KiDz Webservice

Letzte Änderung: 24.01.2024

- Einleitung
  - Unterstützte Module
- Versionsinformationen
- Schnittstellendefinition
- Code-Generierung
- Authentifizierung und Autorisierung
  - Beispielcode JWT Generierung
  - Benutzer API-Token
  - Single Sign-On
- Testen der Anbindung
  - Beispielcode Schnittstellenaufruf

Einleitung

Bei dem KiDz Webservice handelt es sich um eine REST API, die Drittanbietern die Möglichkeit bietet, Daten zwischen Ihren Systemen und KiDz auszutauschen. Die API wurde nach der "OCTO Restful API Design Reference Card" designt, deren Best Practice Prinzipien als De-facto-Standard angesehen werden können.

Unterstützte Module

Derzeit werden folgende Module unterstützt:

- Kindmodul
- Personalmodul
- Betriebserlaubnis

Versionsinformationen

Die Versionierung des Webservices erfolgt nach dem Prinzip des Semantic Versioning. Abwärtsinkompatible Änderungen, d.h. eine Erhöhung der Major-Versionsnummer, werden möglichst vermieden und in jedem Fall rechtzeitig vorher angekündigt, da stets nur die aktuelle Version des Webservices nutzbar ist.

Version	Datum	Kommentar
4.0.0	24.01.2024	Anpassungen Jahresumbruch 23/24 • Kind: Feld strasseAusserhalb entfernt • Personal: Felder istVertretung, vertretungTyp entfernt; istAngestelltBeiDritten und angestelltBei hinzugefügt • PersonalTaetigkeitsbereich: Feld istUebergangsregelung entfernt
3.0.1	17.02.2023	Feld istUebergangsregelung an PersonalTaetigkeitsbereich ergaenzt
3.0.0	02.02.2023	Anpassungen Jahresumbruch 22/23 • Feld anerkennungsjahr bei PersonalQualifikaton entfernt • neue Felder hatKeineKitagFoerderung und hatUkrainischSprache bei PersonalMitarbeiter • neues Feld betreuungsdauerSchulkind bei KindBetreuungMerkmal • neue Felder hatEingliederungshilfe und istUkraineFluechtling bei Kind • Anpassungen Ausprägungen s. fachliche Hinweise zu Änderungen in KiDz, z.B. neue Tarifgruppen TV-L S
2.1.0	02.06.2022	Single Sign-On freigeschaltet
2.0.1	20.05.2022	Filter für Personalnummer an GET Personal hinzugefügt
2.0.0	14.03.2022	• Feld istBthgFinanziert bei KindBetreuungMerkmal entfernt • Wochentags-Angabe bei Vertretung PersonalTaetigkeitsbereich hinzugefügt • Feld geburtsdatum PersonalMitarbeiter ab sofort als Tagesangabe immer der erste Tag des Monats • Anpassungen Ausprägungen s. fachliche Hinweise zu Änderungen in KiDz, u.a. PersonalMitarbeiter (neue Tarifgruppe TV-L) bzw. PersonalTaetigkeitsbereich (neue Gründe Ausfallzeit/Unterbrechung)
1.3.2	18.02.2022	GET Betriebserlaubnis mitHistorie Filter entfernt, da bereits über stichtag-Filter abgedeckt
1.3.1	17.12.2021	Fix Tippfehler Property KindBetreuungMerkmal::istBtghFinanziert und PersonalTaetigkeitsbereich::istBthg (Btgh => Bthg)
1.3.0	17.12.2021	POST und PUT kind sowie personal liefern nun direkt die DTOs zurück
1.2.6	10.11.2021	Filter an GET betriebserlaubnisse hinzugefügt zur Abfrage historisierter Betriebserlaubnisse, Bugfixes, Doku
1.2.5	29.10.2021	Plausibilitäts-Hinweise an Kind- und Personal-DTO hinzugefügt, Bugfixes
1.2.0	11.08.2021	DELETE kinder und DELETE personal hinzugefügt, Bugfixes

Version	Datum	Kommentar
1.1.0	10.08.2021	GET betriebserlaubnisse hinzugefügt, Bugfixes
1.0.0	01.08.2021	Initiale Version

