REST-Zugriff: API-Keys und Zugriffsrechte

Externe Clients (Kiosk-Terminals, mobile Apps, Integrationen) sprechen Webdesk über die REST-Schnittstelle an (Pfad /rest/... unterhalb des Webdesk-Kontextpfads). Diese Seite beschreibt, wie sich ein solcher Client authentifiziert, wie ein API-Key angelegt und verwaltet wird und wie über **Zugriffsrechte** und **Zugriffsrecht-Sets** festgelegt wird, welche REST-Aufrufe er ausführen darf. Sie richtet sich an Systemadministratoren und Berater. Die Zugriffsrechte für REST-Endpunkte sind unabhängig von den Aktionsberechtigungen der Berechtigungssteuerung[1] (siehe Berechtigungstypen[2]): Diese steuern, wer eine Aktion in der Oberfläche aufrufen darf, die hier beschriebenen Zugriffsrechte steuern, welche REST-URLs mit welcher HTTP-Methode aufgerufen werden dürfen.

Zwei Wege der Authentifizierung

Merkmal	Anmeldung mit Benutzer (Session)	API-Key
Typischer Einsatz	Integrationen und Apps, die im Namen eines Webdesk-Benutzers arbeiten	Geräte und Systeme ohne Benutzeranmeldung, z. B. Kiosk-Terminals
Ablauf	POST /rest/auth/login mit JSON-Body username, password und optional rememberLogin. Antwort 200 mit Sitzungsdaten (u. a. authentication-state, session-timeout, login-token) bzw. 401 bei falscher Anmeldung. Die folgenden Aufrufe müssen das Sitzungs-Cookie mitsenden.	Jeder Aufruf trägt den HTTP-Header X-Auth-Token: <your-api-key>. Eine vorherige Anmeldung ist nicht nötig.
Berechtigung	Zugriffsrechte der **Rollen** des Benutzers (Rollen-Reiter "Zugriffsrechte")	Zugriffsrechte des **API-Keys** (direkt am Key und über zugeordnete Zugriffsrecht-Sets)

Ist der Header X-Auth-Token gesetzt, wird der Aufruf als API-Key-Aufruf behandelt, andernfalls als Session-Aufruf. (Nur im Betriebsmodus dev oder test wird zusätzlich Basic Authentication über den Header Authorization akzeptiert; sie ist für Tests gedacht.)

Der Aufruf /rest/auth/login sowie einige weitere öffentliche Endpunkte (z. B. Passwort-Rücksetzung, Systemzeit) werden von der REST-Authentifizierung nicht geprüft. Für Browser-Anmeldungen ist der normale Weg weiterhin die Webdesk-Login-Seite; /rest/auth/login ist für externe Clients vorgesehen.

Session-Aufrufe: Berechtigung über Rollen

Externe Clients mit Benutzeranmeldung benötigen rollenbasierte Zugriffsrechte. Es zählen alle Zugriffsrechte und Zugriffsrecht-Sets aller Rollen, die der angemeldete Benutzer innehat. Ist für URL und HTTP-Methode weder ein erlaubendes noch ein verbietendes Recht vorhanden, wird der Aufruf mit 403 abgewiesen. Einige Endpunkte, die für jeden angemeldeten Benutzer gedacht sind (z. B. unter /rest/po/systeminfo/, /rest/po/users/, /rest/po/actionpermissions/), sind von der Rechteprüfung ausgenommen; für sie genügt die Anmeldung. Ruft die Oberfläche selbst REST-Endpunkte auf (Erkennung über den Referer), gelten stattdessen die in der jeweiligen Aktion hinterlegten Zugriffsrechte (siehe unten).

API-Key mit Benutzer-Identifikation (Impersonation)

Ein API-Key kann zusätzlich im Namen eines Webdesk-Benutzers handeln, z. B. wenn ein Kiosk-Terminal nach dem Scannen einer NFC-Karte Daten dieser Person abruft. Dazu sendet der Client **einen** der beiden Header:

Header	Erklärung
User-Identification-NfcCard	ID der NFC-Karte; der Benutzer ist die Person, der die Karte zugewiesen ist.
User-Identification-Useruid	UID der Person.

