

1.2.0 Dokumentation

Konfiguration

TI-CA Zertifikate

Umgebungsvariablen

URLs der adressierten Systeme

Konfigurationen TIaaS Arbeitsumgebung

DIGA Informationen

ePA Umgebung

Uvicorn Worker

ePA Provider Mapping

Endpunkt

API-Übersicht:

API-Dokumentation: POST /send-document/

Request Body

Beispiel Request in Python:

Rückgabewerte /send-document

ErrorCodes /send-document

API-Dokumentation: POST /localize-record/

Request Body

Beispiel Request in Python

ErrorCodes /localize-record:

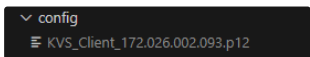
 Gültig ab Version: 1.2.0

Dies ist die Dokumentation für die Verwendung des “**_fbeta ePA Service**”. Im Folgenden werden wichtige Konfigurationsdetails aufgeführt und erläutert. Anschließend werden die Details der API-Schnittstelle beschrieben und ein Beispiel für die Integration in ein Produktivsystem gegeben.

Konfiguration

TI-CA Zertifikate

Legen Sie Ihr p12-Zertifikat in das Verzeichnis `/config` ab. Das Zertifikat wird bei Operationen die mit dem Konnektor kommunizieren mit einbezogen. Lediglich das Passwort für das Zertifikat muss als Umgebungsvariable `KONNEKTOR_CERT_PW` hinterlegt werden.



Umgebungsvariablen

Folgende Umgebungsvariablen müssen in der Datei `.env` gesetzt werden:

URLs der adressierten Systeme

- `KONNEKTOR_URL` : URL des Konnektors
(`'https://konnektor-2.ru.tiaas.rise-ti.de'`)
- `USER_AGENT` : User Agent String für HTTP-Requests
(`'usrAgent/1.1.0'`)
- `KONNEKTOR_CERT_PW` : Passwort für das p12 Zertifikat

Konfigurationen TIaaS Arbeitsumgebung

- `MANDANT_ID` : ID des eingerichteten Mandanten im Konnektor
(`'Test'`)
- `CLIENT_SYSTEM_ID` : ID des eingerichteten Client-Systems im Konnektor
(`'Test_Clientsystem'`)
- `WORKPLACE_ID` : ID des eingerichteten Arbeitsplatzes im Konnektor
(`'Test-Arbeitsplatz'`)
- `USER_ID` : ID des eingerichteten Users (Optional)

DIGA Informationen

- `DIGA_NAME` : Name Ihrer DiGA (Name der Verordnungseinheit)
(`'DIGA-Hersteller'`)
- `DIGA_MANUFACTURER` : Name des DiGA-Herstellers
(`'Sina Gr..fin ..zTEST-ONLY'`)

ePA Umgebung

- `EPA_ENVIRONMENT` : ePA-Umgebung `RU (RU REF)` , `RT (RU DEV)` or `PROD`
- (**Deprecated** ab Version 1.1.2 - Provider wird automatisch aus dem Provider-Mapping bezogen)
`EPA_PROVIDER_NUMBER` : Nummer für ePA-Provider (`IBM=1`, `RISE=2`) (`epa-as-<provider-number>`)

Uvicorn Worker

- `UVICORN_WORKERS` : Anzahl der API-Worker für eine parallelisierte Verarbeitung (Default-Wert: 1)

ePA Provider Mapping

```
1 RECORD_PROVIDER_1 = 'IBM'
2 RECORD_PROVIDER_2 = 'Bitmarck Technik'
3 RECORD_PROVIDER_X = 'PROVIDERNAME' [...]
```

Endpunkt

API-Übersicht:

Eine Übersicht aller API-Endpunkte, sowie Parameter sind über folgende Endpunkte im Container zu erhalten ("Swagger"):

- `/redoc`
- `/docs`
- `/openapi.json`

API-Dokumentation: `POST /send-document/`

Dieser Endpunkt dient zum Senden eines medizinischen Dokuments (MIO) an die ePA. Die Anfrage enthält die KVNR (insurent_id) der versicherten Person, einen SubmissionSet-Eintrag, einen Dokumenteneintrag sowie die XML-Daten des Dokuments.

