UIC-Sicherheitsportal

Dokumentation

Version: 2.0

Erstellt durch:

Deutschlandtarifverbund-GmbH
Speicherstraße 59
60327 Frankfurt am Main

Frankfurt, den 07.07.2026

Versionsverwaltung

Version	Datum	Bemerkung
Version 1.5	01.04.2026	Komplette Überarbeitung der Beschreibung der API-Dokumentation
Version 1.6	22.05.2026	Anpassung Struktur D-Tarif-Barcode Endpunkt Ergänzung Hinweis zur fehlenden RICS-Codes in Kontrolldaten
Version 1.7	16.06.2026	Entfernung obsoleter Beispielaufrufe der API-Endpunkte Ergänzung KVV-Tarif Endpunkt Ergänzung Kapitel zum Datenschutz Ergänzung Information zur Weiterentwicklung
Version 1.8	01.07.2026	Ergänzung normalisierte Darstellung der Datenstrukturen Erweiterung Tabelle 2 um benötigte Rollen Ergänzung NOX-Endpunkt Erweiterung Sperrlistenabruf um Product Owner Auswahl
Version 2.0	07.07.2026	Ergänzung Endpunkte NDS-Tarif und SH-Tarif

Inhaltsverzeichnis

1 Glossar

In diesem Glossar werden grundlegende Begriffe im Kontext des UIC-Sicherheitsportals erklärt.

Begriff	Erläuterung
API-Key (API-Schlüssel)	Zugriffsschlüssel zur Authentifizierung gegenüber der API. Wird im Header einer Anfrage mitgeführt.
Access-Token	Zeitlich begrenztes Zugriffstoken zur Authentifizierung gegenüber der API.
Deutschlandweite Haltestellen-ID (dhid)	Die deutschlandweite Haltestellen-ID (dhid) ist ein standardisierter Identifikator zur eindeutigen Referenzierung von Haltestellen und Bahnhöfen im öffentlichen Personenverkehr in Deutschland. Sie wird im Rahmen der Kontrolle verwendet, um den Ort einer Ticketprüfung eindeutig zu beschreiben. Die dhid basiert auf dem internationalen Standard ifopt und ist der branchenweit abgestimmte Datenstandards. Sie ermöglicht eine systemübergreifende Identifikation von Haltestellen. Beispiel: „de:01056:17016“
EntityTag	Kennung zur Versionierung von Daten. Ermöglicht es festzustellen, ob sich Inhalte seit der letzten Abfrage geändert haben.
Endpunkt (API-Endpunkt)	Ein API-Endpunkt ist eine definierte Schnittstelle innerhalb der API, über die eine konkrete Funktion des Systems aufgerufen werden kann. Ein Endpunkt wird durch eine Kombination aus HTTP-Methode und URL-Pfad beschrieben (z. B. POST /api/v1/validation/validate).
Fahrtberechtigung	Eine Fahrtberechtigung im öffentlichen Personenverkehr ist die rechtliche und organisatorische Grundlage, die einer Person das Recht einräumt, ein bestimmtes Verkehrsmittel oder eine festgelegte Strecke zu nutzen. Sie wird in der Regel durch den Besitz eines gültigen ***Tickets***, Abonnements oder einer entsprechenden Erlaubnis (z. B. Freifahrten, Dienstfahrscheine) nachgewiesen.
Flexible Content Barcode (FCB)	Siehe UIC-Barcode (FCB)
JWT (JSON Web Token)	Signiertes Tokenformat zur sicheren Übertragung von Authentifizierungsinformationen.
Kontrolle	Die Kontrolle bezeichnet die Prüfung eines Tickets im operativen Betrieb. Ziel ist die Bewertung, ob eine gültige Fahrtberechtigung vorliegt. Die Kontrolle kann online oder offline erfolgen und führt zu einem Kontrollergebnis.

Begriff	Erläuterung
Kontrollergebnis	Ergebnis einer Ticketprüfung. Enthält die Bewertung der Gültigkeit sowie zusätzliche Hinweise zur Nutzung eines Tickets.
Kontrollereignis	Ein Kontrollereignis ist eine ergänzende Information zum Kontrollergebnis, der bei einer durchgeführten Ticketprüfung übermittelt wird. Es dient der Erkennung ungewöhnlicher Nutzungsmuster.
KOSE	Sperrlistenservice der VDV eTicket Service GmbH, an den das UIC-Sicherheitsportal über einen Adapter gesperrte Deutschlandtickets übermittelt.
Kundenvertragspartner, KVP	Ein Kundenvertragspartner (KVP) ist eine natürliche oder juristische Person, die als primäre Ansprech- und Vertragsperson für Kunden im öffentlichen Personenverkehr fungiert. Der KVP ist verantwortlich für das Aushandeln, Abschließen und Verwalten von Kundenverträgen und sorgt dafür, dass Kunden eine verbindliche Vereinbarung für die Nutzung von Verkehrsmitteln wie Bussen, Bahnen oder Fähren haben. KVPs bieten umfassenden Support und Dienstleistungen, um sicherzustellen, dass die Kundenbedürfnisse erfüllt und die rechtlichen Rahmenbedingungen eingehalten werden. Beispiele für Kundenvertragspartner sind: ■ Verkehrsunternehmen, die direkte Dienstleistungen und Kundenverträge anbieten. ■ Verkehrsverbünde, die regional unterschiedliche Verkehrsangebote bündeln und Kunden vertraglich binden. ■ Drittanbieter-Plattformen, die als Vermittler zwischen Kunden und Verkehrsanbietern agieren. ■ Lokale Verkaufsagenturen, die spezifisch mit Kunden Verträge eingehen und als Bindeglied zu den Verkehrsdiensten fungieren. ■ Kundendienstzentren, die als zentrale Anlaufstelle für vertragliche Anliegen der Fahrgäste agieren.
Organisationscode	Eindeutige Kennung einer Organisation im UIC-Sicherheitsportal. Der Begriff umfasst sowohl den bisherigen RICS-Code als auch die aktuellen Organisationscodes der ERA. Für den Organisationscode gelten folgende Einschränkungen: Der Wert muss alphanumerisch sein und eine Länge von 4 Stellen haben.

