sitemap: <https://yisu-travel.de/sitemap.xml>

Einen /backend/ Ordner im Root Folder der Domain erstellten und folgende beiden php Dateien einfügen für:

1. send\_mail.php

<?php

// Check if POST request is made

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get the posted form data

$name = htmlspecialchars($\_POST['name'], ENT\_QUOTES, 'UTF-8');

$email = htmlspecialchars($\_POST['email'], ENT\_QUOTES, 'UTF-8');

$subject = htmlspecialchars($\_POST['subject'], ENT\_QUOTES, 'UTF-8');

$message = nl2br(htmlspecialchars($\_POST['message'], ENT\_QUOTES, 'UTF-8'));

// Prepare the email for you

$to = "medi.edi@gmx.de"; // Your IONOS email to receive the contact messages

$email\_subject = "Kontaktformular von YISU Travel: $subject";

$email\_body = "

<!DOCTYPE html>

<html lang='de'>

<head>

<meta charset='UTF-8'>

<meta name='viewport' content='width=device-width, initial-scale=1.0'>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 20px;

background-color: #f7f7f7;

}

.container {

max-width: 600px;

margin: auto;

background: #ffffff;

padding: 30px;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

h2 {

color: #333333;

margin-bottom: 20px;

font-size: 24px;

text-align: left;

}

.info {

font-size: 16px;

color: #333333;

margin-bottom: 10px;

}

.message {

font-size: 16px;

line-height: 1.5;

color: #333333;

margin-top: 15px;

background-color: #ffffff; /\* Hellerer Hintergrund für die Nachricht \*/

border-radius: 4px;

}

.footer {

margin-top: 20px;

font-size: 12px;

color: #777777;

border-top: 1px solid #e0e0e0;

padding-top: 10px;

text-align: center;

}

</style>

</head>

<body>

<div class='container'>

<h2>Neue Nachricht von www.yisu-travel.de</h2>

<div class='info'><strong>Name:</strong> $name</div>

<div class='info'><strong>E-Mail:</strong> $email</div>

<div class='info'><strong>Betreff:</strong> $subject</div>

<div class='message'>

<strong>Nachricht:</strong><br>

$message

</div>

</div>

</body>

</html>

";

// Set the email headers for the main email

$headers = "From: info@yisu-travel.de\r\n";

$headers .= "Reply-To: $email\r\n";

$headers .= "Content-Type: text/html; charset=UTF-8\r\n";

// Send the email to you

if (mail($to, $email\_subject, $email\_body, $headers)) {

// Prepare confirmation email for the user

$user\_email\_subject = "Ihr Kontaktformular an YISU Travel";

$user\_email\_body = "

<!DOCTYPE html>

<html lang='de'>

<head>

<meta charset='UTF-8'>

<meta name='viewport' content='width=device-width, initial-scale=1.0'>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 20px;

background-color: #f7f7f7;

}

.container {

max-width: 600px;

margin: auto;

background: #ffffff;

padding: 30px;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

h2 {

color: #333333;

margin-bottom: 20px;

font-size: 24px;

text-align: left;

}

.info {

font-size: 16px;

color: #333333;

margin-bottom: 10px;

}

.message {

font-size: 16px;

line-height: 1.5;

color: #333333;

margin-top: 15px;

background-color: #ffffff; /\* Hellerer Hintergrund für die Nachricht \*/

border-radius: 4px;

}

.footer {

margin-top: 20px;

font-size: 12px;

color: #777777;

border-top: 1px solid #e0e0e0;

padding-top: 10px;

text-align: center;

}

</style>

</head>

<body>

<div class='container'>

<h2>Vielen Dank für Ihr Kontaktformular an YISU Travel!</h2>

<div class='info'><strong>Name:</strong> $name</div>

<div class='info'><strong>E-Mail:</strong> $email</div>

<div class='info'><strong>Betreff:</strong> $subject</div>

<div class='message'>

<strong>Ihre Nachricht:</strong><br>

$message

</div>

<div class='footer'>

<p>Wir werden uns so schnell wie möglich bei Ihnen melden.</p>

</div>

</div>

</body>

</html>

";