`POST http://localhost:8000/send-document/`

Request Body

Der Request muss im JSON-Format gesendet werden. Die Felder im Payload sind wie folgt:

Feldname	Typ	Beschreibung
<code>metadata.insurantId</code>	<code>string</code>	KVNR der versicherten Person
<code>metadata.submissionSet</code>	<code>object</code>	Objekt mit Metadaten zum Submission Set
<code>metadata.submissionSet.entryUUID</code>	<code>string</code>	Eindeutige UUID des Submission Sets
<code>metadata.documentEntry</code>	<code>object</code>	Objekt mit Metadaten zum eigentlichen Dokument
<code>metadata.documentEntry.creationTime</code>	<code>string</code>	Erstellungszeitpunkt des Dokuments im Format "%Y%m%d%H%M%S" (UTC) (z.B: 20250327165707)
<code>metadata.documentEntry.title</code>	<code>string</code>	Titel des Dokuments
<code>metadata.documentEntry.URI</code>	<code>string</code>	Speicherort oder Name der Datei (z. B. <code>testmio.xml</code>)
<code>metadata.documentEntry.entryUUID</code>	<code>string</code>	Eindeutige UUID für das Dokument Diese UUID sollte von der DiGA gespeichert werden um ggf. eine "replace"-Operation durchzuführen.
<code>metadata.documentEntry.oldEntryUUID</code>	<code>string (optional)</code>	Eindeutige UUID für das Dokument, welches ausgetauscht werden soll (optional)
<code>file</code>	<code>string /bytes</code>	Als Bytes gelesener Inhalt des Dokuments

Beispiel Request in Python:

```

1 import json
2 import os
3 import requests
4 import uuid
5 import time
6
7 # Define the path to the file to be uploaded
8 xml_path = os.path.join(os.path.dirname(__file__), '..', 'data', 'REAL_
9
10 # Generate UUIDs - They should also be safed
11 document_uuid = str(uuid.uuid4())
12 submission_set_uuid = str(uuid.uuid4())
13
14 # Get file creation time from metadata
15 file_creation_time = os.path.getctime(xml_path)
16 # Transform into UTC-Time
17 adjusted_time = time.gmtime(file_creation_time)
18 creation_time = time.strftime("%Y%m%d%H%M%S", adjusted_time)
19
20 url = "http://localhost:8000/send-document/"
21 payload = {
22     "insurantId": "X110580673", # Insert a valid KVN here
23     # See: https://gemspec.gematik.de/docs/gemSpec/gemSpec_Aktensystem_
24     "providerId": 2, # Insert the assigned number of the ePA provider o
25     "submissionSet": {
26         "entryUUID": submission_set_uuid
27     },
28     "documentEntry": {
29         # UTC
30         "creationTime": creation_time,
31         "title": "Testdokument",
32         "URI": "REAL_EXAMPLE_1_KBV_PR_MIO_DIGA_Bundle.xml",
33         "entryUUID": document_uuid,
34         # OPTIONAL - If you want to replace a document
35         # "oldEntryUUID": "929850b6-a5f2-46ca-b742-1abb754cfd45"
36     },
37 }
38
39 print("Payload:", json.dumps(payload, indent=4))
40
41 # Send the request with 120 seconds timeout
42 response = requests.post(url,
43     data={
44         "metadata": json.dumps(payload),
45     },
46     files={
47         "file": open(xml_path, "rb"),
48     },
49     timeout=120
50 )
51
52 response_dict = json.loads(response.text)
53 print("Response: ", json.dumps(response_dict, indent=4))

```

Rückgabewerte /send-document

Code	Bedeutung	Response-Body (JSON)
200	OK – Dokument erfolgreich verarbeitet und verschickt	success : <i>Boolean</i>. Gibt an, ob der Upload erfolgreich war oder nicht. message : <i>String</i>. Zusammenfassung der Antwort der Patientenakte auf den Upload-Versuch. details : <i>Objekt</i>. Die genaue Antwort der Patientenakte auf den Uploadversuch. submissionSetEntryUUID : <i>String</i>. Die entryUUID des Upload-Versuchs (SubmissionSet). documentEntryEntryUUID : <i>String</i>. Die entryUUID des