Begriff	Erläuterung
Product Owner	Kennzeichnet die fachlich verantwortliche Organisation eines Tarifprodukts und wird im Barcode als numerischer Organisationscode (z. B. RICS-/ERA-Code) angegeben, um eine eindeutige Zuordnung des Tickets zum Tarifkontext unabhängig vom ausstellenden Unternehmen zu ermöglichen.
Rate Limiting	Mechanismus zur Begrenzung der Anzahl von API-Anfragen innerhalb eines bestimmten Zeitraums.
Refresh-Token	Token zur Erneuerung eines Access-Tokens ohne erneute Anmeldung.
Request	Ein Request ist eine Anfrage eines angebundenen Systems an die API. Er enthält alle notwendigen Informationen zur Ausführung einer Aktion, insbesondere Parameter im Header sowie Daten im Request Body.
Response	Eine Response ist die Antwort des Systems auf einen Request. Sie enthält das Ergebnis der Verarbeitung sowie gegebenenfalls zusätzliche Informationen zur weiteren Verwendung im aufrufenden System.
RICS	Kennung von Organisationen im UIC-Kontext. Wird im System aktuell als Identifikator verwendet, perspektivisch jedoch durch ERA-Organisationscodes ersetzt.
Schema	Ein Schema beschreibt die Struktur von Daten, die zwischen Systemen ausgetauscht werden. Es definiert Felder, Datentypen und deren Bedeutung im jeweiligen fachlichen Kontext.
Security Provider	Ein Security Provider ist technisch verantwortlich für die Barcodeerzeugung und Signierung von ***Tickets***. Der Security Provider garantiert für die Echtheit von Tickets – in diesem Falle für die Echtheit der generierten UIC-Barcodes.
Ticket	Ein Ticket im öffentlichen Personenverkehr ist eine Berechtigung, die einem Fahrgast den Zugang zur Nutzung von Verkehrsmitteln wie Bussen, Bahnen, Zügen oder Fähren ermöglicht. Es dient als Nachweis für den Erwerb eines Beförderungsrechts und kann unterschiedliche Formen haben, darunter physische Papierfahrkarten, elektronische Tickets (e-Tickets) oder digitale Tickets, die auf mobilen Endgeräten gespeichert sind. Im Kontext des UIC-Sicherheitsportals wird ein Ticket in Form eines UIC-Barcodes (FCB) dargestellt und verarbeitet.

Begriff	Erläuterung
Ticketaussteller (Ausgeber/Issuer)	Ein Ticketaussteller (Ausgeber bzw. Issuer) ist eine natürliche oder juristische Person, die ***Tickets*** im öffentlichen Personenverkehr erstellt, verkauft oder zur Verfügung stellt. Ticketaussteller agieren als Dienstleister, um Fahrgästen eine rechtsgültige ***Fahrtberechtigung*** für die Nutzung von Verkehrsmitteln wie Bussen, Bahnen oder Fähren anzubieten. Beispiele für Ticketaussteller sind: ■ Verkehrsunternehmen ■ Verkehrsverbünde ■ Drittanbieter und Ticketplattformen ■ Lokale Verkaufsstellen und Agenturen ■ Ticketautomaten und Selbstbedienungsgeräte ■ Internationale Verkehrsunternehmen Im UIC-Sicherheitsportal ist der Ticketaussteller über den Organisationscode eindeutig identifiziert.
TicketId (Issuer PNR)	Eindeutige Ticketkennung, die durch den Ticketaussteller vergeben wird und zusammen mit Organisationscode und Gültigkeitsende zur Identifikation eines Tickets dient.
Ticketkontrolleur (Ticketprüfer, TCO)	Ein Ticketkontrolleur (Ticketprüfer, TCO) ist eine natürliche oder juristische Person, die im öffentlichen Personenverkehr dafür verantwortlich ist, die Gültigkeit und Echtheit von ***Fahrtberechtigungen*** (z. B. ***Tickets***, Abonnements) der Fahrgäste zu überprüfen. Diese Tätigkeit dient der Sicherstellung, dass die Beförderungsbedingungen eingehalten werden und Fahrgäste für die Nutzung der Verkehrsmittel korrekt bezahlt haben. Beispiele für Ticketkontrolleure sind: ■ Mitarbeiter von Verkehrsunternehmen ■ Mitarbeiter von Verkehrsverbünden ■ Beauftragte Sicherheitsdienstleister ■ Automatisierte Kontrollsysteme ■ Fahrer oder Fahrzeugführer
Ticket-Lebenszyklus	Der Ticket-Lebenszyklus beschreibt die verschiedenen Zustände und Prozesse eines Tickets im System. Dazu gehören insbesondere die Ausstellung, die Nutzung (Kontrolle), mögliche Statusänderungen wie Sperrung oder Stornierung sowie die Verarbeitung von Kontroll- und Ausgabedaten.