// Set the email headers for the confirmation email

$user\_headers = "From: info@yisu-travel.de\r\n";

$user\_headers .= "Reply-To: info@yisu-travel.de\r\n";

$user\_headers .= "Content-Type: text/html; charset=UTF-8\r\n";

// Send the confirmation email to the user

mail($email, $user\_email\_subject, $user\_email\_body, $user\_headers);

echo json\_encode(["status" => "success", "message" => "E-Mail wurde erfolgreich gesendet!"]);

} else {

echo json\_encode(["status" => "error", "message" => "Fehler beim Senden der E-Mail."]);

}

} else {

// Return error for non-POST requests

echo json\_encode(["status" => "error", "message" => "Ungültige Anfrage."]);

}

?>

1. send\_termin.php

<?php

// Error reporting for debugging

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

// Check if POST request is made

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get the JSON posted data

$data = json\_decode(file\_get\_contents('php://input'), true);

// Check if data is received

if ($data) {

// Sanitize and prepare the form data

$name = htmlspecialchars($data['name'], ENT\_QUOTES, 'UTF-8');

$email = htmlspecialchars($data['email'], ENT\_QUOTES, 'UTF-8');

// Format the date

$date = htmlspecialchars($data['date'], ENT\_QUOTES, 'UTF-8');

$date = date('d.m.Y', strtotime($date)); // Nur den Tag um 1 erhöhen

$time = htmlspecialchars($data['time'], ENT\_QUOTES, 'UTF-8');

$topic = htmlspecialchars($data['topic'], ENT\_QUOTES, 'UTF-8');

$message = nl2br(htmlspecialchars($data['message'], ENT\_QUOTES, 'UTF-8'));

// Prepare the email for you

$to = "medi.edi@gmx.de"; // Your IONOS email to receive the appointment messages

$email\_subject = "Terminvereinbarung YISU Travel";

$email\_body = "

<!DOCTYPE html>

<html lang='de'>

<head>

<meta charset='UTF-8'>

<meta name='viewport' content='width=device-width, initial-scale=1.0'>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 20px;

background-color: #f7f7f7;

}

.container {

max-width: 600px;

margin: auto;

background: #ffffff;

padding: 30px;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

h2 {

color: #333333;

margin-bottom: 20px;

font-size: 24px;

text-align: left;

}

.info {

font-size: 16px;

color: #333333;

margin-bottom: 10px;

}

.message {

font-size: 16px;

line-height: 1.5;

color: #333333;

margin-top: 15px;

background-color: #ffffff; /\* Hellerer Hintergrund für die Nachricht \*/

border-radius: 4px;

}

.footer {

margin-top: 20px;

font-size: 12px;

color: #777777;

border-top: 1px solid #e0e0e0;

padding-top: 10px;

text-align: center;

}

</style>

</head>

<body>

<div class='container'>

<h2>Neue Terminvereinbarung von www.yisu-travel.de</h2>

<div class='info'><strong>Name:</strong> $name</div>

<div class='info'><strong>E-Mail:</strong> $email</div>

<div class='info'><strong>Datum:</strong> $date</div>

<div class='info'><strong>Uhrzeit:</strong> $time</div>

<div class='info'><strong>Thema:</strong> $topic</div>

<div class='message'>

<strong>Nachricht:</strong><br>

$message

</div>

</div>

</body>

</html>

";

// Set the email headers for the main email

$headers = "From: info@yisu-travel.de\r\n";

$headers .= "Reply-To: $email\r\n";

$headers .= "Content-Type: text/html; charset=UTF-8\r\n";