Schnittstellendefinition

Die Schnittstellendefinition ist nach der OpenAPI Spezifikation beschrieben und lässt sich über Anhängen von "/api/%version%/docs" (z.B. "/api/v4/docs") an die Adresse des KiDz Systems visualisiert abrufen. **Dort befindet sich auch die detaillierte Beschreibung der Endpunkte, Parameter und DTOs etc**. Die Spezifikationsdatei (swagger.json) lässt sich durch Anhängen von "/api/%version%/spec.json" (z.B. "/api/v4/spec.json") abrufen. Für beide Ansichten sind Zugangsdaten erforderlich. Die URLs und Zugangsdaten lauten:

Visuelle Dokumentation: https://kita.service24.rlp.de/api/v4/docs

Spezifikationsdatei: https://kita.service24.rlp.de/api/v4/spec.json

Bitte kontaktieren Sie den Support für die Zugangsdaten (Menüpunkt "Suppport" → "Supportformular"), wenn Sie die Schnittstelle implementieren möchten.

Mit "Swagger explained" kann man sich eine swagger.json-Datei interaktiv erläutern lassen.

Die Schnittstelle ist durch Anhängen von "/api" und der aktuellen Major-Versionsnummer an die KiDz URL abrufbar, z.B. für das Livesystem https://kita.service24.rlp.de/api/v4

Code-Generierung

Aus der Schnittstellendefinition kann mit swagger-codegen direkt der passende Client mit allen benötigten DTOs generiert werden.

Authentifizierung und Autorisierung

Die Authentifizierung erfolgt über **JSON Web Token (JWT) zum einen für das Drittanbietersystem** in Form einer eindeutigen Issuer-ID und einer über ein **Public-/Private-Key-Paar generierten Signatur**, **zum anderen für den KiDz Benutzer mit Hilfe eines API-Tokens**, den der Benutzer in den Benutzerdaten in KiDz generieren kann.

Dazu verfügt jedes angebundene Drittsystem über einen eigenen, bei Anbindung an das KiDz-System zu generierenden **Private Key, der auf dem Drittsystem verbleibt**. Der dazugehörige **Public Key wird vom Drittanbieter an die NPO Applications GmbH übermittelt**. Als **Key-Länge sind 4096 Bits** zu wählen.

Mit OpenSSL können die Keys z.B. so generiert werden:

```
openssl genrsa -out var/keys/rlp_kidz_private.pem -aes256 4096
openssl rsa -pubout -in var/keys/rlp_kidz_private.pem -out var/keys/rlp_kidz_public.pem
```

Bei jedem Aufruf einer Webserviceschnittstelle muss der Client einen neuen **JSON Web Token (JWT)** generieren und mitschicken. Dabei sind folgende Claims im JWT zu übermitteln:

- **iss (issuer)-Claim**: eindeutige ID, die dem Drittanbieter durch den Schnittstellenanbieter bei Registrierung mitgeschickt wird und die mit dem Public-Key verbunden ist
- **exp (expiration time)-Claim**: die TTL (Time to live) sollte so klein wie möglich gewählt werden. Empfohlen wird eine TTL von 30s, wobei der Wert je nach Infrastruktur angepasst werden kann (z.B. aufgrund von nicht korrigierbaren Serverzeitunterschieden).
- **jti (JWT ID)-Claim:** eindeutige Token ID ("jti"), die für jeden Aufruf neu generiert werden muss. Tokens ohne jti werden abgelehnt. Die ID des Tokens wird gegen eine Blacklist geprüft, sodass ein Token auch innerhalb der TTL-Laufzeit gegen Replay-Attacken gesichert ist. Für die Generierung der Token-ID wird UUID4 empfohlen.
- **id-Claim:** API-Token des KiDz Benutzers, für den der Aufruf erfolgt

Für die Signatur des JWTs muss RSA256 verwendet werden. Der JWT Token muss im Request dann entweder als "jwt"-URL-Parameter oder **Authorization-Header "Bearer %TOKEN%" mitgeschickt werden**, wobei letzteres empfohlen wird.