Einschränkungen:

- Der Key muss zuerst für den aufgerufenen Endpunkt berechtigt sein (gleiche Prüfung wie ohne Header).
- Es dürfen nicht beide Header gleichzeitig gesendet werden und der Wert darf nicht leer sein (Antwort 401).
- Ist keine Person zur Karte bzw. UID zu finden, lautet die Antwort 401.
- Impersonation ist nur für **Kiosk-Keys** erlaubt, also Keys, die einem Kiosk zugeordnet sind, und nur für Personen **desselben Mandanten** wie der Kiosk. Bei einem Key ohne Kiosk oder einer Person eines anderen Mandanten lautet die Antwort 403.

Die Zugriffsrechte des Keys bleiben auch bei Impersonation maßgeblich; die Rechte der identifizierten Person erweitern sie nicht.

API-Keys verwalten

Die Masken liegen im Admin-Menü unter "System" (im Ordner "Sicherheit"): **"API Keys"** (Liste) und **"Zugriffsrecht Sammlung"** (Zugriffsrecht-Sets). Je nach Menüaufbau des Systems kann der Eintrag "Zugriffsrecht Sammlung" auch direkt unter "System" stehen.

Liste "API Keys"

Die Liste zeigt alle Keys mit den Spalten "API Key" (der Schlüsselwert selbst), "API Consumer", "Gültig von" und "Gültig bis"; auch ungültig gemachte Keys bleiben sichtbar. Über das Symbol in der ersten Spalte wird ein Key geöffnet, über die Schaltfläche **"Generate New API Key"** ein neuer angelegt.

Neuen Key anlegen

1. In der Liste "API Keys" auf **"Generate New API Key"** klicken.
2. Im Feld **"API Consumer"** den Namen des Clients eintragen (nur zur Beschreibung, z. B. "Terminal Eingang Wien").
3. Im Abschnitt "Api Key Zugriffsrecht-Sets" über "Hinzufügen" ein oder mehrere Zugriffsrecht-Sets zuordnen und/oder im Abschnitt "Api Key-Spezifische Zugriffsrechte" einzelne Rechte erfassen (siehe unten).
4. Speichern. Es erscheint "API Schlüssel erfolgreich gespeichert"; im Feld "API Key" steht nun der erzeugte Schlüsselwert, der im Client einzutragen ist.

Achtung: Ein neuer Key ohne Zugriffsrecht-Set und ohne eigene Rechte ist zwar gültig, darf aber keinen einzigen Endpunkt aufrufen (Antwort 403).

Achtung: Der Schlüsselwert ist das Geheimnis, mit dem sich der Client ausweist, und wird in der Liste und in der Maske im Klartext angezeigt. Der Zugriff auf diese Masken sollte entsprechend eingeschränkt sein; Keys gehören in keine Dokumentation, kein Ticket und keine URL. Webdesk bietet für Bilder (/image/...) zwar zusätzlich den Parameter apiKey an, der Header X-Auth-Token ist aber vorzuziehen.

Key bearbeiten, ungültig machen, löschen

- **"API Consumer"** und die Zugriffsrechte können jederzeit geändert werden. "Gültig von" und "Gültig bis" sind in der Maske nur Anzeige. Beim Speichern werden die Sets und Rechte des Keys durch den Inhalt der Maske ersetzt. Bereits gespeicherte Rechtezeilen sind nur lesbar und können nur gelöscht werden; geänderte Rechte werden als gelöschte und neu erfasste Zeile umgesetzt. Änderungen an den Rechten gelten für die folgenden Aufrufe; eine Neuanmeldung des Clients ist nicht nötig.
- **"ungültig machen"** (Rückfrage: "Sind Sie sicher, dass Sie diesen API-Schlüssel ungültig machen wollen?"): Der Key wird beendet (Gültig bis = Vortag) und liefert sofort 401. Der Eintrag bleibt in der Liste erhalten; die Schaltfläche verschwindet. Ein ungültiger Key kann in der Oberfläche nicht wieder gültig gemacht werden.
- **"Löschen"** (Rückfrage: "Sind Sie sicher, dass Sie diesen API-Schlüssel löschen wollen?", Bestätigung "API-Schlüssel löschen"): Der Key wird endgültig entfernt.