// Send the email to you

if (mail($to, $email\_subject, $email\_body, $headers)) {

// Prepare confirmation email for the user

$user\_email\_subject = "Ihre Terminvereinbarung an YISU Travel";

$user\_email\_body = "

<!DOCTYPE html>

<html lang='de'>

<head>

<meta charset='UTF-8'>

<meta name='viewport' content='width=device-width, initial-scale=1.0'>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 20px;

background-color: #f7f7f7;

}

.container {

max-width: 600px;

margin: auto;

background: #ffffff;

padding: 30px;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

h2 {

color: #333333;

margin-bottom: 20px;

font-size: 24px;

text-align: left;

}

.info {

font-size: 16px;

color: #333333;

margin-bottom: 10px;

}

.message {

font-size: 16px;

line-height: 1.5;

color: #333333;

margin-top: 15px;

background-color: #ffffff; /\* Hellerer Hintergrund für die Nachricht \*/

border-radius: 4px;

}

.footer {

margin-top: 20px;

font-size: 12px;

color: #777777;

border-top: 1px solid #e0e0e0;

padding-top: 10px;

text-align: center;

}

</style>

</head>

<body>

<div class='container'>

<h2>Vielen Dank für Ihre Terminvereinbarung an YISU Travel!</h2>

<div class='info'><strong>Name:</strong> $name</div>

<div class='info'><strong>E-Mail:</strong> $email</div>

<div class='info'><strong>Datum:</strong> $date</div>

<div class='info'><strong>Uhrzeit:</strong> $time</div>

<div class='info'><strong>Thema:</strong> $topic</div>

<div class='message'>

<strong>Ihre Nachricht:</strong><br>

$message

</div>

<div class='footer'>

<p>Wir werden uns so schnell wie möglich bei Ihnen melden.</p>

</div>

</div>

</body>

</html>

";

// Set the email headers for the confirmation email

$user\_headers = "From: info@yisu-travel.de\r\n";

$user\_headers .= "Reply-To: info@yisu-travel.de\r\n";

$user\_headers .= "Content-Type: text/html; charset=UTF-8\r\n";

// Send the confirmation email to the user

mail($email, $user\_email\_subject, $user\_email\_body, $user\_headers);

echo json\_encode(["status" => "success", "message" => "E-Mail wurde erfolgreich gesendet!"]);

} else {

echo json\_encode(["status" => "error", "message" => "Fehler beim Senden der E-Mail."]);

}

} else {

echo json\_encode(["status" => "error", "message" => "Fehlende Daten."]);

}

} else {

// Return error for non-POST requests

echo json\_encode(["status" => "error", "message" => "Ungültige Anfrage."]);

}

?>

1. api.php

<?php

include '../../backend/config.php';

$link = new mysqli($host\_name, $user\_name, $password, $database);

header('Content-Type: application/json');

$method = $\_SERVER['REQUEST\_METHOD'];

$data = json\_decode(file\_get\_contents("php://input"), true);

switch ($method) {

case 'GET':

handleGetRequest($link);

break;

case 'POST':

handlePostRequest($data, $link);

break;

case 'PUT':

echo json\_encode(['message' => 'PUT-Anfrage empfangen', 'data' => $data]);

break;

case 'DELETE':

echo json\_encode(['message' => 'DELETE-Anfrage empfangen', 'data' => $data]);

break;

default:

echo json\_encode(['message' => 'Ungültige Anfrage-Methode']);

break;

}

function handleGetRequest($link) {

$date = $\_GET['date'] ?? date('Y-m-d'); // Default to today's date

// Force the date to UTC to avoid time zone issues

$stmt = $link->prepare("SELECT time FROM appointments WHERE DATE(date) = ?");

$stmt->bind\_param("s", $date);

$stmt->execute();

$result = $stmt->get\_result();

$appointments = [];

while ($row = $result->fetch\_assoc()) {

$appointments[] = $row['time']; // Retrieve only times

}

$stmt->close();

echo json\_encode([

'success' => true,

'message' => 'Appointments retrieved.',

'appointments' => $appointments

]);