		versickten Dokuments (DocumentEntry).
422	Unprocessable Content – Falsche oder fehlende Request-Parameter im JSON-Payload	• detail : Liste an Objekten. Enthält Hinweise zu fehlenden oder falsch verwendeten Feldern in der Anfrage. ◦ type : String. ◦ loc : Objekt. ◦ msg : String. ◦ input : Any.
400+	Bad Request – Ungültiger Payload oder fehlende Felder	• detail : ◦ error : String. Kurze Beschreibung des Fehlers. ◦ code : String. Der Name des ErrorCodes. ◦ type : String. Der Name der Komponente, in dem der Fehler aufgetreten ist. ◦ entryUUID : String. Die submissionSet.EntryUUID des Upload-Versuchs (zu debugging Zwecken). ◦ *detail : Dict<String>. Ist ein Fehler auf eine nicht erfolgreiche Antwort des Aktensystems zurückzuführen, wird der konkrete Response des Aktensystems an dieser Stelle angezeigt. Diese Informationen dienen ausschließlich der Unterstützung beim Troubleshooting und haben keinen programmatischen Nutzen.
500+	Internal Server Error – Fehler bei der Verarbeitung auf Serverseite	

ErrorCodes /send-document

Tritt ein Fehler auf, bevor das Dokument an die Patientenakte gesendet werden kann, antwortet der Service mit einem der folgenden ErrorCodes:

ErrorCode	Bedeutung	Verantwortung / Lösung
VAU_INIT_FAILED	Initialisierung des VAU-Kanals zwischen Client und Server fehlgeschlagen	• DiGA: ◦ Verbindung zwischen ePA-API und Konnektor fehlerhaft.

		◦ SMC-B nicht über Konnektor abrufbar/ SMC-B nicht verifiziert • Aktensystembetreiber/ Gematik: Störungen in der TI oder am Aktensystem
VAU_CERT_VALIDATION_FAILED	Validierung des VAU-Server-Zertifikats nicht möglich	• Aktensystembetreiber/ Gematik: Störungen in der TI oder am Aktensystem
VAU_AUTH_FAILED	Authentifizierung über den VAU-Kanal fehlgeschlagen (ungültiges oder abgelaufenes Token)	• DiGA: Fehler im Redis, Token wird nicht aktualisiert. • Aktensystembetreiber/ Gematik: Störungen in der TI oder am Aktensystem
VAU_COMMUNICATION_ERROR	Allgemeiner Fehler beim Senden oder Verarbeiten von VAU-Nachrichten	• DiGA: ◦ Verbindung zwischen ePA-API und Konnektor fehlerhaft. ◦ SMC-B nicht über Konnektor abrufbar/ SMC-B nicht verifiziert • Aktensystembetreiber/ Gematik: Störungen in der TI oder am Aktensystem
EPA_GETNONCE_ERROR	Fehler beim Abrufen eines Nonce vom ePA-Autorisierungsservice	• DiGA: ◦ Verbindung zwischen ePA-API und Konnektor fehlerhaft. ◦ SMC-B nicht über Konnektor abrufbar/ SMC-B nicht verifiziert • Aktensystembetreiber/ Gematik:

		Störungen in der TI oder am Aktensystem
EPA_AUTHZ_ERROR	Autorisierung mit dem ePA-Dienst fehlgeschlagen	• DiGA: ◦ Verbindung zwischen ePA-API und Konnektor fehlerhaft. ◦ SMC-B nicht über Konnektor abrufbar/ SMC-B nicht verifiziert • Aktensystembetreiber/ Gematik: Störungen in der TI oder am Aktensystem
EPA_NOT_ENTITLED	Benutzer hat den Zugriff auf die ePA nicht über die FdV-App freigegeben	User: muss die DiGA in der FdV-App zum schreiben auf die ePA freigeben.
EPA_SEND_ERROR	Fehler beim Senden eines Dokuments an das ePA-System	
KONNEKTOR_INIT_FAILED	Initialisierung der Verbindung zum Konnektor fehlgeschlagen	• DiGA: ◦ Verbindung zwischen ePA-API und Konnektor fehlerhaft.
KONNEKTOR_REQUEST_FAILED	Anfrage an den Konnektor fehlgeschlagen	• DiGA: ◦ Verbindung zwischen ePA-API und Konnektor fehlerhaft.
KONNEKTOR_MALFORMED_REQUEST	An den Konnektor gesendete Anfrage war fehlerhaft	• DiGA: ◦ Fehlerhafte Eingabeparameter
CARD_NOT_FOUND	Die SMC-B-Karte wurde nicht im Kartenterminal gefunden	• DiGA: ◦ SMC-B nicht über Konnektor abrufbar
CARD_CERTIFICATE_ERROR	Fehler beim Abrufen oder Verarbeiten des Karten-Zertifikats	• DiGA: ◦ SMC-B nicht über Konnektor

