

D.B.A.F. Dokumentation

Deutsche Bahn
Anwendungs-Frontend

Planung & Dokumentation
Erik Ament

Planung, Entwicklung & Dokumentation
Oliver Hollmann

Planung & Entwicklung
Christopher Falk

Inhaltsverzeichnis

| | |
|--|----|
| 1. Die Idee | 3 |
| 2. Verwendete Programme und Hilfsmittel..... | 3 |
| 3. Installationshinweise | 3 |
| 4. Funktionalitäten | 4 |
| a. Stationsdaten | 4 |
| b. Fahrplanauskunft | 4 |
| c. Routenplaner..... | 5 |
| d. Karte | 5 |
| 5. Funktionalitäten für Fortgeschrittene | 6 |
| a. DevExtreme-Komponenten..... | 6 |
| b. Selbstdefinierte Funktionalitäten..... | 7 |
| 6. Erweiterungsmöglichkeiten | 8 |
| 7. Arbeitsnachweis..... | 8 |
| a. Alle Arbeitszeiten nach Datum..... | 8 |
| b. Arbeitsverteilung nach Person | 12 |
| 8. Quellenverzeichnis..... | 12 |

1. Die Idee

Die Idee unseres Projektes ist es, Daten von der Deutschen Bahn API einfach und übersichtlich darzustellen, sodass sich der Nutzer einfach durch die Datenmenge navigieren kann, die durch Aufrufe der API zurückkommen, um alle wissenswerten Informationen zu erhalten.

Für diese Aufgabe haben wir uns folgende Schwerpunkte für die verschiedenen Sichtungsmöglichkeiten festgelegt:

- Anzeige der API-Daten aller Stationen deutschlandweit
- Fahrplanauskunft für alle Abfahrten von und Ankünfte an einem Bahnhof an einem vom Nutzer festgelegten Datum
- Routenplanung zur Darstellung verschiedener Daten für eine Route von Startbahnhof A zu Endbahnhof B
- eine Karte für alle Bahnhöfe in Deutschland, welche nach Bundesland gefiltert werden können

2. Verwendete Programme und Hilfsmittel

Zur Entwicklung des Source-Codes verwendeten wir Visual Studio Code (<https://code.visualstudio.com/>) und für die Ausführung des Quellcodes nutzten wir XAMPP (<https://www.apachefriends.org/>). Unsere Datenquelle ist eine API (<https://v6.db.transport.rest/>), welche intern auf die Deutsche Bahn API zugreift, wofür aber kein Zugangs-Schlüssel benötigt wird, um die jeweiligen Daten zu Bahnhöfen und Fahrplänen zu erhalten.

Zusätzlich zu den eben genannten Programmen und Hilfsmitteln bedienten wir uns der DevExtreme-Bibliothek (JavaScript-Komponenten) von DevExpress (<https://js.devexpress.com/>) zur einfachen Erstellung und Gestaltung der Benutzeroberfläche und zur übersichtlichen Anzeige aller Daten für den Nutzer.

3. Installationshinweise

Alle Quelldateien sowie diese Dokumentation befinden sich auf GitHub unter <https://github.com/chfalkk/DBAF>.

Zur Ausführung der Dateien im Ordner "src" kann entweder XAMPP genutzt oder das Projekt direkt per Ausführung der index.html gestartet werden, da das gesamte Projekt über JavaScript gesteuert wird, wofür kein Zusatzprogramm benötigt wird, außer der Browser, sofern er JavaScript unterstützt.

Hinweis: Für die Ausführung muss JavaScript im Browser aktiviert sein. Auf der Seite <https://support.google.com/adsense/answer/12654?hl=de> wird am Beispiel von Chrome beschrieben, wie JavaScript zu aktivieren ist, wenn es bereits deaktiviert wurde.