}

function handlePostRequest($data, $link) {

$name = $data['name'] ?? '';

$email = $data['email'] ?? '';

$date = $data['date'] ?? '';

$time = $data['time'] ?? '';

if (empty($name) || empty($email) || empty($date) || empty($time)) {

echo json\_encode(['success' => false, 'message' => 'Alle Felder müssen ausgefüllt sein.']);

return;

}

// Berechnung der Zeitspanne

$datetime = new DateTime("$date $time");

$startTime = $datetime->modify('-29 minutes')->format('Y-m-d H:i:s');

$endTime = (clone $datetime)->modify('+29 minutes')->format('Y-m-d H:i:s');

// Überprüfen, ob ein Termin in den nächsten 30 Minuten existiert

$stmt = $link->prepare("SELECT COUNT(\*) FROM appointments WHERE date = ? AND (time BETWEEN ? AND ?)");

$stmt->bind\_param("sss", $date, $startTime, $endTime);

$stmt->execute();

$stmt->bind\_result($count);

$stmt->fetch();

$stmt->close();

if ($count > 0) {

handleExistingAppointments($date, $startTime, $endTime, $link);

} else {

insertNewAppointment($name, $email, $date, $time, $link);

}

}

function handleExistingAppointments($date, $startTime, $endTime, $link) {

$stmt = $link->prepare("SELECT date, time FROM appointments WHERE date = ? AND (time BETWEEN ? AND ?)");

$stmt->bind\_param("sss", $date, $startTime, $endTime);

$stmt->execute();

$result = $stmt->get\_result();

$appointments = [];

while ($row = $result->fetch\_assoc()) {

$appointments[] = [

'date' => $row['date'],

'time' => $row['time']

];

}

$stmt->close();

echo json\_encode([

'success' => false,

'message' => 'Ein Termin kann nicht gebucht werden, da innerhalb der nächsten 30 Minuten bereits ein Termin besteht.',

'appointments' => $appointments

]);

}

function insertNewAppointment($name, $email, $date, $time, $link) {

$stmt = $link->prepare("INSERT INTO appointments (name, email, date, time) VALUES (?, ?, ?, ?)");

$stmt->bind\_param("ssss", $name, $email, $date, $time);

if ($stmt->execute()) {

echo json\_encode(['success' => true, 'message' => 'Termin erfolgreich gespeichert.']);

} else {

echo json\_encode(['success' => false, 'message' => 'Fehler beim Speichern des Termins: ' . $stmt->error]);

}

$stmt->close();

}

$link->close();

?>

Dabei ist zu beachten, dass der „include '../../backend/config.php';“ Bereich die config files enthält zur Anmeldung (Bereitgestellt von IONOS), die sollten außerhalb der öffentlichen Domain liegen. (z.B. test/angular prod files, dann test/backend/config.php)

1. config.php

<?php

$host\_name = 'db5016367545.hosting-data.io';

$database = 'dbs13306900';

$user\_name = 'dbu1673899';

$password = 'Y1sU\_Tr4v3l.Datab4se\_87!\_.&';

?>

Allgemeine Informationen:

<https://mdbootstrap.com/>

**Favicons richtig anzeigen lassen:**

https://www.softicons.com/icon-tools/icon-converter

runterladen und für jedes Format im index.html verweisen:

<link rel="icon" type="image/png" sizes="16x16" href="public/favicon/favicon1 (2).png">

<link rel="icon" type="image/png" sizes="24x24" href="public/favicon/favicon1 (3).png">

<link rel="icon" type="image/png" sizes="32x32" href="public/favicon/favicon1 (5).png">

<link rel="icon" type="image/png" sizes="48x48" href="public/favicon/favicon1 (7).png">

<link rel="icon" type="image/png" sizes="128x128" href="public/favicon/favicon1 (9).png">

