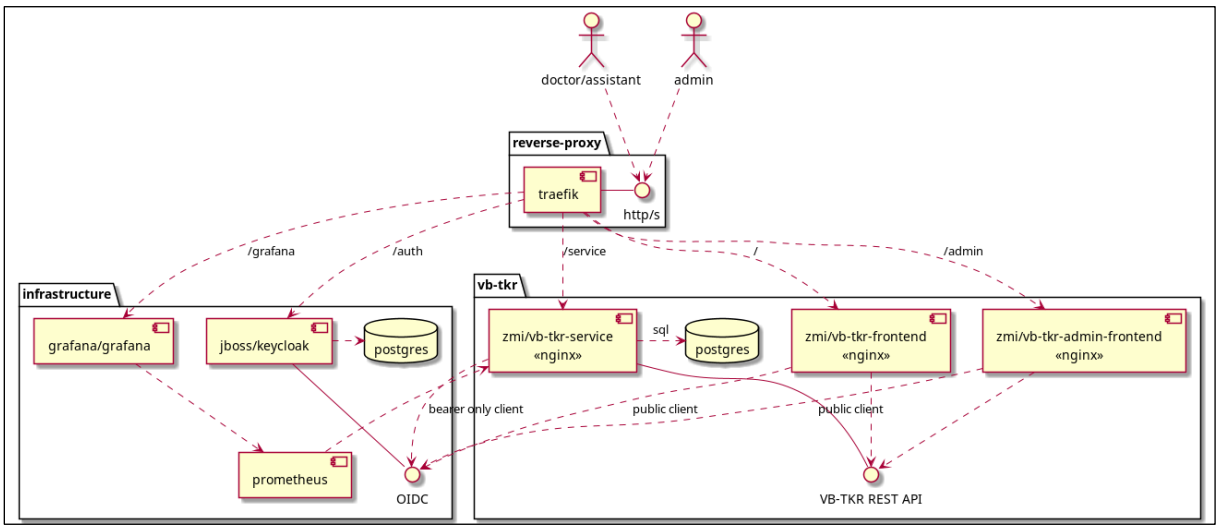
1. Architektur / Deployment-Schema



Das im Value-basedTKR entwickelte EKIT-Tool läuft als containerisierte Anwendung auf einem Docker-Host. Im Produktivbetrieb besteht die Anwendung aus den folgenden Docker-Containern, die über Docker-Compose-Files orchestriert werden:

- Admin-Frontend
- Frontend
- Backend
- Datenbank der Anwendung
- Keycloak
- Datenbank Keycloak
- Grafana
- Prometheus
- Systemvoraussetzung: Traefik

2. Betriebsumgebung

Betriebssystem

- Betriebssystem: Linux
- Distribution: Ubuntu
- Version: 20.x (z. B. 20.04 LTS)

Hardware-Anforderungen

- CPU: Quad-Core Prozessor, > 2 GHz
- RAM: 16 GB
- Festplattenspeicher: 50 GB verfügbarer Speicher
- Architektur: x86_64

Softwareabhängigkeiten / Laufzeitumgebung

- Containerplattform: Docker
- Orchestrierung: Docker Compose
- Reverse Proxy: Traefik v2.4

Benutzerrechte / Systemkonfiguration

- Initiale Benutzer werden bei der Inbetriebnahme automatisch angelegt
- Passwörter werden vor der Inbetriebnahme definiert
- Nutzer können über die Admin-GUI manuell angelegt werden
- Nutzer müssen zwingend einem Standort zugeordnet werden

Umgebungsvariablen / Konfigurationsdateien

- Deployment-State: dev, test, prod
- Datenbank: Postgres 13
- Secrets-Verzeichnis: Pfad zu vertraulichen Dateien
- Netzwerkname: Name des Docker-Netzwerks
- Hostname: Angabe des Hostnamens

Sicherheitsanforderungen

- Lokale Root-Zertifikate (SSL-Verschlüsselung) bereits in das Deployment integriert

Software-Aktualisierung

- Aktualisierungen: Über Docker-Images
- Quelle: Docker-Registry, aus der die neuesten Images heruntergeladen werden