4. Funktionalitäten

a. Stationsdaten

D.B.A.F. Stations-Daten Fahrplanauskunft Routenplaner Karte Über uns

Stationsdaten

In dieser Tabelle werden alle in Deutschland verfügbaren Stationen mit verschiedenen Daten angezeigt. Dieser gesamte Datenbestand kann nach Daten durchsucht, gefiltert und gruppiert werden.

| Stationen | Stadt | Postleitzahl | Strasse | Alle einblenden | Gruppieren |
|--|------------------|--------------|-----------------------------|-------------------|------------|
| Q | Q | Q | Q | Stufenlos eingang | |
| Bundesland: Baden-Württemberg (Anzahl: 899) (Klicken auf der nächsten Seite) | | | | | |
| Adler Hbf | Adler | 72620 | Am Bahnhof 1 | ✓ | |
| Achern | Achern | 77811 | Friedrichstr. 40 | ✓ | |
| Achheimen Nord | Ottensheim | 74706 | Endenburger Str. 48 | ✓ | |
| Achheimen Ost | Achheimen | 74740 | Am Bahnhof 2 | ✓ | |
| Ahlbeck | Übersum | 74182 | Johannestraße 18 | ✓ | |
| Ahlbeckhausen | Ahlbeckhausen | 74814 | Bahnhofstraße 40 | ✓ | |
| Ahlbeck | Sülzheim | 74619 | Verderstraße 2 | ✓ | |
| Ahlbeck | Ahlbeck | 86337 | Poststr. 8 | ✓ | |
| Ahlbeck | Ahlbeck | 99771 | Bahnhofstr. 1 | ✓ | |
| Ahlbeck-Übersum | Ahlbeck | 72620 | Am Bahnhof 1 | ✓ | |
| Ahlbeck-Übersum | Ahlbeck | 72620 | Poststr. 27 | ✓ | |
| Ahlbeck-Lauten Ost | Ahlbeck | 72619 | Stadtweg 1 | ✓ | |
| Ahlbeck-Lauten West | Ahlbeck | 72619 | Von Vöhringstr. Straße 17/1 | ✓ | |
| Ahlbeck | Ahlbeck | 78534 | Bahnhofstr. 3 | ✓ | |
| Ahlbeck | Ahlbeck | 78534 | Karlshafen Str. 14 | ✓ | |
| Ahlbeck | Ahlbeck | 86644 | Bahnhofstr. 11 | ✓ | |
| Ahlbeck | Ahlbeck | 72112 | Bahnhofstr. 10 | ✓ | |
| Ahlbeck | Ahlbeck | 72176 | Bahnhofstrasse 1 | ✓ | |
| Ahlbeck-Ost | Feldberg-Ahlbeck | 79968 | Feldbergstr. 18 | ✓ | |

Anzahl: 899

1 2 3 4 5 ... 189 >

Die Stationsdaten aller Bahnhöfe in Deutschland werden in einer Tabelle angezeigt und können nach Bezeichnung, Stadt, Postleitzahl, Straße und der Verfügbarkeit eines stufenlosen Eingangs gefiltert und durchsucht werden. Standardmäßig werden alle Daten nach Bundesland gruppiert, wobei die Gruppierung vom Nutzer jederzeit angepasst werden kann.

b. Fahrplanauskunft

D.B.A.F. Stations-Daten Fahrplanauskunft Routenplaner Karte Über uns

Fahrplanauskunft

Mit der Eingabe eines Railcodes und dem Datum können alle Abfahrten und Ankünfte mit allen Daten ab diesem Bahnhof angezeigt werden.

Fahrplan-Suche

Stationen: Magdeburg Hbf

Datum: 01.04.2023

Fahrplan laden

Übersicht

Für diese Station ist ein stufenloser Eingang vorhanden.

Abfahrten ab Magdeburg Hbf

| Gestapelte Stationen (Prognose) | Genauer Stationen | Entfernung | Plattform |
|---------------------------------|-------------------|-------------------|-----------|
| Q | Q | Q | Q |
| 30.04.2023, 09:42 | 30.04.2023, 09:42 | Frankfurt/Oder | unbekannt |
| 30.04.2023, 09:51 | 30.04.2023, 09:51 | Aachen | 5 A-C |
| 30.04.2023, 09:53 | 30.04.2023, 09:53 | Köln | 4 D-G |
| 30.04.2023, 09:56 | 30.04.2023, 09:56 | Leipzig/Halle | 2 D-G |
| 30.04.2023, 09:58 | 30.04.2023, 09:58 | Köln Hbf | 8 |
| 30.04.2023, 10:04 | 30.04.2023, 10:04 | Berlin Ostbahnhof | 7 |
| 30.04.2023, 10:14 | 30.04.2023, 10:14 | Leipzig/Halle | 3 A-C |
| 30.04.2023, 10:19 | 30.04.2023, 10:19 | Frankfurt/Oder | unbekannt |
| 30.04.2023, 10:21 | 30.04.2023, 10:21 | Frankfurt/Oder | 6 |
| 30.04.2023, 10:27 | 30.04.2023, 10:27 | Leipzig/Halle | 7 |
| 30.04.2023, 10:31 | 30.04.2023, 10:31 | Braunschweig Hbf | 6 |