Begriff	Erläuterung
Ticketstatus (fachlich)	Der fachliche Ticketstatus beschreibt die Bewertung eines Tickets im Rahmen der Kontrolle. Er gibt an, ob ein Ticket zum Zeitpunkt der Prüfung als gültig oder ungültig einzustufen ist. Die Bewertung erfolgt auf Basis verschiedener Kriterien, insbesondere des Gültigkeitszeitraums, des technischen Ticketstatus sowie weiterer Prüfungen (z. B. Sperrlistenabgleich). Der fachliche Ticketstatus bildet die Entscheidungsgrundlage im Kontrollprozess.
Ticketstatus (technisch)	Der technische Ticketstatus beschreibt den internen Zustand eines Tickets im System. Er gibt an, welche Statusinformationen zu einem Ticket gespeichert sind, beispielsweise ob ein Ticket aktiv, gesperrt oder storniert ist. Der technische Ticketstatus ist eine Eingangsgröße für die fachliche Bewertung eines Tickets, entspricht dieser jedoch nicht direkt.
UIC	Die UIC (Union Internationale des Chemins de fer) ist der internationale Verband der Eisenbahnunternehmen und definiert unter anderem Standards für den Datenaustausch und die Gestaltung von Tickets im Bahnverkehr.
UIC-Barcode (FCB)	Maschinenlesbare Darstellung eines Tickets gemäß UIC-Standard 918-9 (Flexible Content Barcode). Er enthält alle relevanten Ticketinformationen in codierter Form.
UIC IRS 918-9	Die UIC IRS 918-9 ist ein internationaler Standard zur Definition von Barcode-basierten Tickets im Eisenbahnverkehr. Er beschreibt beispielsweise den Aufbau und die Inhalte des Flexible Content Barcode (FCB).
validityFlags	Zusätzliche Hinweise im Kontrollergebnis, die auf besondere Nutzungsmuster oder Auffälligkeiten eines Tickets hinweisen. Sie ergänzen die Gültigkeitsbewertung, ohne diese direkt zu verändern. Im Kapitel 7.3.6 findet sich eine umfangreiche Erklärung dazu.

2 Einleitung

Die API des UIC-Sicherheitsportals bildet die technische Grundlage für zentrale Prozesse im digitalen Ticketing, insbesondere für die Ausstellung, Validierung und Kontrolle von Tickets mit UIC 918-9-Barcode im Deutschlandtarif, für Deutschlandtickets und darüber hinaus.

Diese Dokumentation beschreibt das Portal sowie seine Schnittstellen, Datenstrukturen und Endpunkte. Ziel ist es, die Inhalte aus fachlicher und technischer Sicht zu erläutern und in den Gesamtzusammenhang des Ticket-Lebenszyklus einzuordnen.

Dabei werden insbesondere folgende Fragen beantwortet:

- Wozu und wie nutze ich das UIC-Sicherheitsportal?
- In welchem fachlichen Kontext werden die verschiedenen Endpunkte verwendet?
- Welche Bedeutung haben einzelne Felder für Ausgabe, Kontrolle und Verarbeitung?
- Welche fachlichen Entscheidungen werden durch die gelieferten Informationen unterstützt?

Die Dokumentation richtet sich an:

- Fachbereiche im Vertrieb und in der Kontrolle
- Produktverantwortliche und Systemarchitekten
- sowie technische Integratoren mit fachlicher Verantwortung.

Zur besseren Orientierung ist das Dokument in mehreren aufeinander aufbauenden Kapiteln gegliedert, die unterschiedliche Perspektiven auf das System einnehmen:

- Die Kapitel 3 und 4 beschreiben die grundlegenden fachlichen Konzepte und die Nutzung des Systems, insbesondere die beteiligten Organisationen, Identifikationslogiken sowie die Struktur und Bedeutung von Tickets im UIC-Kontext.
- Kapitel 5 und 6 erläutern die technischen Grundlagen für den Zugriff auf die API, insbesondere die Authentifizierungsverfahren und sicherheitsrelevanten Rahmenbedingungen.
- Kapitel 7 beschreibt die verwendeten Datenstrukturen in Form von Anfrage- und Antwortschemata. Dabei werden die Inhalte nicht nur technisch dargestellt, sondern auch fachlich eingeordnet und anhand von Use Cases erläutert.
- Kapitel 8 beschreibt die konkreten API-Endpunkte. Hier wird dargestellt, wie die zuvor eingeführten Datenstrukturen und Konzepte in der praktischen Nutzung angewendet werden. Der Fokus liegt dabei auf der fachlichen Einordnung der einzelnen Endpunkte sowie deren Verwendung im operativen Betrieb.
- Kapitel 9 behandelt die Rückmeldungen des Systems, insbesondere technische Fehler sowie deren Interpretation und Behandlung im Kontext der Integration.
- Kapitel 10 behandelt praktische Themen wie Supportstrukturen und technische Restriktionen.

Durch diese Struktur wird eine schrittweise Annäherung an das System ermöglicht – von den fachlichen Grundlagen über die Datenstrukturen bis hin zur konkreten Nutzung der API im Betrieb. Die beschriebenen Inhalte richten sich sowohl an fachliche Anwender als auch an technische Integratoren und unterstützen beide Perspektiven gleichermaßen.

Damit dient dieses Dokument als Brücke zwischen technischer Schnittstellenbeschreibung und fachlicher Anwendung im Betrieb.

3 Grundlagen

3.1 Organisationen und RICS- bzw. Organisationscodes

Im UIC-Sicherheitsportal werden alle beteiligten Akteure als Organisationen abgebildet. Jede Organisation wird dabei eindeutig über einen Organisationscode identifiziert.

Historisch erfolgt diese Identifikation über den sogenannten RICS-Code (Railway Interchange Coding System), einen international standardisierten Identifikator für Unternehmen im Bahn- und ÖPNV-Umfeld.

Seit dem 01.01.2026 wird der RICS-Code im europäischen Kontext sukzessive durch den Organisationscode der Europäischen Eisenbahnagentur (ERA) ersetzt. Dieser übernimmt die Funktion einer eindeutigen Identifikation von Organisationen im europäischen Eisenbahnsystem. Die Beantragung eigener Organisationscodes erfolgt bei der ERA unter https://www.era.europa.eu/domains/registers/ocr_en.