Monitoring / Logging

- Optional: Prometheus und Grafana, entsprechenden Komponenten werden mit der Software ausgeliefert

Fehlermeldung bzw. Fehlerbehandlung

- Initiale Meldung an den 1st-Level Kontakt (z.B. Studienleiter)
- 1st-Level-Kontakt prüft und leitet Fehler per E-Mail an den Anwendungsbetreuer / Entwickler

Kompatibilität

- Keycloak: Wird mitgeliefert, eigenes Keycloak kann verwendet werden
- Kompatibilität: Das Programm ist nur mit Keycloak 20 kompatibel

3. Übersicht über den verwendeten Toolstack auf der Entwicklungsumgebung

Ziel der Umgebung

Modulare Entwicklung der Anwendung VB-TKR (Module: Frontend, Admin-Frontend, Backend)

Verwendete Technologien und Tools:

- **Hardware-Anforderungen:** Quad-Core CPU, 2 > GHz, 16 GB RAM, 50GB HDD
- **OS Entwicklungsumgebung:** Ubuntu 20, Python 3.8, OpenJDK-JRE 8
- **IDE's / Quellcode-Editoren** Pycharm, Visual Studio Code
- **Debugging-Werkzeuge**: Vue Devtools
- **Programmiersprachen**: Python 3.8, Typescript
- **Frameworks**: Vue.JS 2, Flask 2, Open-Api, SQLAlchemy 1.3, yoyo migration (Datenbankmigration)
- **Paketmanager**: npm, pip
- **Verwendete Code-Generatoren**: Open-Api-Generator CLI, Marshmallow
- **Testtools und Frameworks**: Cypress (UI-Tests), Pytest (Unit-Tests, Integrationstests)
- **Versionskontrollsystem**: Git
- **Build-Tools:** GitLab CI, Pybuilder
- **Containerisierung:** Docker v2, Docker-Compose v3.7
- **Datenbanken:** Posgres 13
- **Monitoring:** Grafana, Prometheus
- **Authentifizierung:** Keycloak 20
- **IT-Sicherheit / Zugriffsschutz:** CORS (Reglementierung der Zugriffe auf den Webserver), JWT (Authorisierung/Autentifizierung), Trivy (CVE-Scan), Traefik 2.4(Reverse-Proxy)

Für das entwickelte Programm wurden zahlreiche Drittbibliotheken verwendet. Diese werden durch Paketmanagement-Tools verwendet. Pro Modul und Paketmanager werden die jeweils verwendeten Drittbibliotheken inklusive ihrer Version in den Quellcode-Dateien hinterlegt. Damit werden die Abhängigkeiten im Quellcode bereits automatisch dokumentiert. Die wichtigsten Abhängigkeiten sind:

Für den Betrieb von Frontend / Admin-Frontend

- Node.JS: 15.5.0
- Vue: 2.6.11
- Vuetify: 2.4.0
- vue-router: 3.2.0
- vuex: 3.4.0
- vuex-persist: 2.2.0
- generator-cli: 5.0.0
- Axios: 0.21.1
- openapi-client-axios: 3.7.8
- jwt-decode: 3.1.2
- development: openapitools/openapi-generator-cli: 2.1.16

Für den Betrieb des Python/Flask-Backends

- Flask: 2.2.2
- Flask-Cors: 3.0.10
- SQLAlchemy: 1.3.23
- psycopg2-binary: 2.8.6
- yoyo-migrations: 8.2.0
- openapi-generator-cli: 4.3.1
- connexion[swagger-ui]: 2.7.0
- marshmallow: 3.10.0
- swagger-marshmallow-codegen: 0.6.4

4. Externe Schnittstellen

REST-API des lokalen Treuhandstellen-Dispatchers

Identifizierende Daten (IDATs: Name, Geschlecht, Geburtdatum, Versicherungsnummer, Anschrift, Einwilligungsstatus Studie und Register) werden an den Dispatcher der Treuhandstelle gesendet, THS übermittelt Probanden-ID für Studie zurück. Der Dispatcher ist der Workflow-Manager für die THS-Tools, also E-PIX, gPAS und gICS. Im E-PIX werden die IDATs gespeichert, im gICS die Consentdaten und im gPAS die Pseudonyme generiert.