Anzahl: 243

1 2 3 4 5 ... 189 >

Ankünfte in Magdeburg Hbf

Bei der Fahrplanauskunft können Station und Datum angegeben werden, wobei Abfahrten und Ankünfte an dem eingegebenen Datum angezeigt werden. Zusätzlich wird unter „Übersicht“ die Verfügbarkeit eines stufenlosen Eingangs angezeigt. Der Nutzer kann hier nach der geplanten Ankunfts-/Abfahrtszeit (Prognose), genauer Ankunfts-/Abfahrtszeit, Herkunft/Richtung und Plattform filtern, suchen und sortieren.

c. Routenplaner

Bei der Routenplanung kann der Nutzer Start- und Zielbahnhof mit einem Startzeitpunkt angeben. Die Berechnung der Reise und Zwischenstopps werden durch die API übernommen. Alle Daten, welche durch die API zurückgegeben werden, werden in einer Übersicht und Tabelle angezeigt.

In der Übersicht werden alle Gesamtergebnisse angezeigt, wie die gesamte Fahrtzeit, Wartezeit, Reisedauer sowie die Anzahl der Umstiege und der voraussichtlichen Kosten (sofern ermittelbar).

In der Tabelle werden alle Daten zu jeder einzelnen Fahrt der Reise angezeigt, wie der Linien-Name, die Start- und Endstation, die Start- und Endzeit jeder Fahrt, die Fahrtzeit zwischen zwei Stationen und die Wartezeit zur nächsten Abfahrt.

Es sollte beachtet werden, dass für Richtigkeit der Daten nicht garantiert werden kann, da sich Fahrpläne ändern, verzögern oder gestört werden können.

d. Karte

In der Karte werden alle Stationen deutschlandweit angezeigt. Die Filterung nach Bundesland ist möglich, wenn im Dropdown über der Karte der entsprechende Wert angegeben wird.

Mit einem Klick auf einen Stations-Marker in der Karte werden einige Bahnhofsdaten zu dieser Station

angezeigt und wenn nach Auswahl eines Markers ein Klick auf "Abfahrten und Ankünfte zu diesem Bahnhof anzeigen" ausgeführt wird, so wird der Nutzer zur Fahrplanauskunft weitergeleitet. Dort wird das Formular automatisch mit der zuletzt auf der Karte ausgewählten Station ausgefüllt.

5. Funktionalitäten für Fortgeschrittene

a. DevExtreme-Komponenten

Zur einfachen Erstellung von übersichtlichen Oberflächen haben wir die von DevExpress erstellten DevExtreme JavaScript-Komponenten genutzt. In der DevExtreme-Library werden über 50 Komponenten mitgeliefert, inklusive Icons, Skins bzw. Designs (dunkler Stil, heller Stil, Farben für die Seitendarstellung), Spracheinstellungen und weiteren nützlichen Funktionen.

```

211 // Tabelle der Fahrpläne
212 let eGridJourneys = $('#journeyTable').dxDataGrid({
213     dataSource: eJourneys,
214     keyExpr: 'ID',
215     allowColumnResizing: true,
216     allowColumnReordering: true,
217     filterRow: { visible: true },
218     searchPanel: { visible: true },
219     loadPanel: { enabled: true },
220
221     // Unterteilung der Tabelle in Seiten
222     pager: {
223         showPageSizeSelector: true,
224         allowedPageSizes: [5, 10, 20, 50],
225         showNavigationButtons: true
226     },
227
228     toolbar: {
229         // Such-Option anzeigen
230         items: [
231             'searchPanel'
232         ]
233     },
234
235     // Spalten der Tabelle
236     columns: [
237         {
238             dataField: 'ID',
239             visible: false,
240             sortOrder: 'asc' // Sortiert nach ID, obwohl diese Spalte nicht angezeigt wird
241         },
242         {
243             dataField: 'line',
244             caption: 'Linie'
245         },
246         {
247             dataField: 'startStation',
248             caption: 'Abfahrts-Station'
249         },
250         {
251             dataField: 'endStation',
252             caption: 'Ankunfts-Station'
253         },
254         {
255             dataField: 'timeStart',
256             caption: 'Startzeit',
257             dataType: 'datetime'
258         },
259         {
260             dataField: 'timeEnd',
261             caption: 'Endzeit',
262             dataType: 'datetime'
263         },
264         {
265             dataField: 'timeWaiting',
266             caption: 'Wartezeit vom vorherigen Stop'
267         },
268         {
269             dataField: 'timeTrip',
270             caption: 'Voraussichtliche Trip-Zeit'
271         }
272     ],
273
274     // Zusammenfassung
275     summary: {
276         // Anzeige der Trip-Details
277         totalItems: [
278             {
279                 column: 'timeWaiting',
280                 summaryType: 'count'
281             }
282         ]
283     }
284 });