Im UIC-Sicherheitsportal sowie in der aktuellen API wird vorübergehend weiterhin der Begriff RICS-Code verwendet. Diese Codes sind fachlich als Organisationsidentifikatoren zu verstehen und werden perspektivisch datengleich zu ERA-Organisationscodes migriert.

Für den Organisationscode gelten folgende Einschränkungen: Der Wert muss alphanumerisch sein und eine Länge von 4 oder 5 Stellen haben.

Im weiteren Verlauf wird der Begriff „Organisationscode“ als übergreifender Begriff verwendet. Dieser umfasst sowohl den bestehenden RICS-Code als auch ERA-Organisationscodes. Lediglich im Datenkontext des Barcodes wird bis auf weiteres der RICS genutzt. Unabhängig von der konkreten Ausprägung des Identifikators gilt das Folgende:

Der Organisationscode übernimmt im System mehrere zentrale Funktionen:

- eindeutige Identifikation der ausgebenden Organisation (Issuer),
- Zuordnung von Tickets zu ihrer Ursprungsorganisation,
- Grundlage für Berechtigungsprüfungen innerhalb der API,
- sowie Referenz für Kontroll- und Statusprozesse.

Jede Organisation wird im System genau einmal angelegt und mit ihrem Organisationscode verknüpft. Auf dieser Basis wird festgelegt, welche Rolle eine Organisation im Gesamtsystem einnimmt.

Dabei kann eine Organisation grundsätzlich folgende Rollen übernehmen:

- ausgebende Organisation: erstellt und verantwortet die fachlich korrekte Ausgabe von Tickets,
- kontrollierende Organisation: prüft Tickets im operativen Betrieb,
- oder eine Kombination aus beiden Rollen.

Die Rollen einer Organisation sind nicht nur fachlich relevant, sondern wirken sich auch direkt auf die technischen Berechtigungen innerhalb der API aus. So kann beispielsweise nur eine als ausgebend gekennzeichnete Organisation Tickets erzeugen oder Statusänderungen für eigene Tickets durchführen.

Ein zentrales Prinzip ist dabei die klare Verantwortungszuordnung: Eine Organisation kann ausschließlich auf Tickets zugreifen, die ihrem eigenen Organisationscode zugeordnet sind.

Dadurch wird sichergestellt, dass:

- Verantwortlichkeiten eindeutig abgegrenzt sind,
- keine unberechtigten Zugriffe auf fremde Tickets erfolgen
- und die Integrität der Ticket- und Statusinformationen gewährleistet bleibt.

Der Organisationscode bildet somit die zentrale Klammer zwischen fachlicher Rolle, technischer Identität und operativer Nutzung im UIC-Sicherheitsportal.

3.2 API-Key und Berechtigungen

Für die Nutzung der API des UIC-Sicherheitsportals werden organisationsbezogene Zugangsdaten bereitgestellt. Diese erfolgen in Form von API-Schlüsseln, JWT- oder OAuth2-Zugängen und dienen dazu, anfragende Systeme eindeutig einer Organisation zuzuordnen. Jeder Zugriff auf die API erfolgt dabei immer im Kontext einer Organisation und ist damit unmittelbar mit dem jeweiligen Organisationscode verknüpft. Auf diese Weise ist jederzeit nachvollziehbar, welche Organisation eine bestimmte Aktion ausführt. Innerhalb einer Organisation können mehrere technische Zugänge eingerichtet werden. So ist es möglich, unterschiedliche Systeme oder auch externe Dienstleister getrennt anzubinden und individuell zu steuern. Jeder dieser Zugänge kann dabei mit eigenen Berechtigungen ausgestattet werden.

Die API unterstützt verschiedene Verfahren zur Authentifizierung. In einfacheren Fällen erfolgt der Zugriff über einen API-Schlüssel, während für umfangreichere oder sicherheitsrelevante Anwendungsfälle in der Regel JWT und für die Barcodeerzeugung ausschließlich OAuth2 eingesetzt wird. Unabhängig vom Verfahren ist sichergestellt, dass jeder Aufruf eindeutig einem technischen Zugang zugeordnet werden kann.

Mit jedem technischen Zugang sind bestimmte Berechtigungen verbunden. Diese legen fest, welche Funktionen der API genutzt werden dürfen. Die Berechtigungen orientieren sich dabei an der fachlichen Rolle der Organisation im System. Das System gibt dabei bestimmte Rahmenbedingungen vor. So können nur Organisationen, die als ausgebend gekennzeichnet sind, Funktionen zur Änderung des Ticketstatus nutzen, etwa das Sperren oder Entsperren von Tickets. Organisationen, die ausschließlich kontrollieren, haben diese Möglichkeiten nicht. Eine vollständige Darstellungen dazu ist in Tabelle 2 in Kapitel 6 zu finden.

Darüber hinaus gilt grundsätzlich, dass eine Organisation nur mit denjenigen Tickets arbeiten kann, die ihr selbst zugeordnet sind. Eine Bearbeitung von Tickets anderer Organisationen ist nicht möglich. Das System selbst gibt bestimmte Einschränkungen für die Berechtigungen eines API-Key vor:

- Wenn eine Organisation nicht als "Ticket-Aussteller" gekennzeichnet ist, kann sie keinen mit der Berechtigung zum Erzeugen, Stornieren, Sperren oder Entsperren von Tickets erstellen.
- Jede Organisation kann nur Tickets stornieren, sperren oder entsperren, die zu dieser Organisation gehören (der RICS-Code des Tickets muss mit dem der Organisation übereinstimmen).

Ein wesentliches Prinzip ist die Trennung zwischen der Organisation als fachlicher Einheit und dem technischen Zugang. Die Organisation trägt die Verantwortung für alle Aktionen, die über ihre Zugänge ausgeführt werden, unabhängig davon, welches System diese tatsächlich ausführt.