Keycloak

Tool zur Verwaltung von Benutzeranmeldungen und -rechten, Keycloak verwendet den Standard OIDC (OpenID Connect), um die Authentifizierung und Autorisierung von Benutzern zu handhaben. OIDC ist eine Erweiterung des OAuth 2.0-Protokolls und dient dazu, die Identität eines Benutzers sicher zu überprüfen. Dabei dient Keycloak als Identity Provider (IdP), der den Anmeldeprozess übernimmt. Der Vorteil von OIDC ist, dass es eine standardisierte Methode bietet, um Benutzerdaten über ein sicheres Protokoll auszutauschen – auch über Programmmodule und Anwendungen hinweg.

CSV-Export-Schnittstelle

Erstellung von CSV-Dateien: locations.csv (an der Studie teilnehmende Standorte), querstionnaires.csv (ggf. gefiltert nach Bearbeitungsstatus), patients.csv (Patientenliste ggf. nach Standort gefiltert), cancelled.csv (Informationen zu allen abgebrochen Befragungen)

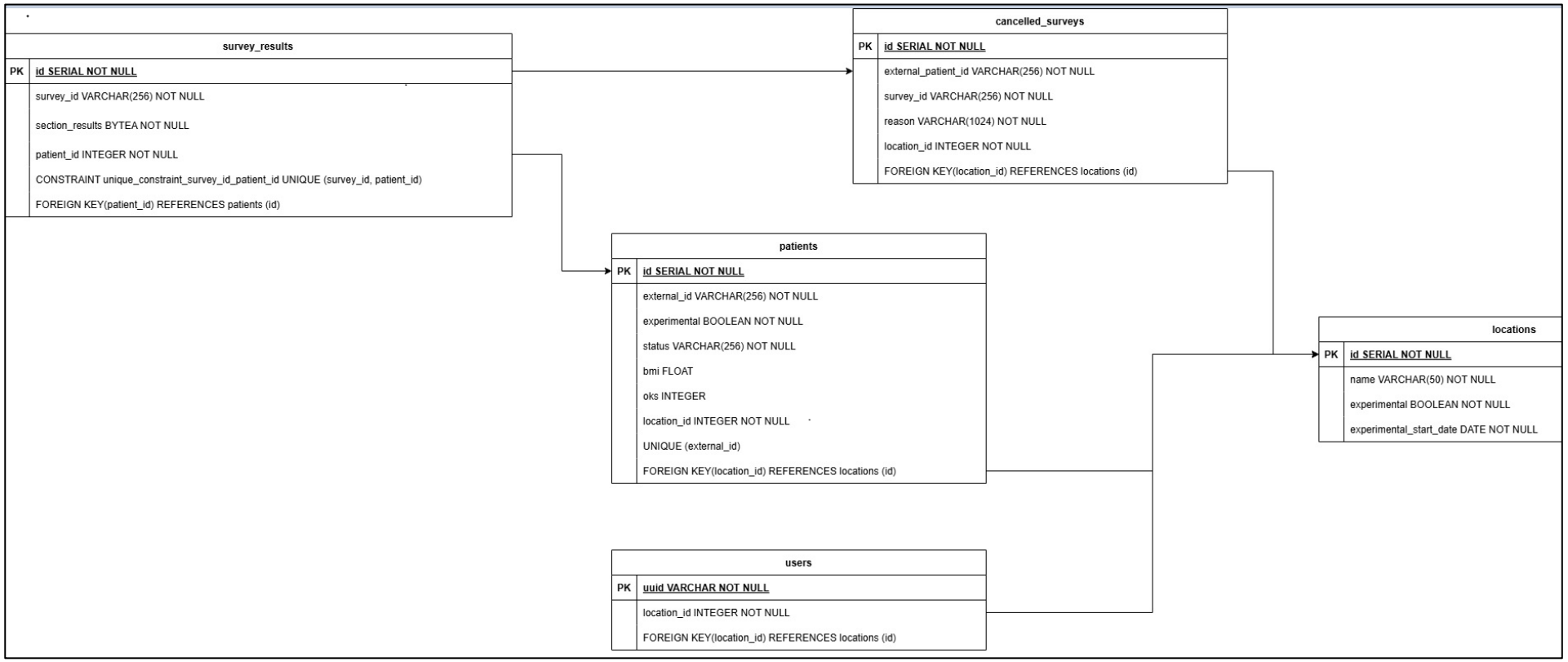
API-Spezifikation / API-Endpunkte

Die OpenAPI-Spezifikation (OAS) definiert eine standardisierte, sprachunabhängige Schnittstelle für HTTP-APIs, die sowohl für den Menschen verständlich als auch maschinenlesbar ist. Eine OpenAPI-Beschreibungen wurde bei der Programmierung verwendet, um die API-Dokumentationen zu erstellen, Code für die verschiedenen Programmmodule bzw. in verschiedenen Programmiersprachen zu generieren sowie Tests durchzuführen.

Die im Folgenden aufgezählten REST-Endpunkte werden innerhalb der Anwendung verwendet, um die Kommunikation zwischen den verschiedenen Programmmodulen (Frontend, Admin-Frontend, Backend) zu realisieren. Eine vollständige Dokumentation der API kann der Anlage 3.1 entnommen werden.

- Benutzer (User) Endpunkte:
  - Benutzer-Objekte abfragen, anlegen, editieren, löschen.
- Standort (Location) Endpunkte:
  - Teilnehmende Studieneinrichtungen abfragen, anlegen, ändern und löschen.
- Patienten (Patient) Endpunkte
  - Studienteilnehmer abfragen, filtern, anlegen, ändern, löschen.
- Umfrageergebnisse (Survey Result) Endpunkte:
  - Ergebnisse der Befragung bzw. einzelner Abschnitte abfragen, filtern, anlegen, ändern, löschen.
- THS Patienten (THS Patients) Endpunkte:
  - Anlegen eines neuen Patienten über die Treuhandstellen-Schnittstelle und Rückgabe der THS-ID.
- Abgebrochene Umfragen (Cancelled Surveys) Endpunkte:
  - Anlegen eines Datensatzes je abgebrochener Umfrage.
- Standortmetriken (Location Metrics) Endpunkte:
  - Abfragen der Metriken je teilnehmender Studieneinrichtung.
- Anwendungsmetriken (Application Metrics) Endpunkte:
  - Abfrage applikationsspezifischer Metriken / Monitoring.
- Export-Endpunkte:
  - CSV-Export für angeforderte Tabellen (z.B. locations, patients, questionnaires, cancelled info)

5. Datenmodell (Datenbank-Ebene)



6. Quellcode

Die Veröffentlichung des Quellcodes erfolgt nur unter Einschränkungen und auf ausdrückliche Anfrage. Der Code ist proprietär, wird nicht weiter gepflegt und ist eng an die spezifische IT-Infrastruktur der Klinik am Standort Dresden gebunden. Er basiert auf standortspezifischen Konfigurationen sowie lokalen IT-Systemen und kann daher nicht ohne erhebliche Anpassungen in anderen Umgebungen betrieben werden.

Die implementierten Sicherheitsmechanismen orientieren sich am Stand von Technik und Wissen zur Projektdurchführung und wurden auf die lokalen Gegebenheiten abgestimmt. Eine Verwendung außerhalb dieser Umgebung birgt potenzielle Sicherheitsrisiken. Darüber hinaus ist die Software nicht mehr in Betrieb, sodass enthaltene Komponenten keine aktuellen Sicherheitsupdates mehr erhalten. Es existiert keine technische Dokumentation für Dritte, was eine Nachvollziehbarkeit und Weiterentwicklung durch externe Entwickler:innen erheblich erschwert. Für die Nutzung sind tiefgehende Kenntnisse der klinikspezifischen IT-Landschaft erforderlich.

Ein weiteres Risiko sehen wir in einem möglichen Einsatz der Software im Kontext der medizinischen Versorgung. Eine nicht autorisierte Weiterverwendung oder Modifikation könnte dazu führen, dass die Software als Medizinprodukt im Sinne des Medizinprodukterechts eingestuft wird. In diesem Fall bestünde die Gefahr, dass wir als Inverkehrbringer betrachtet werden – mit entsprechenden haftungsrechtlichen Konsequenzen. Dieses Risiko möchten wir ausdrücklich vermeiden.