```

Beispiel: Source-Code der Fahrplan-Tabelle im Routenplaner (als Komponente „cxDataGrid“)

Alle Komponenten werden über sogenannte Properties (= Eigenschaften) eingestellt und die Daten werden in einer Variable gespeichert (hier: eGridJourneys), wodurch einzelne Einstellungen während der Laufzeit geändert oder gelesen werden können (z.B. Eingabewerte in Formularen oder Änderung/Aktualisierung einer Datenquelle für eine Tabelle).

Die wichtigsten Einstellungsmöglichkeiten bei einer DevExtreme-Tabelle sind:

- dataSource (Datenquelle)
- keyExpr (Primärschlüssel der Datenquelle – hier: ID)
- columns (Spalten):
 - dataField (Name des zugehörigen Datenfeldes in der Datenquelle)
 - visible (Sichtbarkeit)
 - caption (Anzeigenname der Spalte in der Tabelle)
 - sortOrder (Sortierung)
- und viele weitere

Eine Übersicht zu allen Komponenten von DevExtreme befindet sich unter <https://js.devexpress.com/Overview/Widgets/>. Alle Dateien des Libraries befinden sich bei der Installation unseres Projektes im lokalen Ordner "dx_22.2.4"

b. Selbstdefinierte Funktionalitäten

Alle selbstdefinierten JavaScript-Funktionalitäten befinden sich im Ordner „js“.

In der folgenden Tabelle werden die Aufgaben- bzw. Funktionsbereiche jeder selbsterstellten JavaScript-Datei aufgelistet:

| Dateiname | Beschreibung |
|-------------------|--|
| dbaf.js | Diese Datei beinhaltet alle Konstanten und Einstellungen, welche für jede Seite nötig sind. |
| request.js | Diese Datei beinhaltet alle Funktionalitäten, um Anfragen an die API auszuführen. |
| toolbar.js | Diese Datei beinhaltet die Einstellungen für die Navigationsleiste auf jeder Seite. |
| ui.js | Diese Datei beinhaltet die nötigen Funktionen, um Komponenten auf der Oberfläche mit speziellen Anzeige-Einstellungen zu versehen. |

Nachdem die Funktion jeder Datei geklärt wurde, werden nun einzelne Funktionen bzw. Klassen jeder Datei in der folgenden Tabelle aufgelistet:

| Dateiname | Funktion | Beschreibung |
|-------------------|---|--|
| request.js | Klasse DBAF_Request mit Konstruktor constructor(RequestParam, Async, OnSuccess) | Erstellung einer Anfrage an die API mit Erstellung eines DBAF_Request-Objektes. RequestParam – Festlegung, welche Daten abgeholt werden sollen Async – Synchrone oder asynchrone Ausführung der Anfrage OnSuccess – Übergabe der Anfrageergebnisse, falls die Anfrage erfolgreich war |
| | IsAPIAvailable() | Gibt zurück, ob die API erreichbar ist. |
| | GetAllStations(FederalState, OnSuccess = null) | Alle Stationen, gefiltert nach Bundesland, zurückgeben. FederalState – Bundesland nach dem gefiltert werden soll („Alle“ für ganz Deutschland) OnSuccess – Rückgabe des Ergebnisses |
| | GetAllDepartures(StationID, TargetDate, OnSuccess = null) | Alle Abfahrten von einer Station an einem festgelegten Datum abholen. StationID – ID der Zielstation TargetDate – gewünschtes Datum OnSuccess – Rückgabe des Ergebnisses |
| | GetAllArrivals(StationID, TargetDate, OnSuccess = null) | Alle Ankünfte an einer Station an einem festgelegten Datum abholen. StationID – ID der Zielstation TargetDate – gewünschtes Datum OnSuccess – Rückgabe des Ergebnisses |