Für die Integration bedeutet dies, dass jeder API-Aufruf korrekt authentifiziert sein muss und nur im Rahmen der vergebenen Berechtigungen erfolgen kann. Wenn diese Voraussetzungen nicht erfüllt sind, wird der Zugriff entsprechend abgelehnt.

3.3 Identifikation von Tickets

Im gesamten System wird ein Ticket eindeutig über eine Kombination von drei Merkmalen identifiziert. Maßgeblich sind dabei der Organisationscode des ausgebenden Unternehmens, die TicketId sowie das Gültigkeitsende des Tickets. Diese Kombination stellt sicher, dass jedes Ticket im System eindeutig referenziert werden kann, auch dann, wenn einzelne Merkmale für sich genommen nicht eindeutig sind.

Die TicketId entspricht dabei der im UIC-Barcode enthaltenen IssuerPNR und wird durch das ausgebende System vergeben. Da diese Kennung nicht global eindeutig sein muss, ist die Kombination mit dem Organisationscode (issuerNum im UIC-Barcode) erforderlich, um eine eindeutige Zuordnung zu gewährleisten. Für den Fall, dass der UIC-Barcode kein Feld issuerNum enthält, ist das Feld securityProviderNum heranzuziehen.

Das Gültigkeitsende des Tickets (validTo im UIC-Barcode) ist ebenfalls Bestandteil der Identifikation. Dies ist insbesondere deshalb relevant, da identische TicketIds zu unterschiedlichen Zeitpunkten erneut verwendet werden können, beispielsweise bei wiederkehrenden Produkten wie Abonnements. Durch die Einbeziehung des Gültigkeitsendes wird sichergestellt, dass unterschiedliche Ausprägungen eines Tickets eindeutig unterschieden werden können. Hierbei ist zu beachten, dass das Gültigkeitsende im Barcode relativ zum Gültigkeitsbeginn angegeben wird. Der Gültigkeitsbeginn wiederum wird relativ zum Ausgabezeitpunkt angegeben. Gültigkeitsbeginn und -ende sind also im Kontrollprozess zu berechnen.

Tabelle 1: Beispiel für die Gültigkeitsberechnung

Feldname	Feldwert	resultierendes Datum
issuingYear	2026	-
issuingDay	123	03.05.2026
validFromDay	5	08.05.2026
validUntilDay	1	09.05.2026

Ein wesentliches Prinzip der Identifikation von Tickets ist, dass nur die Werte verwendet werden, die im UIC-Barcode selbst enthalten sind. Maßgeblich ist hierbei der im Barcode enthaltene FCB-Datensatz.

Eine Ableitung, Veränderung oder Neuinterpretation dieser Werte im aufrufenden System ist nicht zulässig, da dies zu fehlerhaften Zuordnungen führen kann. Insbesondere bei der Validierung, Sperrung oder Nachverfolgung von Tickets ist es entscheidend, dass exakt die im Ticket codierten Werte verwendet werden.

Die Ticket-Identifikation bildet damit die Grundlage für alle weiteren Prozesse im UIC-Sicherheitsportal. Sämtliche Operationen, wie beispielsweise die Prüfung eines Tickets, das Sperren oder Entsperren sowie die Verarbeitung von Kontrollereignissen, beziehen sich immer auf diese eindeutige Kombination von Merkmalen.

3.4 Datenverwendung und Datenschutz

Das UIC-Sicherheitsportal ist mit dem Fokus auf Datensparsamkeit entwickelt worden. Es werden nur diejenigen Daten erfasst und gespeichert, die für die Funktionalitäten des Portals zwingend notwendig sind. Besonderes Augenmerk ist dabei auf personenbezogene Daten im Sinne der DSGVO gelegt worden. Beispielsweise werden Daten von Fahrgästen, die bei der Generierung von Barcodes benötigt werden, zu keinem Zeitpunkt gespeichert oder geloggt. Dementsprechend können auch erzeugte Barcodes nicht erneut abgerufen werden. Im Falle eines Verlusts von Barcodedaten muss ein neuer Request erfolgen und das verlorene Ticket ggf. storniert werden.

Weitere Informationen zur Datenverwendung und zum Datenschutz sind in **Anlage 4.1** zu finden. Die Datenschutzerklärung des UIC-Sicherheitsportals kann unter https://sicherheitsportal.deutschlandtarifverbund.de/privacy-policy abgerufen werden.

4 Nutzung des Portal-Frontends

Nach erfolgreicher Authentifizierung im UIC-Sicherheitsportal wird dem Benutzer das Frontend als zentrale Arbeitsoberfläche bereitgestellt.

Das Frontend dient der Verwaltung von Organisationen, Benutzern und technischen Zugängen sowie der Unterstützung operativer Prozesse wie Ticketprüfung und Sperrlistenverarbeitung. Es ergänzt die API um eine visuelle Oberfläche, ersetzt diese jedoch nicht.

4.1 Anlage und Einrichtung von Organisationen

Für die Nutzung des UIC-Sicherheitsportals ist zunächst die Anlage einer Organisation erforderlich. Diese bildet die Grundlage für alle weiteren Aktivitäten, wie die Ausstellung oder Kontrolle von Tickets sowie die Nutzung der API.

Das UIC-Sicherheitsportal richtet sich an Verkehrsunternehmen, Verkehrsverbünde und weitere Marktteilnehmer, die Barcodetickets im UIC-Standard ausgeben oder kontrollieren. Dabei ist zwischen zwei Rollen zu unterscheiden: Ticketaussteller und Ticketkontrolleur.