Aus diesen Gründen erfolgt eine Herausgabe des Quellcodes **nur auf Anfrage** und unter Anerkennung spezifischer Nutzungsbedingungen. Dazu zählt insbesondere die schriftliche Bestätigung, dass der Code ausschließlich zu Forschungszwecken verwendet wird und nicht für eine klinische Nutzung vorgesehen ist. Die Herausgabe erfolgt über eine zentrale Kontaktstelle:

Dipl.-Ing. (FH) Daniela Barnett
Teamleiterin Softwareentwicklung
Datenintegrationszentrum - DIZ
Zentrum für Medizinische Informatik
Universitätsklinikum Dresden
Daniela.Barnett@ukdd.de

Unser Ziel ist es, Transparenz über die bestehenden Einschränkungen zu schaffen und potenzielle Risiken durch eine nicht sachgerechte Nutzung zu minimieren.

VB-TKR-Rest-Api 0.0.1 OAS 3.0

user All endpoints for users

GET /users Returns a list of users.

Parameters Try it out

No parameters

Responses

Code	Description	Links

200	An array of users.
	Media type: application/json
	Controls Accept header.
	Example Value Schema
	<pre>[
	  {
	  "uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
	  "location-id": 0
	  }
	]</pre>
		No links

POST /users Creates a new user

Parameters Try it out

No parameters

Request body application/json

Example Value Schema

<pre>{
  "uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "location-id": 0
}</pre>

Responses

Code	Description	Links
201	response	*No links*

GET /users/{user-uuid} Returns user object for a given user uuid

Parameters Try it out

Name	Description
user-uuid * required string($uuid) *(path)*	User UUID user-uuid

Responses

Code	Description	Links

200	The user object
	Media type: application/json
	Controls Accept header.
	Example Value Schema
	<pre>{
	  "uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
	  "location-id": 0
	}</pre>
		No links

PUT /users/{user-uuid} Updates an user

Parameters

Try it out

Name	Description
user-uuid * `string($uuid)` *(path)*	User UUID user-uuid

Request body — application/json

Example Value Schema

```
{
  "uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "location-id": 0
}
```

Responses

Code	Description	Links
200	The user object Media type: application/json Controls Accept header. **Example Value** Schema `{ "uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "location-id": 0 }`	*No links*

DELETE `/users/{user-uuid}` Deletes an user

Parameters Try it out

Name	Description
user-uuid * `string($uuid)` *(path)*	User UUID user-uuid

Responses

Code	Description	Links
204	If the user was deleted	*No links*

location All location endpoints

GET `/locations` Returns locations

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	A list of locations Media type: application/json Controls Accept header. **Example Value** Schema `[ { "id": 0, "name": "string", "experimental": true, "experimental-start-date": "2025-05-19" } ]`	*No links*

POST `/locations` Creates a location

Parameters Try it out

No parameters

Request body

application/json

Example Value Schema


```
{
  "id": 0,
  "name": "string",
  "experimental": true,
  "experimental-start-date": "2025-05-19"
}
```


Responses

Code	Description	Links
200	Created location Media type: application/json Controls `Accept` header. **Example Value** Schema	*No links*


```
{
  "id": 0,
  "name": "string",
  "experimental": true,
  "experimental-start-date": "2025-05-19"
}
```


GET `/locations/{location-id}` Returns location object for a given location id

Parameters Try it out

Name	Description
location-id * `integer` *(path)*	Location id location-id

Responses

Code	Description	Links
200	The location object Media type: application/json Controls `Accept` header. **Example Value** Schema	*No links*


```
{
  "id": 0,
  "name": "string",
  "experimental": true,
  "experimental-start-date": "2025-05-19"
}
```


PUT `/locations/{location-id}` Updates a location object

Parameters Try it out

Name	Description
location-id * `integer` *(path)*	Location id location-id

Responses

Code	Description	Links
200	updated The location object Media type: application/json Controls `Accept` header. **Example Value** Schema	*No links*


```
{
  "id": 0,
  "name": "string",
  "experimental": true,
  "experimental-start-date": "2025-05-19"
}
```


DELETE `/locations/{location-id}` Deletes a locaiton

Parameters Try it out

Name	Description
location-id * `integer` *(path)*	Location id location-id

Responses

Code	Description	Links
204	If the location was deleted successfully	*No links*

patient All patient endpoints

GET `/patients` Returns a list of patients.

