

# Interne-Berichte

Entwicklerdokumentationen zu den internen Berichten

- [Telefon-Auswertung Entwicklerdoku](#)
- [Datenfluss in internen Berichten](#)

# Telefon-Auswertung

## Entwicklerdoku

Um Daten aus der alten Telefonanlage und der neuen Placetel API auswerten, wurde ein Frontend und ein Backend entwickelt, welche dies ermöglichen.

## Backend

Das Backend besteht aus einer Express.js Rest API und einer Postgresql Datenbank.

## Installation

Um das Backend zu installieren, muss zunächst das Git-Repository geklont werden:

```
git clone http://gitlab.opendynamic.de/intern/telephone-statistic-backend.git
```

Daraufhin können alle benötigten Abhängigkeiten, mit folgendem Befehl installiert werden:

```
yarn
```

Im nächsten Schritt müssen die benötigten Umgebungsvariablen initialisiert werden. Um dies so einfach wie möglich zu machen, gibt es die Datei `.env_example`. Der Inhalt dieser Datei kann kopiert werden und in eine neue `.env` geschrieben werden.

Diese sollte dann wie folgt aussehen:

```
DATABASE_URL=postgresql://USER:PASSWORD@HOST:PORT/DBNAME?schema=public  
PLACETEL_TOKEN=
```

Das Backend greift auf eine Datenbank zu. Dafür muss in der `.env` Datei eine `DATABASE_URL` eingetragen werden. Hierfür müssen folgende Beispiel-Werte durch echte Werte ersetzt werden:

- USER → der Datenbankbenutzer
- PASSWORD → das Passwort des Datenbank-Benutzers

- HOST → die Adresse der Datenbank zum Beispiel <http://localhost>
- PORT → der Port unter welchem die Datenbank erreichbar ist, in den meisten Fällen ist dies 5432
- DBNAME → der Name der Datenbank

Daraufhin muss noch ein gültiger API-Token für Placetel eingetragen werden.

Wurden all diese Werte eingetragen, kann Prisma (das verwendete ORM) verwendet werden, um die benötigten Tabellen in die Datenbank zu schreiben. Dafür kann folgender Befehl ausgeführt werden:

```
yarn migrate:dev
```

Wenn in die Datenbank die Daten aus der alten Telefonanlage eingespielt werden sollen, kann folgender Befehl ausgeführt werden:

```
yarn import-old-data
```

Nun kann das Backend mit folgendem Befehl gestartet werden:

```
yarn start
```

## Projekt-Aufbau

Gestartet wird das Projekt über den Aufruf der `index.js` Datei. Dieses startet unter anderem vor allem den Router. Der Code zu diesem befindet sich unter `src/router.js`. Dieser behandelt alle eingehenden Anfragen und leitet Anfragen an den jeweiligen Controller weiter.

Die Controller laden angeforderte Daten, entweder aus der Datenbank mit der Verwendung von Prisma oder laden Daten aus anderen REST-API Schnittstellen mithilfe von eigens dafür angelegten Services.

So gibt es zum Beispiel den Service `src/service/getEmployees.js`, welcher dazu verwendet wird, um alle derzeit aktiven Mitarbeiter zu laden.

Hilfsfunktionen, wie zum Beispiel das Prüfen, ob ein übergebener Wert in ISO-Date-String ist, können im Ordner `src/util/` gefunden werden.

## Frontend

Das Frontend ist ein Next.js Frontend, welches mithilfe von Antd, Tailwind und Tremor entwickelt wurde.