- Ticketaussteller sind für die Erstellung und Ausgabe von Tickets verantwortlich und benötigen für die Nutzung des Produktivsystems eine vertragliche Vereinbarung mit der Deutschlandtarifverbund GmbH (DTVG).
- Ticketkontrollierende Unternehmen hingegen können das System ohne eigene Tarifteilnahme nutzen, da für sie insbesondere die Kontrollfunktion im Vordergrund steht. Für sie steht eine vereinfachte Nutzungsvereinbarung zur Verfügung, die vorrangig Fragen der Datennutzung und -weitergabe regelt.

Die Anlage einer Organisation im UIC-Sicherheitsportal erfolgt ausschließlich durch die DTVG. Verbundunternehmen des Deutschlandtarifs können hierfür ein Stammdatenblatt übermitteln oder sich direkt an die DTVG wenden. Für neue Teilnehmer ist im Vorfeld eine entsprechende vertragliche Grundlage erforderlich.

Für die Einrichtung einer Organisation werden immer folgende Informationen benötigt:

- der Name des Unternehmens,
- der Organisationscode (RICS- bzw. ERA-Organisationscode),
- sowie zwei verantwortliche Personen, die als Organisationsadministratoren (Org-Admins) fungieren.

Die benannten Org-Admins werden im System hinterlegt und per E-Mail zur Registrierung eingeladen. Die **Einladung** ist zeitlich begrenzt und **muss innerhalb von 24 Stunden angenommen werden**. Danach läuft der Registrierungslink ab und muss neu versendet werden.

Die DTVG fungiert darüber hinaus als zentrale Instanz für die Vergabe von Rollen und Berechtigungen sowie für die Sicherstellung der Einhaltung der definierten Standards.

Die Einrichtung der Organisation kann sowohl für das Produktivsystem als auch für das Testsystem erfolgen, sodass eine getrennte Nutzung beider Umgebungen möglich ist.

4.2 Nutzeraccounts

Benutzer des UIC-Sicherheitsportals haben persönliche Accounts für den Zugang. Nutzeraccounts können einer oder mehreren Organisationen zugeordnet werden. Dies erfolgt ausschließlich durch Einladung in die Organisation durch einen OrgAdmin.

Die Organisation, in der ein Nutzer aktuell angemeldet ist, wird im Portal in jeder Ansicht angezeigt. Nutzer können ihre aktive Organisation im eingeloggten Zustand wechseln und erhalten so Zugriff auf die jeweiligen Funktionen, für dessen Nutzung die jeweilige Organisation berechtigt ist.

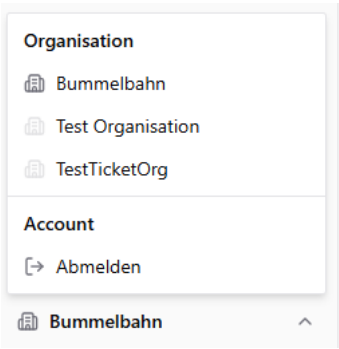

Abbildung 1: Anzeige und Wechsel der aktiven Organisation im Frontend

Nutzer können sich im UIC-Sicherheitsportal auch selbst registrieren. Zugang zu einer Organisation erhalten sie aber immer nur durch Einladung eines OrgAdmins.

4.3 Benutzer- und Rollenverwaltung

Im UIC-Sicherheitsportal wird zwischen zwei fachlichen Rollen unterschieden:

- **Ticketaussteller**, mit Berechtigung zur Erstellung und Verwaltung von Tickets,
- **Ticketkontrolleur**, mit Berechtigung zur Durchführung von Ticketprüfungen.

Eine Organisation kann eine oder beide Rollen wahrnehmen. Die Rollen bestimmen die verfügbaren Funktionen im System. Die initiale Vergabe erfolgt durch die DTVG im Rahmen der Organisationsanlage.

Für jede Organisation werden zwei Organisationsadministratoren (Org-Admins) benannt. Nach initialer Anlage durch die DTVG übernehmen die Org-Admins die weitere Verwaltung ihrer Organisation im Portal. Das bedeutet insbesondere, dass sie für die Einladung und Löschung von weiteren Benutzern verantwortlich sind. Die weiteren Funktionen des Portals wie die Erstellung und Verwaltung von API-Schlüsseln sowie die Nutzung der für die Organisation freigeschalteten Funktionen sind auch von weiteren Benutzern nutzbar. Auf Wunsch können weitere Benutzer durch die DTVG auch mit eingeschränkten Rechten angelegt werden. Aktuell ist der Zugriff auf die Benutzerübersicht sowie auf die API-Key-Verwaltung individuell steuerbar.

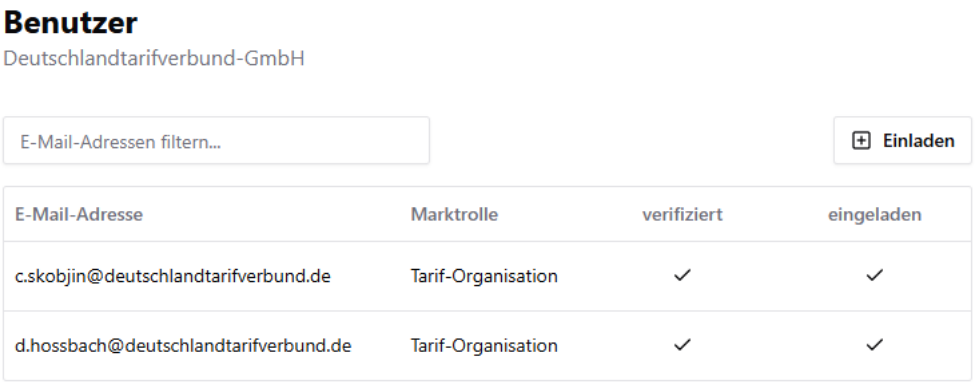

Abbildung 2: Benutzermanagement im UIC-Sicherheitsportal

Rollen, Berechtigungen und organisatorische Zuordnungen sind im persönlichen Account einsehbar.