Beispielcode JWT Generierung

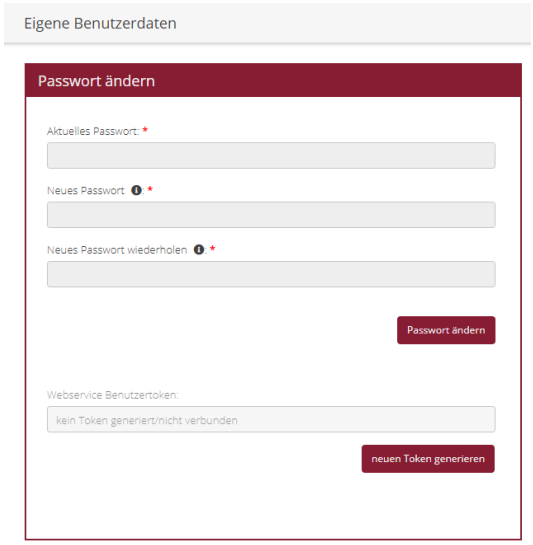
Folgendes PHP Code-Beispiel generiert bei Verwendung der lcobucci/jwt-Bibliothek (empfohlene Implementierung im PHP Client) einen passenden JWT für den Aufruf einer Schnittstellenfunktion:

```php
$privateKey = \Lcobucci\JWT\Signer\Key\InMemory::file('file://pfad/zum/private/key.pem', 'passwortEmpfohlen');
$ttl = 30;
$jwtConfig = \Lcobucci\JWT\Configuration::forSymmetricSigner(new Sha256(), $privateKey);
$builder = $jwtConfig->builder()
    ->issuedBy($systemId)
    ->withClaim('id', $benutzerApiKey);
    ->issuedAt(new \DateTimeImmutable('@' . (time())))
    ->expiresAt(new \DateTimeImmutable('@' . (time() + $ttl)))
    ->identifiedBy(\Ramsey\Uuid\Uuid::uuid4()->serialize());

$jwt = $builder
    ->getToken($jwtConfig->signer(), $jwtConfig->signingKey())
    ->toString();
```

Benutzer API-Token

Jeder KiDz Benutzer kann in den Benutzerdaten einen API-Token generieren:



Wird ein API-Token generiert, erscheint dieser in dem Eingabefeld und kann vom Benutzer an den Drittanbieter übermittelt bzw. in dessen Software eingegeben werden. Sobald der Token das erste Mal durch den Drittanbieter in einem Schnittstellenaufruf verwendet wird, **ist der API-Token an den Drittanbieter gebunden** und kann nicht mehr durch andere Drittanbieter verwendet werden. Im Eingabefeld oben erscheint dann "verbunden mit: %Name des Drittsystems%". Will der Benutzer ein anderes Drittanbietersystem nutzen, muss er einen neuen API-Token generieren. Da der bisher angebundene Anbieter dabei seine Berechtigung für den Benutzer verliert, erscheint in diesem Fall bei der Neugenerierung eine Sicherheitsabfrage für den Benutzer.

Single Sign-On

Der für einen Benutzer erzeugte JWT kann auch zum Single Sign-On in das KiDz-System verwendet weden. Dazu wird wie oben beschrieben ein JWT generiert und KiDz über Anhängen von /sso?jwt=%JWT% aufgerufen, z.B.

```
https://kita.service24.rlp.de/sso?jwt=%JWT%
```

Testen der Anbindung

Um Softwareherstellern während der Implementierungsphase und vor der Anbindung des Livesystems Funktionstests sowie eine Qualitätssicherung der konsumierenden Schnittstellen zu ermöglichen, wird ein **Testsystem bereitgestellt,** mit dem die Anbindung zunächst getestet werden kann. Dieses Vorgehen vermeidet Testeingaben im Echtsystem und stellt den störungsfreien Echtbetrieb von KiDz sicher. Für die Bereitstellung eines Testsystem-Zugangs wenden sie sich bitte an die **NPO Applications GmbH**.

Zusätzlich existiert ein **spezieller Endpunkt zum Testen der Schnittstellenanbindung, der auch im Livesystem genutzt werden kann**, da keine Veränderungen vorgenommen werden: GET /ping (siehe Schnittstellendefinition). Dieser Endpunkt ist der einzige, der ohne einen Benutzer API-Token aufgerufen werden kann und ermöglicht es so dem Drittanbieter, auch ohne angebundenen Benutzer die Verbindung zum KiDz Webservice zu überprüfen.

Beispielcode Schnittstellenaufruf

Die Schnittstelle akzeptiert und antwortet nur im JSON-Format. %JWT% ist der zuvor generierte JWT-Token (siehe Aufrufbeispiel oben).

```
curl -X GET -i --header "Content-Type: application/json" --header "Accept: application/json" --header "Authorizatio
```