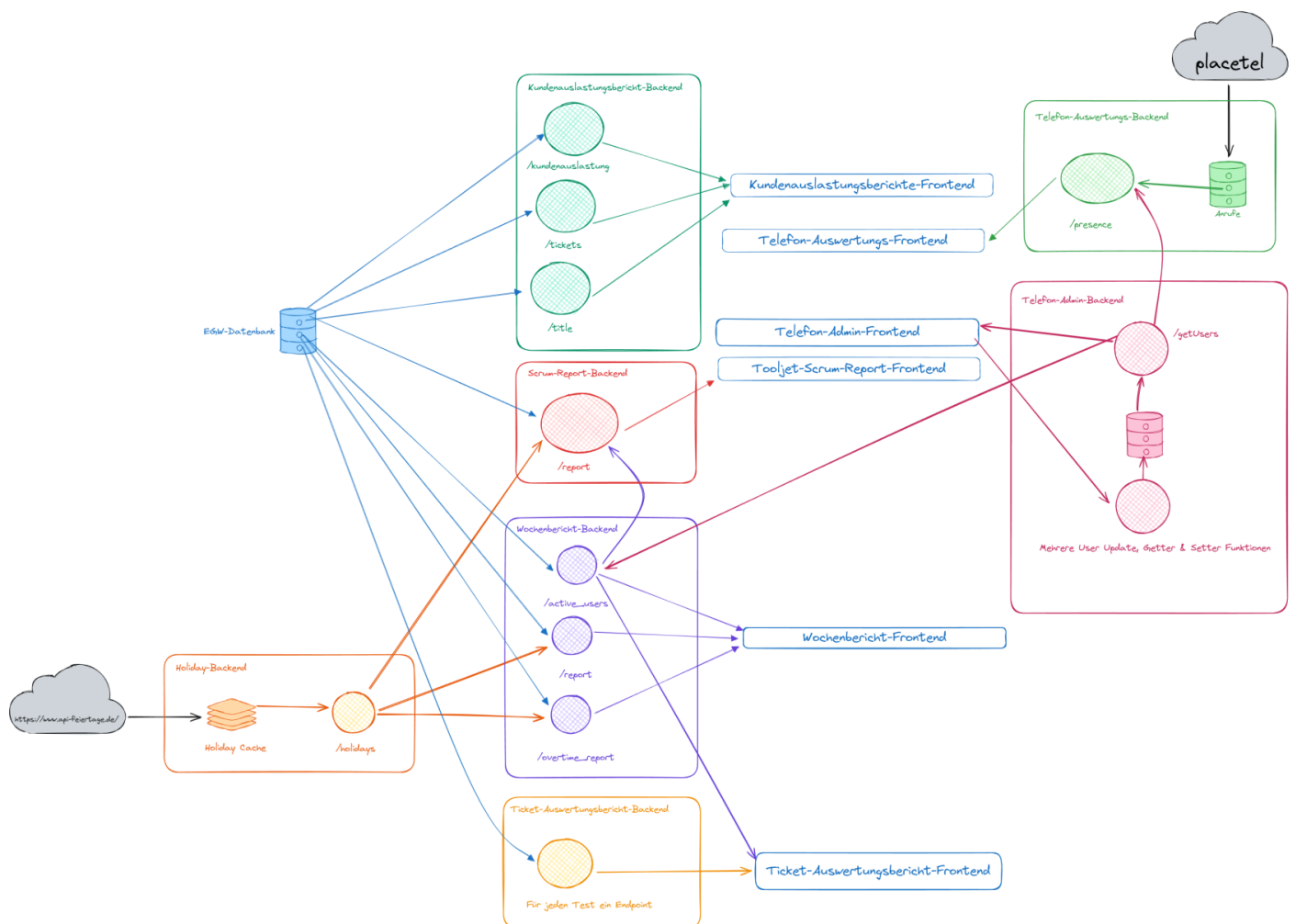
Jede einzelne Statistik soll dort als Karte eingefügt werden. Und diese Karten werden dabei untereinander geschrieben. Dadurch wird ab einem gewissen Punkt ein Scrollbalken am rechten Rand entstehen.

## Projekt-Aufbau

Jede weitere Statistik soll als Komponente im Ordner Components hinzugefügt werden. Diese werden dann im Ordner pages innerhalb der index.jsx aufgerufen.

Alle Funktionen, die Daten aus einer REST-API laden, sollen unter dem Ordner Service gesammelt werden.

# Datenfluss in internen Berichten



Datei zum überarbeiten der Grafik: [InterneBerichte.excalidraw](#) diese kann in excalidraw geöffnet werden.

## Legende

In dieser Grafik, wird dargestellt, wie Daten durch die internen Berichte fließen.

Jeder Pfeil in der Grafik stellt dar, von wo Daten kommen und wohin sie fließen.

Die einzelnen farbigen Kästen sind Rest-API Backends. Die Kreise in den Kästen stellen Rest-API Endpoints dar. Manche Backends besitzen zudem noch eine eigene Lösung, um Daten zu

speichern. So haben manche Backends eine eigene Datenbank oder verwenden einen Cache.

Die blauen schmalen Kästen, in der Grafik, stellen die einzelnen Frontends dar, welche ein Benutzer aufrufen kann.

Zudem gibt es noch außerhalb der Kästen externe Datenquellen, auf welche zugegriffen wird.

Die Wolken stellen dar, dass auf Daten aus dem Internet zugegriffen wird.