Parameters Try it out

Name	Description
external-id `string` *(query)*	External id of the patient external-id
status `string` *(query)*	Status string to match against status
status-doctor `string` *(query)*	Status-doctor string to match against status-doctor

Responses

Code	Description	Links
200	An array of patients. Media type: application/json Controls `Accept` header. **Example Value** Schema	*No links*


```
[
  {
    "id": 0,
    "external-id": "string",
    "location-id": 0,
    "experimental": true,
    "status": "init",
    "status-updated": "2025-05-19T07:41:28.695Z",
    "status-doctor": "created",
    "status-doctor-updated": "2025-05-19T07:41:28.695Z",
    "bmi": 0,
    "oks": 0
  }
]
```


POST `/patients` Creates a new patient

Parameters Try it out

No parameters

Request body application/json

Example Value Schema


```
{
  "id": 0,
  "external-id": "string",
  "location-id": 0,
  "experimental": true,
  "status": "init",
  "status-updated": "2025-05-19T07:41:28.631Z",
  "status-doctor": "created",
  "status-doctor-updated": "2025-05-19T07:41:28.631Z",
  "bmi": 0,
  "oks": 0
}
```


Responses

Code	Description	Links
201	response	*No links*

GET `/patients/{patient-id}` Returns patient object for a given patient id

Parameters Try it out

Name	Description
patient-id * `integer` *(path)*	Patient id patient-id

Responses

Code	Description	Links
200	The patient object Media type application/json Controls Accept header. **Example Value** Schema { "id": 0, "external-id": "string", "location-id": 0, "experimental": true, "status": "init", "status-updated": "2025-05-19T07:41:28.696Z", "status-doctor": "created", "status-doctor-updated": "2025-05-19T07:41:28.696Z", "bmi": 0, "oks": 0 }	*No links*

PUT `/patients/{patient-id}` Updates a patient

Parameters Try it out

Name	Description
patient-id * `integer` *(path)*	Patient id patient-id

Request body application/json

Example Value Schema

{ "id": 0, "external-id": "string", "location-id": 0, "experimental": true, "status": "init", "status-updated": "2025-05-19T07:41:28.637Z", "status-doctor": "created", "status-doctor-updated": "2025-05-19T07:41:28.637Z", "bmi": 0, "oks": 0 }

Responses

Code	Description	Links
200	The user object Media type application/json Controls Accept header. **Example Value** Schema { "id": 0, "external-id": "string", "location-id": 0, "experimental": true, "status": "init", "status-updated": "2025-05-19T07:41:28.697Z", "status-doctor": "created", "status-doctor-updated": "2025-05-19T07:41:28.697Z", "bmi": 0, "oks": 0 }	*No links*

DELETE `/patients/{patient-id}` Deletes a patient

survey-result All survey result endpoints

GET `/survey-results` Returns a list of survey results.

Parameters Try it out

Name	Description
patient `integer` *(query)*	id of the patient, will return only the ones that match patient
survey `string` *(query)*	survey id to filter for survey

Responses

Code	Description	Links
200	An array of results. Media type: application/json Controls `Accept` header. **Example Value** Schema	*No links*

```
[
  {
    "id": 0,
    "survey-id": "string",
    "patient-id": 0,
    "section-results": {
      "additionalProp1": "string",
      "additionalProp2": "string",
      "additionalProp3": "string"
    }
  }
]
```

POST `/survey-results` Creates a new survey result

Parameters Try it out

No parameters

Request body — application/json

Example Value Schema

```
{
  "id": 0,
  "survey-id": "string",
  "patient-id": 0,
  "section-results": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
}
```