| | | |
|-------|---|---|
| | ConvertMinutesToStr(Minutes, ShowZero) | Minuten zu einem String „x Stunde(n), y Minute(n)“ konvertieren. Minutes – Anzahl der Minuten ShowZero – wenn als True festgelegt: 0 Stunde(n), 0 Minute(n) wird zurückgegeben, falls Minutes = 0; wenn als False festgelegt: keine Rückgabe, wenn Minutes = 0 |
| | GetAllJourneys(StartStationID, EndStationID, StartDateTime, OnSuccess = null) | Fahrplan von einer Startstation zu einer Endstation ab einer gewünschten Startzeit abholen. StartStationID – ID der Start-Station EndStationID – ID der End-Station StartDateTime – Zeit, ab wann die nächste Reise stattfinden soll OnSuccess – Rückgabe des Ergebnisses |
| ui.js | StretchToClient(ElementID) | Ein HTML-Element auf die maximal verfügbare Höhe strecken. ElementID – ID des Elementes als String |
| | StickToClient(ElementID) | Ein HTML-Element immer auf der maximal verfügbaren Höhe halten (auch wenn sich die Größe des Fensters ändert). ElementID – ID des Elementes als String |

6. Erweiterungsmöglichkeiten

Eine Erweiterungsmöglichkeit für einen fortgeschrittenen Nutzer wäre, dass die Fahrstühle für jedes Gleis in der Tabelle der Fahrplanauskunft in einer Extra-Spalte (z.B. „Fahrstuhl zum Gleis vorhanden“) ausgegeben werden, anstelle des stufenlosen Eingangs einer Station. Hierfür müsste eine Abfrage an die FaSta (Station Facility Status) der Deutschen Bahn API ausgeführt werden.

Eine weitere Möglichkeit, unser Projekt zu erweitern, wäre die Einstellung eines Filters der bevorzugten Transportmittel beim Routenplaner (nach Bus, Bahn, S-Bahn, U-Bahn, Regionalbahn und/oder ICE). Zurzeit wird die nächstmögliche Reiseroute angegeben, ohne die Transportmittel zu beachten.

7. Arbeitsnachweis

a. Alle Arbeitszeiten nach Datum

Diese Tabelle listet auf, wann welches Gruppenmitglied an welcher Aufgabe für wie lange gearbeitet hat:

| Aufgabe | Mitarbeiter | Beschreibung | Zeit | Datum |
|---|-----------------------|---|------|------------|
| Planung des Grundaufbaus des Projekts, Einrichtung GitHub-Projekt | Falk, Ament, Hollmann | Issues erstellt (zu erledigende Aufgaben) | 1.5h | 17.01.2023 |
| Erstellung der | Hollmann | Klasse DBAPI_Base, DBAPI_Fahrplan | 4h | 17.01.2023 |