4.4 Dashboard & Funktionsbereiche

Nach der Anmeldung wird dem Benutzer ein Dashboard als Einstiegsseite angezeigt. Das Dashboard bietet einen Überblick über die für den Benutzer relevanten Informationen und ist auf die zugewiesene Rolle sowie die zugehörigen Organisationen beschränkt.

Es dient der Orientierung und als Ausgangspunkt für die Nutzung der weiteren Funktionen des Portals.

Die Nutzung des Portals erfolgt über verschiedene Funktionsbereiche, die nach fachlichen Aufgaben gegliedert sind. Die verfügbaren Funktionen sind abhängig von der Rolle des Benutzers und umfassen insbesondere:

- Verwaltung von Organisationen und Benutzern,
- Verwaltung von API-Schlüsseln und technischen Zugängen,
- Erstellung, Sperrung und Prüfung bzw. Suche von Tickets (für berechtigte Rollen),
- sowie der Zugriff auf Sperrlisten.

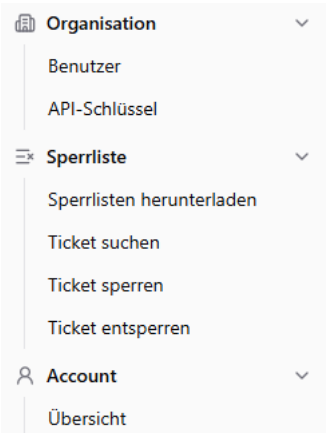

Abbildung 3: Funktionsbereiche im Frontend

Die Funktionsbereiche sind so strukturiert, dass sowohl administrative Aufgaben als auch operative Prozesse im laufenden Betrieb unterstützt werden.

Die bereitstehenden operativen Funktionen umfassen:

- Zugriff auf Sperrlisten,
- Suche nach Tickets,
- manuelles Sperren von Tickets,
- manuelles Entsperren von Tickets,
- sowie die Erstellung und Verwaltung der API-Keys.

Die Verfügbarkeit der Funktionen ist abhängig von der Rolle und den Berechtigungen des Benutzers.

Sperrlisten

Sperrlisten stellen eine zentrale Grundlage für die Ticketkontrolle dar. Sie enthalten Tickets, die als ungültig markiert wurden und im Rahmen der Kontrolle entsprechend berücksichtigt werden müssen. Im Portal können aktuelle Sperrlisten eingesehen und heruntergeladen werden. Dabei werden verschiedene Versionen der Sperrlisten bereitgestellt, die sich anhand ihres Erstellungszeitpunkts und der auf ihnen vermerkten Ticketsortimente unterscheiden. Die Sperrlisten können in unterschiedlichen Formaten (json, csv) bereitgestellt werden. Über ein Auswahlfeld kann konfiguriert werden, welche Ticketsortimente enthalten sein sollen. Sie dienen der Integration in Kontrollsysteme sowie der Unterstützung von Offline-Kontrollen.

Abbildung 4: Download von Sperrlisten im Frontend

Ticketsuche

Über die Ticketsuche können einzelne Tickets gezielt im System recherchiert werden. Die Suche erfolgt anhand der **Identifikationsmerkmale Ticket-ID, Ticketreferenz oder Bestellnummer in Kombination mit dem Organisationscode**. Als Ergebnis werden die zum Ticket vorhandenen Informationen angezeigt, insbesondere der aktuelle Status sowie die Kontrollhistorie. Die Kontrollhistorie enthält Informationen über durchgeführte Prüfungen, einschließlich Zeitpunkt, Organisation und Art der Kontrolle.

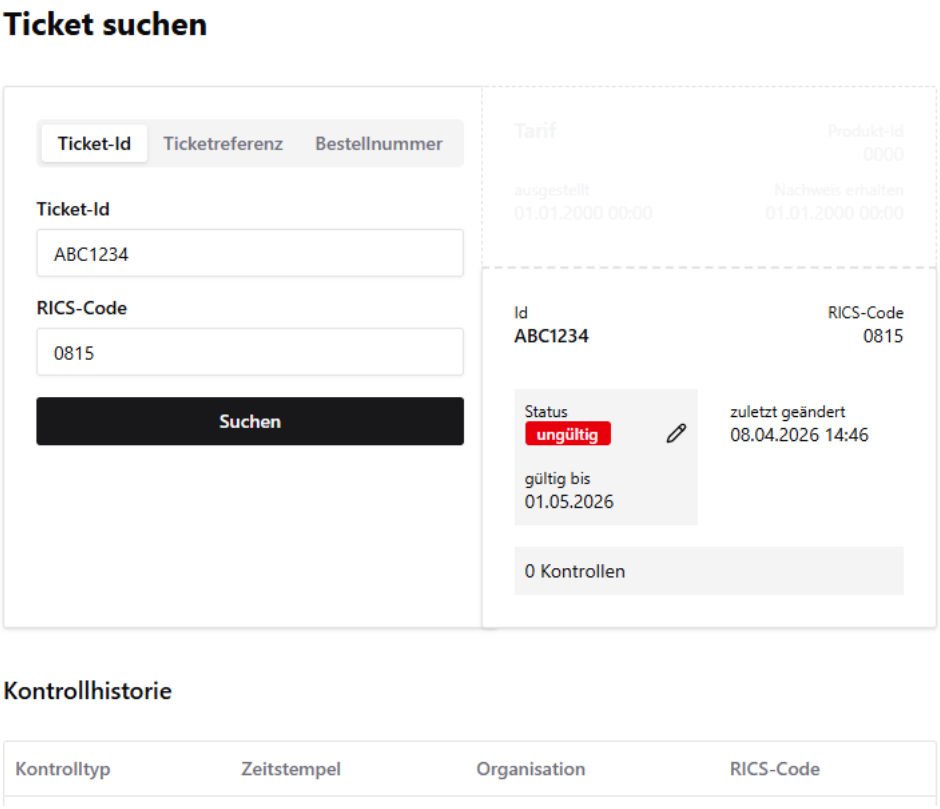

Abbildung 5: Ticketsuche im Frontend

Falls mehrere Tickets mit denselben Identifikationsmerkmalen in der Datenbank gefunden werden, muss das gesuchte Ticket anhand des Gültigkeitsendes ausgewählt werden.