Gehört der (gültige) Key zu einem Kiosk, weist die Rückfrage darauf hin, dass dieser Kiosk danach nicht mehr funktioniert. Das Ungültigmachen oder Löschen löst die Verbindung zwischen Kiosk und Key; einen neuen Key erzeugt man in der Kiosk-Maske (siehe unten).

Keys von Kiosks

Für Kiosks (Menü "Karten u. Kiosk" > "Kiosks") verwaltet Webdesk den Key selbst:

- Beim Anlegen oder Registrieren eines Kiosks wird automatisch ein Key erzeugt (API Consumer = Kioskname) und diesem das **Standard-Zugriffsrecht-Set** des Kiosk-Typs zugeordnet (siehe unten).
- In der Kiosk-Maske zeigt das Feld "API key" mit der Schaltfläche "API-Schlüssel anzeigen" den Schlüssel an. Die Schaltfläche steht nur Benutzern zur Verfügung, die alle Mandanten einsehen dürfen (Systemadministration).
- Hat der Kiosk keinen gültigen Key, erscheint die Schaltfläche **"API-Schlüssel generieren"**. Sie erzeugt einen neuen Key mit dem Standard-Set und entfernt den alten Key.
- Wird ein Kiosk gelöscht, wird sein Key ungültig gemacht bzw. mitgelöscht.
- Wird ein bestehender PTM-Kiosk zusätzlich als TA-Kiosk registriert (oder umgekehrt), wird er zum Typ PTM_TA und erhält ebenfalls einen neuen Key.

Zugriffsrechte und Zugriffsrecht-Sets

Aufbau eines Zugriffsrechts

Eigenschaft	Erklärung	mögliche Wert(e)
"Selektor"	Pfad des REST-Endpunkts, ohne Kontextpfad und ohne Query-Parameter, beginnend mit /rest/..., z. B. /rest/po/kiosks. Platzhalter: * = beliebige Zeichenfolge (auch über / hinweg), ? = genau ein beliebiges Zeichen. Der Selektor muss auf den gesamten Pfad passen.	Text, Pflichtfeld
"Privileg"	Erlaubte Art des Zugriffs. Zuordnung zur HTTP-Methode siehe unten.	READ, CREATE, UPDATE, DELETE, FULL
"Ablehnen"	Bei gesetztem Haken verbietet das Recht den Zugriff, statt ihn zu erlauben.	Ja/Nein (Standard: Nein, das Recht erlaubt)

HTTP-Methode	benötigtes Privileg
GET	READ
POST	CREATE
PUT, PATCH	UPDATE
DELETE	DELETE

FULL entspricht READ, CREATE, UPDATE und DELETE zusammen. Andere HTTP-Methoden (z. B. HEAD) können nicht zugeordnet werden und führen zu einer Fehlerantwort 401.