| | | | | |
|---|-----------------------|---|------|------------|
| Grundfunktionen für Fahrplan- und Station-Data-API | | und DBAPI_StaDa erstellt mit allen Funktionen | | |
| Weitere Planung und Fortsetzung der Grundfunktionen | Falk, Ament, Hollmann | Font eingestellt; begonnen Icons einzufügen; Verfügbarkeit der API kann abgefragt werden; Auskommentierung der Funktionen der API | 1.5h | 18.01.2023 |
| Grundfunktionen fortsetzen | Falk | Header erstellt, API-Status wird angezeigt, Bootstrap, jQuery und popper.js importiert | 4h | 27.01.2023 |
| Grundfunktionen fortsetzen, Optimierungen, kleinere Überarbeitungen | Hollmann | Lesbarkeit verbessert, Auskommentierungen, Grundstruktur für Fahrplanauskunft und „Über uns“ eingefügt, Festinstallation der Dateien von jQuery, Bootstrap und popper.js; Klasse MainHandler für alle API-Funktionen erstellt | 2.5h | 03.02.2023 |
| Eingabe für Fahrplanauskunft erstellt | Falk | Responsive Maske zur Eingabe der Fahrplansuchkriterien begonnen | 4h | 06.02.2023 |
| Anfänge Suchlogik, Vorbereitung Übergabefelder | Falk | Request-Klassen erstellt, PanelBuilder erstellt | 2h | 07.02.2023 |
| Responsives UI, JavaScript Form modifizieren | Falk | Eingabemaske für die Fahrplanauskunft fortgesetzt, Vorbereitung Fahrplananzeige | 3.5h | 15.02.2023 |
| Design der Index-Seite begonnen | Hollmann | Kacheln und Grunddesign auf der Index-Seite erstellt (erste Idee) Fehlerbehebung in der Fahrplanauskunft, kleinere Codeanpassungen | 1.5h | 17.02.2023 |
| Komponenten-Suche | Ament, Hollmann | Suche nach Komponenten zur Verbesserung und Vereinfachung der Oberfläche | 1h | 28.02.2023 |
| Projektplanung | Falk, Ament, Hollmann | Weitere Planung des Projektes (Umbau von PHP-Funktionen auf JavaScript), Nutzung von DevExpress-Komponenten („DevExtreme“) geplant | 1.5h | 06.03.2023 |
| Implementierung der DevExpress-Komponenten, Stationsübersicht erstellt | Hollmann | Download und Implementierung der DevExpress JavaScript-Komponenten, Erstellung einer Tabelle mit allen Stationen (Auslesen, Suchen, Filtern), Einbinden einer neuen API (V5-DB-REST-API) | 3h | 06.03.2023 |

| | | | | |
|---|-----------------------|--|------|------------|
| Weiterer Umbau des Projektes | Falk, Ament, Hollmann | Kartenübersicht begonnen (Download einer Shape-Datei von Deutschland), Umstellung der Komponenten in der Fahrplanübersicht begonnen | 1.5h | 08.03.2023 |
| Formular für Fahrplan-Abfragen fortgesetzt | Hollmann | Füllen des Dropdowns bei der Bahnhofsauswahl für die Fahrplanauskunft; Datumsauswahl angepasst | 2h | 09.03.2023 |
| Formular für Fahrplan-Abfragen fortgesetzt | Hollmann | Funktion zur Abfrage der Ankunftsdaten an einem Bahnhof erstellt; „Paging“ beim Dropdown hinzugefügt (Daten werden seitenweise nacheinander geladen) | 1h | 05.04.2023 |
| Formular für Fahrplan-Abfragen fortgesetzt, Navigationsleiste erstellt | Hollmann | Tabellen in der Stationsdaten-Abfrage erstellt; Funktionen zur Abfrage aller Ankunfts- und Abfahrtszeiten für einen Bahnhof an einem festgelegten Datum erstellt; Navigationsleiste für alle Seiten erstellt; Fehlerbehebungen bei den Stationsdaten | 3h | 06.04.2023 |
| Routenplaner begonnen | Hollmann | Formular mit Auswahl von zwei Bahnhöfen erstellt, Funktion erstellt, um eine Route zwischen zwei Bahnhöfen mit allen Zwischenstopps von der API zu laden, Routen-Ergebnisse werden in einer Tabelle dargestellt (Linien-Name, Abfahrt- und Endstation und Trip-Zeit), kleinere Anzeigeverbesserungen bei der Navigationsleiste | 2.5h | 11.04.2023 |
| Routenplaner fertiggestellt | Hollmann | Übersicht einer Fahrt wird angezeigt (gesamte Trip-Dauer, gesamte Wartezeit zwischen Stationen, Anzahl an Umstiegen/Laufwegen und die Fahrtkosten), Anzeige der Wartezeiten zwischen Stationen und Start- und Endzeit in der Tabelle, kleinere Anpassungen der Texte bei der Routenplanung | 1.5h | 13.04.2023 |
| Kartenübersicht fortgesetzt, Begrenzung der Anfragen-Anzahl an die API | Hollmann | Anzahl der Anfragen an die API auf 100/Minute in der Query-Funktion begrenzt (Vorgabe der API), Kartenübersicht fortgesetzt: Anzeige der Weltkarte (Fokus auf | 1h | 14.04.2023 |