		abrufbar/ SMC-B nicht verifiziert
CARD_PIN_NOT_VERIFIED	Die PIN der SMC-B-Karte ist nicht verifiziert/aktiviert	- DiGA: - SMC-B nicht verifiziert
CARD_OPERATION_ERROR	Während einer Kartenoperation ist ein Fehler im Konnektor aufgetreten	- DiGA: - SMC-B händisch prüfen - Steckt SMC-B - Wird SMC-B über GetCards richtig gelesen - Wird SMC-B über getPINStatus als verifiziert angezeigt. - Konnektorbetreiber: Störung in der TI oder am Konnektor
DOC_DECODE_ERROR	Fehler beim Dekodieren der Dokumentdaten	Aktensystembetreiber/ Gematik: Störungen in der TI oder am Aktensystem
DOC_MISSING_DATA	Erforderliche Daten für das Dokument fehlen	DiGA: Erforderliche Daten für das Dokument fehlen (URI).
DOC_UPLOAD_DUPLICATE	Dokument besteht bereit in der elektronischen Patientenakte der KVNR. (File-Hash bereits im Aktensystem)	DiGA: MIO oder PDF-A wurde seit dem letzten Upload nicht angepasst, bzw. es wird versehentlich erneut hochgeladen.
DOC_UPLOAD_ERROR	Fehler beim Hochladen des Dokuments in das ePA-System	- DiGA: - Verbindung zwischen ePA-API und Konnektor fehlerhaft. - SMC-B nicht über Konnektor abrufbar/ SMC-B nicht verifiziert

		◦ Fehler im Datenformats des Uploaddokuments. • Aktensystembetreiber/ Gematik: Störungen in der TI oder am Aktensystem
DOC_REPLACE_UUID_ERROR	Das Dokument konnte nicht überschrieben bzw. ersetzt werden, da die angegebene oldEntryUUID keiner im Aktensystem der betreffenden KVNR vorhandenen entryUUID entspricht.	• DiGA: ◦ entryUUID des zu ersetzenden Dokuments wurde nicht korrekt gesichert, wiedergeholt oder gesetzt.
IDP_VERIFICATION_ERROR	Fehler bei der Verifizierung von Tokens oder Signaturen vom Identity Provider	• Gematik: Störungen in der TI oder am zentralen IDP
IDP_REQUEST_ERROR	Anfrage an den Identity Provider fehlgeschlagen	• DiGA: Verbindung zwischen ePA-API und IDP fehlerhaft. • Gematik: Störungen in der TI oder am zentralen IDP
UNEXPECTED_ERROR	Ein unerwarteter Fehler, der nicht in andere Kategorien fällt	-

API-Dokumentation: POST /localize-record/

Der Endpunkt durchsucht systematisch alle konfigurierten Aktensystem-Provider (z.B. die verschiedenen Krankenkassen-Systeme), um herauszufinden, bei welchem Anbieter die elektronische Patientenakte eines bestimmten Versicherten gespeichert ist. Das Ergebnis wird cached, um wiederholte Anfragen zu optimieren.

POST http://localhost:8000/localize-record/

Request Body

Der Request muss im JSON-Format gesendet werden. Die Felder im Payload sind wie folgt:

Feldname	Type	Beschreibung
insurantId	string	Die Versichertennummer

		(KVNR) des Patienten, dessen Akte lokalisiert werden soll
<code>ignoreCache</code>	<code>bool</code>	Boolean-Flag, ob der Cache ignoriert werden soll (optional, Default: <code>false</code>)

Beispiel Request in Python

```

1 url = "http://localhost:8000/localize-record/"
2
3 # Send the request with 120 seconds timeout
4 payload = {
5     "insurantId": "Z987654321",
6     # "ignoreCache": True
7 }
8 response = requests.post(url, json=payload, timeout=120)

```

ErrorCodes /localize-record:

ErrorCodes	Bedeutung	Verantwortung / Lösung
<code>RECORD_EXISTS_AND_ACTIVATED</code>	KVNR ist bei einem der Aktensystembetreiber registriert und aktiviert	-
<code>RECORD_NOT_FOUND</code>	KVNR wurde in keinen der Aktensystembetreiber gefunden.	User: muss sich mit KVNR bei einem Aktensystembetreiber registrieren.
<code>RECORD_STATUS_CONFLICT</code>	KVNR wurde zwar bei einem Aktensystembetreiber gefunden, scheint aber nicht aktiviert zu sein (ePA ist deaktiviert)	User: muss sich bei dem angemeldeten Aktensystembetreiber wieder freischalten lassen.