Wo Zugriffsrechte vergeben werden

Ort	Wirkung
API-Key (Abschnitte "Api Key Zugriffsrecht-Sets" und "Api Key-Spezifische Zugriffsrechte")	Für Aufrufe mit X-Auth-Token. Es gilt die Summe aus den eigenen Rechten des Keys und den Rechten aller zugeordneten Sets.
Rolle (Reiter "Zugriffsrechte" in der Rollenmaske, Abschnitte "Rollenspezifische Zugriffsrecht-Sets" und "Rollenspezifische Zugriffsrechte"; siehe Rollen[3])	Für Session-Aufrufe aller Rolleninhaber. Der Reiter ist nur für Systemadministratoren und nicht bei abgeleiteten Mandanten-Rollen sichtbar.
Aktion (Datei act-descr.xml)	Für REST-Aufrufe, die die Oberfläche aus dieser Aktion heraus tätigt. Nicht über die Oberfläche änderbar: Die Aktion zeigt sie im Reiter "Zugriffsrechte" nur an, Änderungen dort werden nicht gespeichert.
Zugriffsrecht-Set	Wiederverwendbare Sammlung von Rechten, die beliebig vielen Keys und Rollen zugeordnet werden kann.

In act-descr.xml werden Rechte als <accessrights><accessright selector="/rest/po/audit/label/*" privilege="GET"/></accessrights> deklariert. Das Attribut privilege nimmt eine oder mehrere (durch Komma getrennte) HTTP-Methoden entgegen, z. B. GET,POST. Solche Rechte erlauben immer; ein Ablehnen ist nicht möglich.

Zugriffsrecht-Sets pflegen

1. Admin-Menü "System" > **"Zugriffsrecht Sammlung"** öffnen (Liste "Accessright Sets" mit der Spalte "Name").
2. Auf **"Neues Zugriffsrecht Set"** klicken bzw. ein bestehendes Set öffnen.
3. Einen eindeutigen **"Name"** vergeben (Pflichtfeld; der Name darf nur einmal vorkommen).
4. Über "Hinzufügen" Zeilen mit "Selektor", "Privileg" und gegebenenfalls "Ablehnen" erfassen.
5. Speichern. Bestehende Zeilen sind nur lesbar und können nur gelöscht werden; für eine Änderung die Zeile löschen und neu erfassen. Das Set kann anschließend im API-Key bzw. in der Rolle zugeordnet werden (Auswahlfeld "Zugriffsrecht-Set"; ein Sprung-Symbol öffnet das Set).

Achtung: Änderungen an einem Set wirken auf alle Keys und Rollen, denen es zugeordnet ist.

Standard-Sets für Kiosks

Für Kiosk-Keys legt Webdesk beim ersten Bedarf ein Set mit dem Namen Default <Typ> Kiosk Access Right Set an (Typ = PTM, TA oder PTM_TA) und ordnet es dem neuen Key zu. Die Sets enthalten (alle erlaubend):

Selektor	Privileg	in Set
/rest/po/kiosks*	FULL	PTM, TA
/rest/po/nfctags*	FULL	PTM, TA
/rest/ptm/*	FULL	PTM, TA
/image/*	READ	PTM, TA
/rest/ta/*	FULL	nur TA

"Default PTM_TA Kiosk Access Right Set" vereinigt die Rechte beider Sets. Ein einmal angelegtes Set wird über seinen Namen wiedergefunden; spätere Änderungen an der Standardkonfiguration verändern ein bestehendes Set nicht. Man kann ein Set jedoch in der "Zugriffsrecht Sammlung" bearbeiten, dann ändern sich die Rechte aller Kiosks dieses Typs.

Wie die Berechtigung eines Aufrufs entschieden wird

Für einen API-Key-Aufruf werden die Rechte des Keys und aller seiner Sets zusammengeführt und gegen Pfad und HTTP-Methode des Aufrufs geprüft:

1. Der Pfad des Aufrufs (ohne Kontextpfad, z. B. /rest/po/kiosks/4711) wird mit den Selektoren verglichen. Methode und benötigtes Privileg ergeben sich aus der obigen Tabelle.
2. Ein Selektor **ohne Platzhalter**, der genau dem Pfad entspricht und für das benötigte Privileg ein Recht enthält, entscheidet sofort (erlaubt oder abgelehnt).
3. Sonst entscheidet unter den passenden Selektoren **mit Platzhalter** der spezifischste, das ist der mit den meisten Zeichen, die keine Platzhalter sind. Selektoren ohne Recht für das benötigte Privileg zählen dabei nicht. Bei gleich spezifischen Selektoren ist die Reihenfolge nicht festgelegt; solche Überschneidungen sollten vermieden werden.
4. Innerhalb desselben Selektors gewinnt **"Ablehnen"** immer gegen "erlauben", unabhängig von der Reihenfolge.
5. FULL wirkt wie READ, CREATE, UPDATE und DELETE. Erlaubt ein Selektor UPDATE oder CREATE, ist READ für diesen Selektor mit erlaubt (sofern READ dort nicht abgelehnt ist).
6. Trifft kein Recht zu, ist der Aufruf **nicht berechtigt** (403). Es gibt keinen Standard-Zugriff.

Für Session-Aufrufe werden dieselben Regeln auf die Rechte der Rollen des Benutzers angewendet. Auch hier gilt: kein passendes Recht bedeutet 403.

Beispiel: Schlüssel für ein Terminal, das nur lesen darf

Ein externes Terminal soll Personendaten abrufen, aber nichts verändern. (Die Pfade sind Beispiele; die tatsächlichen Pfade eines Endpunkts zeigt die Swagger UI.)

1. "System" > "Zugriffsrecht Sammlung" > "Neues Zugriffsrecht Set": "Name" = Terminal nur lesen.
2. Zeile hinzufügen: "Selektor" = /rest/po/persons/*, "Privileg" = READ, "Ablehnen" nicht gesetzt. Speichern.
3. "System" > "API Keys" > "Generate New API Key": "API Consumer" = Terminal Eingang Wien; im Abschnitt "Api Key Zugriffsrecht-Sets" das Set "Terminal nur lesen" hinzufügen; speichern.
4. Den angezeigten Schlüssel im Terminal als Header X-Auth-Token eintragen.

Ergebnis: GET /rest/po/persons/... wird beantwortet. POST, PUT oder DELETE auf denselben Pfad sowie jeder Aufruf eines anderen Pfads liefern 403. Wird später der Zugriff entzogen, genügt es, das Set zu ändern oder den Key über "ungültig machen" zu sperren (401).

Hinweis: Der Terminal-Key aus diesem Beispiel ist keinem Kiosk zugeordnet und kann daher keine Benutzer-Identifikation (User-Identification-...) verwenden. Keys für Kiosks entstehen über die Kiosk-Verwaltung.

Fehlersuche: 401 und 403

Antwort	mögliche Ursachen und Prüfung
401 (nicht authentifiziert)	• Header X-Auth-Token fehlt oder ist leer und es gibt keine Session. • Der Key ist unbekannt (Tippfehler, gelöscht) oder nicht mehr gültig: in "API Keys" "Gültig von"/"Gültig bis" prüfen. • Impersonation: beide User-Identification-...-Header gleichzeitig, leerer Wert oder keine Person zur Karte bzw. UID gefunden. • Nicht unterstützte HTTP-Methode (z. B. HEAD) oder ein unerwarteter Fehler; dann steht im Server-Log "Problem during authentication/ authorization.". • Session-Aufruf: Sitzung abgelaufen oder nie angemeldet (/rest/auth/login liefert bei falscher Anmeldung 401).
403 (nicht berechtigt)	• Der Key ist gültig, aber kein Recht passt auf Pfad und Methode, oder ein "Ablehnen"-Recht greift. Im API-Key die Sets und Rechte prüfen: stimmt der Selektor (beginnt mit /rest/, ohne Kontextpfad)? Passt das Privileg zur Methode (GET = READ, POST = CREATE)? Endet der Pfad mit einem Schrägstrich, den der Selektor nicht abdeckt? • Impersonation mit einem Key, der keinem Kiosk zugeordnet ist, oder mit einer Person eines anderen Mandanten. • Session-Aufruf: die Rollen des Benutzers haben kein passendes Recht (Rollenmaske, Reiter "Zugriffsrechte").