| | | | | |
|--|----------|--|------|------------|
| | | Deutschland) über Bing mit Hilfe der dxMap-Komponente von DevExtreme | | |
| Kartenübersicht fortgesetzt | Hollmann | Alle Bahnhöfe werden auf der Karte angezeigt und können nach Bundesland gefiltert werden, Tooltips werden für Bahnhöfe angezeigt, kleinere Fehlerbehebungen | 1.5h | 18.04.2023 |
| Kartenübersicht fertiggestellt, vollständige Auskommentierung begonnen, Issues aktualisiert | Hollmann | In den Tooltips der Bahnhöfe auf der Karte kann zu der Fahrplanauskunft für den ausgewählten Bahnhof gesprungen werden, Optimierung der Streckung von Komponenten auf den maximal verfügbaren Platz, Beschreibungen zu allen Seiten hinzugefügt, „Über uns“-Seite ausgefüllt, Kommentierungen verständlicher formuliert und Kommentierungen für einige Funktionen hinzugefügt, Issues aktualisiert, globale Umstellung der API von v5.db.transport.rest auf v6.db.transport.rest | 1.5h | 19.04.2023 |
| Issues bearbeitet (<noscript>-Tags, Cleanup, index.html vorbereitet) | Hollmann | <noscript>-Tags zu allen Seiten hinzugefügt & getestet, Auskommentierungen / Refactoring / Code-Optimierungen, Flackern bei Seitenwechsel behoben, index.html und erste Version der Service-Übersicht erstellt, Anzeige für API-Status auf der Startseite hinzugefügt (Fehlermeldung, wenn API nicht erreichbar) & getestet | 2h | 20.04.2023 |
| Styling der index.html | Falk | Einbindung von „Font Awesome“ in die index.html, Styling der Service-Übersicht, Icons auf der index.html eingefügt | 1h | 21.04.2023 |
| Vorbereitung der Dokumentation, Absprachen zum Projekt | Hollmann | Dokumentations-Vorlage erstellt, Absprachen zur Dokumentation und noch offenen Issues, Issues aktualisiert, index.html: Service-Übersicht fertiggestellt, Aufräumen der Daten auf GitHub, Aktualisierung der README.md-Datei | 1.5h | 23.04.2023 |
| Bugfixes, Anzeige „stufenloser | Hollmann | „Stufenloser Eingang“-Eigenschaft wird nun in den Stations-Daten und in | 1h | 25.04.2023 |

| | | | | |
|----------------------|----------|--|------|------------|
| Eingang“ | | der Übersicht für die Fahrplanauskunft angezeigt, Laden der Stationen verbessert, Anzeige der Ergebnis-Übersichten beim Routenplaner verbessert, sonstige kleine Code/GUI-Verbesserungen | | |
| Tests | Falk | Tests mit allen Modulen | 0.5h | 25.04.2023 |
| Tests | Ament | Tests mit allen Modulen | 0.5h | 27.04.2023 |
| Dokumentation | Ament | Vorbereitung der Dokumentation | 2.5h | 28.04.2023 |
| Tests | Hollmann | Tests mit allen Modulen, Behebung eines Anzeigefehlers in der Fahrplanauskunft und im Routenplaner | 0.5h | 30.04.2023 |
| Dokumentation | Hollmann | Dokumentation fertiggestellt | 2h | 30.04.2023 |

b. Arbeitsverteilung nach Person

In dieser Tabelle werden alle Arbeitszeiten zusammengerechnet nach Person dargestellt:

| Mitarbeiter | Aufgabenbereiche | Arbeitszeit insgesamt |
|--------------------|-------------------------------------|------------------------------|
| Ament | Planung, Dokumentation | 10 Stunden |
| Falk | Planung, Entwicklung | 21 Stunden |
| Hollmann | Planung, Entwicklung, Dokumentation | 39 Stunden |

8. Quellenverzeichnis

| Bezeichnung | URL | Aufrufdatum |
|---|---|--------------------|
| DevExtreme-Komponenten | https://js.devexpress.com/ | 28.02.2023 |
| V6-REST-API zur Abfrage aller Daten | https://v6.db.transport.rest/ | 06.03.2023 |
| Fontawesome (für die Icons auf index.html) | https://fontawesome.com/start | 22.04.2023 |