Responses

Code	Description	Links
201	response	*No links*

GET `/survey-results/{survey-result-id}` Returns the survey result object for a given survey result id

Parameters Try it out

Name	Description
survey-result-id * required `integer` *(path)*	Survey result id survey-result-id

Responses

Code	Description	Links
200	The patient object Media type: application/json Controls `Accept` header. **Example Value** Schema	*No links*

```
{
  "id": 0,
```

Code	Description	Links


```
  "survey-id": "string",
  "patient-id": 0,
  "section-results": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
}
```


PUT /survey-results/{survey-result-id} Updates a survey by its id

Parameters Try it out

Name	Description
survey-result-id * `integer` *(path)*	Survey result id survey-result-id

Request body: application/json

Example Value Schema


```
{
  "id": 0,
  "survey-id": "string",
  "patient-id": 0,
  "section-results": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
}
```


Responses

Code	Description	Links
200	survey result object Media type: application/json Controls `Accept` header. **Example Value** Schema	*No links*


```
{
  "id": 0,
  "survey-id": "string",
  "patient-id": 0,
  "section-results": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
}
```


DELETE /survey-results/{survey-result-id} Deletes a survey result

Parameters Try it out

Name	Description
survey-result-id * `integer` *(path)*	Survey result id survey-result-id

Responses

Code	Description	Links
204	If the survey result was deleted	*No links*

GET /survey-results/{survey-result-id}/section-results/{section-result-id} Returns the survey section result object for a given survey result id and section id

Parameters Try it out

Name	Description
survey-result-id * `integer` *(path)*	Survey result id survey-result-id
section-result-id * `string` *(path)*	Section result id section-result-id

Responses

Code	Description	Links
200	The section object Media type: application/json Controls Accept header. Example Value Schema { "additionalProp1": {} }	No links

PUT /survey-results/{survey-result-id}/section-results/{section-result-id} Updates a survey section results by its id

Parameters Try it out

Name	Description
survey-result-id * integer (path)	Survey result id survey-result-id
section-result-id * string (path)	Section result id section-result-id

Request body application/json

Example Value Schema

```
{
  "additionalProp1": {}
}
```

Responses

Code	Description	Links
200	The survey section result object Media type: application/json Controls Accept header. Example Value Schema { "additionalProp1": {} }	No links

DELETE /survey-results/{survey-result-id}/section-results/{section-result-id} Deletes a survey section result by its id

Parameters Try it out

Name	Description
survey-result-id * integer (path)	Survey result id survey-result-id
section-result-id * string (path)	Section result id section-result-id

Responses

Code	Description	Links
204	SurveySectionResult was deleted	No links

application-metrics All endpoints connected to application metrics

GET /location-metrics Returns a list of location application metrics.

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	An array of location metrics. Media type: application/json Controls Accept header. **Example Value** Schema	*No links*

```
[
  {
    "location-id": 0,
    "patients-completed-surveys": 0,
    "patients-cancelled-surveys": 0,
    "patients-count": 0,
    "success-rate": 0,
    "users-count": 0
  }
]
```

GET `/application-metrics` Returns the sum of all location specific metrics.

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Metric for the complete application (aggregates all locations) Media type: application/json Controls Accept header. **Example Value** Schema	*No links*

```
{
  "patients-completed-surveys": 0,
  "patients-cancelled-surveys": 0,
  "patients-count": 0,
  "success-rate": 0,
  "locations-count": 0,
  "users-count": 0
}
```

patientThs

POST `/ths-patients` Creates a new patient in THS rest-API

Parameters Try it out

No parameters

Request body application/json

Example Value Schema

```
{
  "locationStudyCenter": "string",
  "firstName": "string",
  "lastName": "string",
  "title": "string",
  "birthDate": "string",
  "insuranceNumber": "string",
  "gender": "M",
  "street": "string",
  "postalCode": "string",
  "city": "string",
  "consentEprd": "accepted",
  "consentDdRegister": "accepted"
}
```