Zusätzlich protokolliert der Dienst auf Debug-Ebene für jede Prüfung eine Zeile der Art "Check if api key ... has permission to make a GET call over /rest/...". Sie nennt Methode und Pfad des Aufrufs (Logger at.workflow.webdesk.po.impl.PoAccessRightServiceImpl).

Browser-Clients: CORS

Für Aufrufe aus Browsern, die von einer anderen Adresse stammen, setzt Webdesk die CORS-Header. Die erlaubten Ursprünge stehen im Systemparameter allowedOriginsForCORS der Bean SecurityHeadersOptions (Maske "Systemparameter"). Eine Anfrage mit passendem Origin erhält Access-Control-Allow-Origin mit diesem Wert und Access-Control-Allow-Credentials: true. Einträge dürfen * enthalten, z. B. https://app.example.com oder https://*.example.com. Erlaubte Request-Header sind u. a. X-Auth-Token, User-Identification-NfcCard und User-Identification-Useruid. Native Apps und Terminals senden keinen Origin-Header und sind davon nicht betroffen.

Achtung: Im ausgelieferten Standard ist die Liste leer. In diesem Fall wird jeder gesendete Origin unverändert zurückgegeben, d. h. es wird jeder Ursprung erlaubt. Für Systeme, die aus dem Internet erreichbar sind, sollten die erlaubten Ursprünge eingetragen werden.

Sicherheitshinweise

- **CSRF** ist in Webdesk derzeit nicht implementiert. Der Schutz über die sitzungsgebundene Continuation-ID der Formulare gilt nicht für REST-Aufrufe und login_process. Das ist bei der Bewertung von Sicherheitsthemen und beim Hinzufügen neuer POST-Endpunkte zu beachten.
- Kurzlebige Anmelde-Tokens (LoginTokenUtils, z. B. für serverseitige Report-Downloads mit Benutzerwechsel) und der temporäre Zugriff über security-key liegen im Cache des jeweiligen Knotens. Im Cluster können Aufrufe, die auf einem anderen Knoten landen, deshalb scheitern, wenn die Requests nicht auf einen Knoten fixiert werden.
- Session- und Cookie-Einstellungen (z. B. SameSite, Header, Timeouts) sind umgebungsabhängig und müssen vor Ort geprüft werden.

Swagger UI und Postman-Collection

- **Swagger UI:** Der Menüeintrag **"Swagger Rest-API Explorer"** (Admin-Menü "System") öffnet in einem neuen Fenster rest/swagger-ui/index.html mit den REST-Endpunkten von Webdesk, ihren Pfaden, Methoden und Parametern. Aufrufe aus der Swagger UI laufen mit der Sitzung des angemeldeten Benutzers. Zulässig sind sie für Systemadministratoren und für Benutzer, die über irgendeine ihrer Aktionen Zugriffsrechte auf den Endpunkt haben. Die Swagger UI eignet sich, um Pfade für Selektoren nachzuschlagen.
- **Postman-Collection:** Im Repository liegt postman/Webdesk - Public API.postman_collection.json. Sie enthält den Aufruf "Login" (POST {{restURL}}/auth/login) sowie Beispielaufrufe zur Zeitwirtschaft (Abwesenheitsgründe, eigene Buchungen, aktueller Status, Buchung auslösen). Die Variable restURL ist in der Collection nicht vorbelegt und muss auf die REST-Basisadresse des Systems (Webdesk-Adresse mit Zusatz /rest) gesetzt werden. Die Collection verwendet die Anmeldung per Session, keine API-Keys.

1. /daisy/webdesk-manual/admin/po-overview/5479-dsy.html
2. /daisy/webdesk-manual/admin/po-overview/5479-dsy/2650-dsy.html
3. /daisy/webdesk-manual/admin/po-overview/3065-dsy/2640-dsy/3320-dsy.html