<link rel="icon" type="image/png" sizes="256x256" href="public/favicon/favicon1 (1).png">

**Wichtig: canonical definieren, um die wichtigste Seite anzugeben:**

<link rel="canonical" href="https://yisu-travel.de/startseite" />

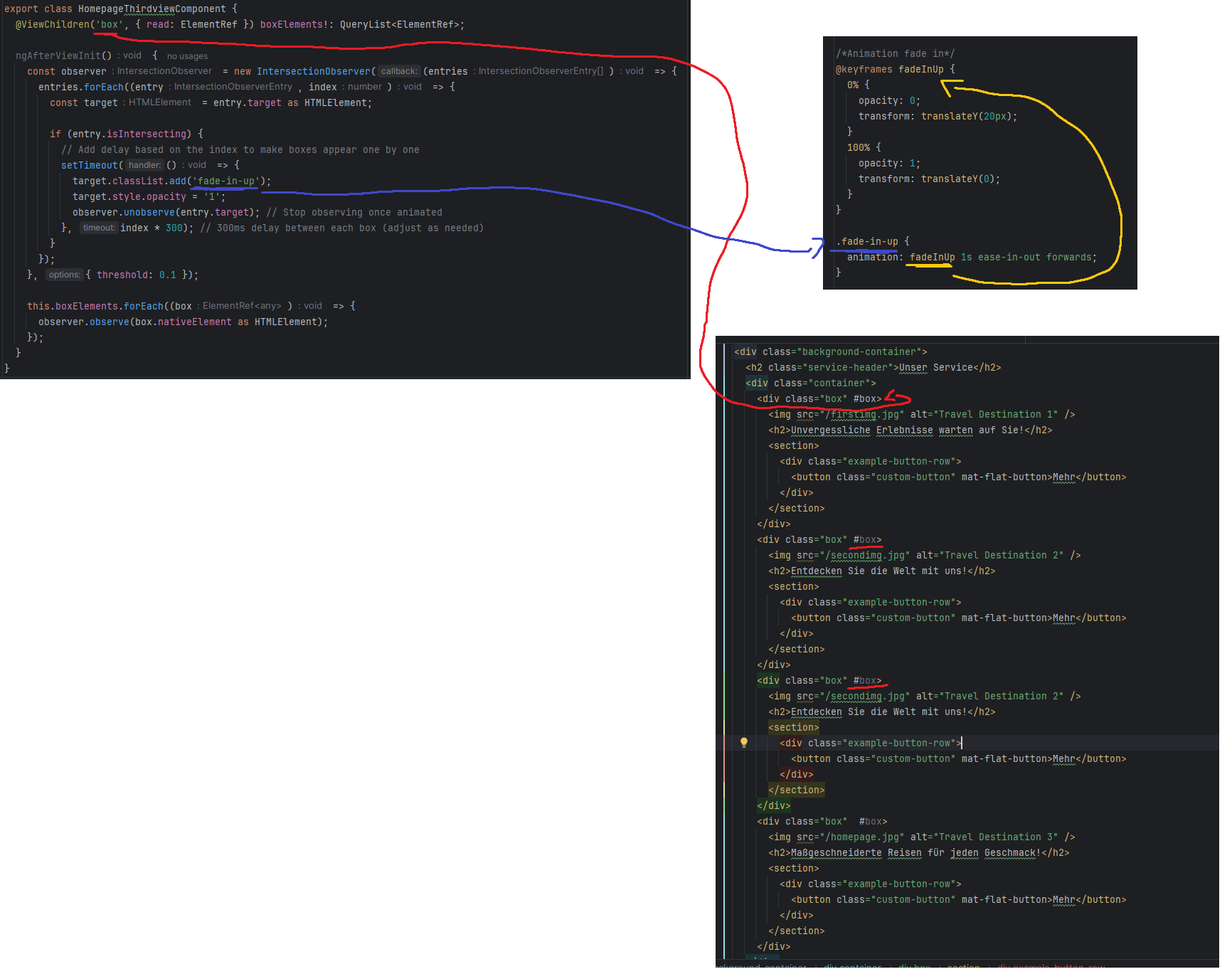
1. /backend/version.php

<?php

header('Content-Type: application/json');

echo json\_encode(['version' => '1.0.0']);