Responses

Code	Description	Links
201	return THS ID of the created patient Media type: application/json Controls Accept header. **Example Value** Schema	*No links*

```
{
  "thsId": "string"
}
```

cancelled-survey

POST `/cancelled-surveys` Creates an entry for a cancelled questionnaire

Parameters Try it out

No parameters

Request body application/json

Example Value Schema

```
{
  "id": 0,
  "external-patient-id": "string",
  "location-id": 0,
  "survey-id": "t0",
  "reason": "string"
}
```

Responses

Code	Description	Links
201	response Media type: application/json Controls Accept header. **Example Value** Schema `{}`	*No links*

default

GET `/export` Returns a Csv file of the requested table

Parameters Try it out

Name	Description
requested string *(query)*	*Available values* : locations, patients, questionnaires, cancelled_info --

Responses

Code	Description	Links
200	returns a test string Media type: text/csv Controls Accept header. **Example Value** Schema `string`	*No links*

Schemas

UserDTO {

description:	Users are doctors or assistants in the sense of the api
uuid	string($uuid) UUID of the user that is able to identify a user within the used external authentication provider (i.e. Keycloak)
location-id	integer($int64) Id of the location the user belongs to

}

PatientDTO {

id	integer($int64) Integer based id for the patient
external-id	string External id for the patients identity, will be supplied in anonymised form
location-id	integer($int64) Id of the location the patient belongs to
experimental	boolean Whether or not the patient is part of the experimental group
status	string Status of the patient Enum: Array [10]
status-updated	string($date-time) Timestamp of the last update of the patients status field
status-doctor	string Status of the patient for the doctor (T1 for doctor and patient can be at the same time) Enum: Array [9]
status-doctor-updated	string($date-time) Timestamp of the last update of the patients status-doctor field
bmi	number Body mass index of the patient
oks	integer Oxford knee score

}

SurveyResultDTO {

id	integer($int64) Integer based id for the identification of one survey result
survey-id	string Identifier for a survey, e.g. t0, t1
patient-id	integer($int64) Identifier for a user

section-results	{
	description: Map of section results
	
	}

}

LocationDTO {

description:	A location is a logical group for the participants of a study
id	integer($int64) Integer based id for the object
name	string Name of the location
experimental	boolean Whether or not the location is experimental state
experimental-start-date	string($date) Start date of the experimental phase

}

PatientThsDTO {

locationStudyCenter	string name of the study center that admitted the THS-patient
firstName	string first name of THS-patient
lastName	string last name of THS-patient
title	string optional: title of THS-patient
birthDate	string birth date of THS-patient in format yyyy-mm-dd
insuranceNumber	string health insurance number of THS-patient
gender	string gender of THS-patient (M=male, F=female, O=other) Enum: Array [3]
street	string street and house nr. of THS-patient
postalCode	string postal code of THS-patient
city	string city name of THS-patient
consentEprd	string whether the THS-patient provided consent to send data to EPRD Enum: Array [2]
consentDdRegister	string just for Dresden patients, whether the THS-patient provides consent to send data to the local register Enum: Array [2]

}

CancelledSurveyDTO {

id	integer($int64) Integer based id for the object
external-patient-id	string External patient id, which is not a foreign key, in case the patient needed to be removed
location-id	integer($int64) Id of the location the patient belongs to
survey-id	string Identifies the cancelled survey Enum: Array [4]
reason	string Patient Reason for cancelling the survey

}

LocationMetricDTO {

location-id	integer($int64) Id of the location
patients-completed-surveys	integer Amount of patients that have completed the surveys
patients-cancelled-surveys	integer Amount of users that have cancelled the surveys
patients-count	integer Amount of users for the location
success-rate	number Fraction of patients completed the surveys (status 't1-finished', status_doctor 't1-finished') in comparison with the total amount of users for the location, between 0 and 1.
users-count	integer Amount of users

}

ApplicationMetricDTO {

patients-completed-surveys	integer Amount of patients that have completed the surveys
patients-cancelled-surveys	integer Amount of users that have cancelled the surveys
patients-count	integer Amount of patients
success-rate	number Fraction of patients completed the surveys (status 't1-finished', status_doctor 't1-finished') in comparison with the total amount of users for all locations, between 0 and 1.
locations-count	integer Amount of locations
users-count	integer Amount of users

}