**Richtig animieren mit Angular:**

****

**Cache Problem lösen**

Die Datei ngsw-config.json bearbeiten:

Das hinzufügen:  
"dataGroups": [  
 {  
 "name": "cache",  
 "urls": ["/backend/version"],  
 "cacheConfig": {  
 "strategy": "freshness",  
 "maxAge": "0s",  
 "timeout": "0s",  
 "maxSize": 0  
 }  
 }  
]

Ganzer Code:  
{  
 "$schema": "./node\_modules/@angular/service-worker/config/schema.json",  
 "index": "/index.html",  
 "assetGroups": [  
 {  
 "name": "app",  
 "installMode": "prefetch",  
 "resources": {  
 "files": [  
 "/favicon.ico",  
 "/index.html",  
 "/manifest.webmanifest",  
 "/\*.css",  
 "/\*.js"  
 ]  
 }  
 },  
 {  
 "name": "assets",  
 "installMode": "lazy",  
 "updateMode": "prefetch",  
 "resources": {  
 "files": [  
 "/\*\*/\*.(svg|cur|jpg|jpeg|png|apng|webp|avif|gif|otf|ttf|woff|woff2)"  
 ]  
 }  
 }  
 ],  
 "dataGroups": [  
 {  
 "name": "cache",  
 "urls": ["/backend/version"],  
 "cacheConfig": {  
 "strategy": "freshness",  
 "maxAge": "0s",  
 "timeout": "0s",  
 "maxSize": 0  
 }  
 }  
 ]  
}

**Danach Service erstellen um von dort manuell die Versionsnummer jedes Mal zu ändern, nachdem man einen Deploy gemacht hat:**

export class VersionControlService {  
 private localVersion = '1.0.0'; // Example local version, adjust as necessary  
  
 constructor(private http: HttpClient) {}  
  
 checkVersion(): Observable<{ version: string }> {  
 return this.http.get<{ version: string }>('/backend/version.php');  
 }  
  
 isVersionMismatch(remoteVersion: string): boolean {  
 return remoteVersion !== this.localVersion;  
 }  
}

**Anschließend im App Component ts aufrufen:**

ngOnInit() {

this.checkVersion();  
 // Register the service worker  
 if ('serviceWorker' in *navigator*) {  
 *navigator*.serviceWorker.register('ngsw-worker.js').then(registration => {  
 *console*.log('Service Worker registered with scope:', registration.scope);  
 }).catch(error => {  
 *console*.error('Service Worker registration failed:', error);  
 });  
 }  
}

checkVersion() {  
 this.versionService.checkVersion().subscribe(  
 (response) => {  
 if (this.versionService.isVersionMismatch(response.version)) {  
 *console*.log('Version mismatch detected. Refreshing page...');  
 this.updateServiceWorker();  
 }  
 },  
 (error) => {  
 *console*.error('Failed to fetch version:', error);  
 }  
 );  
}  
  
updateServiceWorker() {  
 if ('serviceWorker' in *navigator*) {  
 *navigator*.serviceWorker.getRegistrations().then(registrations => {  
 for (let registration of registrations) {  
 registration.unregister().then(() => {  
 *console*.log('Service Worker unregistered. Reloading page...');  
 *location*.reload(); // Reload the page after unregistering the service worker  
 });  
 }  
 }).catch(error => {  
 *console*.error('Failed to unregister service workers:', error);  
 });  
 }  
}

Project im Netzwerk erreichbar machen:  
ng serve --host 0.0.0.0