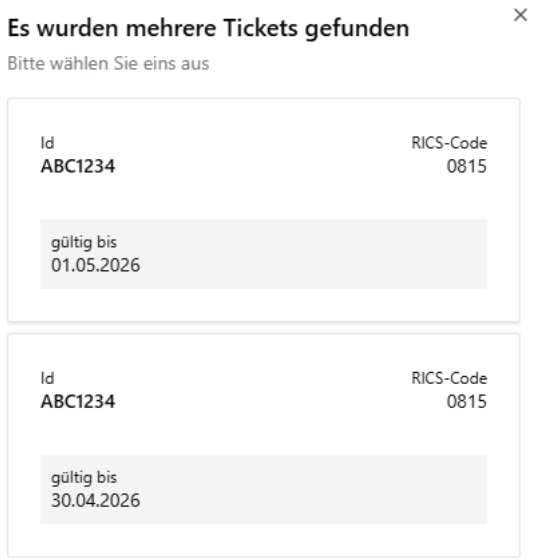

Abbildung 6: Auswahlbildschirm in der Ticketsuche

Ticket sperren

Tickets können im Portal manuell gesperrt werden, um diese als ungültig zu kennzeichnen. Für die Sperrung eines Tickets werden die **eindeutige Ticket-ID, der Organisationscode sowie der letzte Gültigkeitstag** des Tickets benötigt.

Durch die Sperrung wird das Ticket in die Sperrlisten aufgenommen und bei zukünftigen Kontrollen als ungültig bewertet.

Achtung: Es können auch Tickets gesperrt werden, für die kein Ausgabenachweis vorliegt. Es ist daher darauf zu achten, bei der Angabe der Ticketdaten **exakt die Werte aus dem UIC-Barcode** zu verwenden. Tickets, die über das UIC-Sicherheitsportal ausgegeben wurden, können in der Ticketsuche gefunden und dort gesperrt werden, ohne dass die Gefahr besteht, abweichende Daten anzugeben.

Deutschlandtickets, die über das UIC-Sicherheitsportal gesperrt werden, werden ebenfalls an den Sperrlistenservice KOSE der VDV eTicket Service gemeldet und dort auf die Sperrliste geschrieben.

Ticket entsperren

Die Entsperrung eines Tickets ermöglicht es, zuvor gesperrte Tickets wieder als gültig zu kennzeichnen. Hierfür wird das entsprechende Ticket zunächst im System gesucht. Auf Basis der vorhandenen Ticketinformationen kann der Status anschließend angepasst werden. Die Entsperrung wirkt sich direkt auf die Bewertung des Tickets im Kontrollprozess aus und wird in den nachfolgenden Sperrlisten entsprechend berücksichtigt. Auch in an den UIC-Sicherheitsportal angebundenen Sperrlistenservice KOSE der VDV eTicket Service wird das Ticket über eine Entsperr-Meldung von der Sperrliste genommen.

API-Keys und technische Zugänge

Für die Anbindung externer Systeme an das UIC-Sicherheitsportal werden API-Keys als technische Zugangsdaten verwendet. Die Verwaltung der API-Keys erfolgt organisationsbezogen und wird durch die Organisationsadministratoren gesteuert.

Ein API-Key dient der Authentifizierung gegenüber dem Portal und legt fest, welche Funktionen durch ein angebundenes System genutzt werden dürfen. Bei der Erstellung eines API-Key werden folgende Eigenschaften festgelegt:

- ein Anzeigename zur Identifikation des Schlüssels,
- der Typ des Schlüssels (Standard oder OAuth2),
- die zugehörigen Berechtigungen,
- sowie optional ein Ablaufdatum.

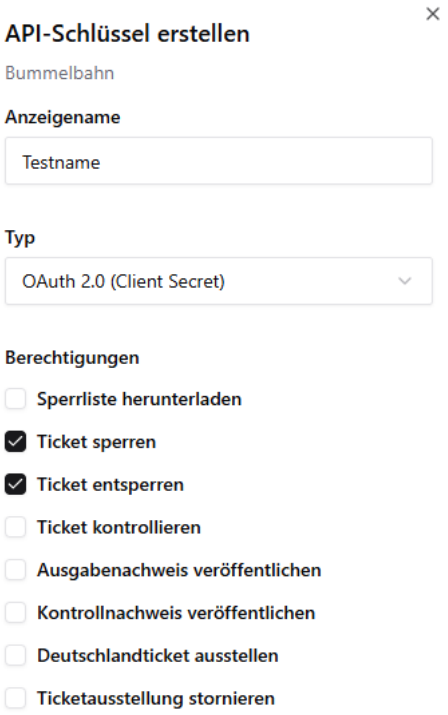

Abbildung 7: Konfiguration von API-Keys im Frontend

Die Berechtigungen bestimmen die zulässigen Aktionen, wie beispielsweise das Herunterladen von Sperrlisten, das Sperren von Tickets oder die Durchführung von Ticketkontrollen. Die Vergabe der Berechtigungen erfolgt entsprechend der fachlichen Rolle der Organisation und des vorgesehenen Einsatzzwecks. Für die Erstellung oder Löschung eines API-Key ist eine zusätzliche Absicherung durch Zwei-Faktor-Authentifizierung erforderlich.

Für die erstmalige Nutzung der API mit OAuth2 ist es erforderlich, dass der Login für den jeweiligen Schlüssel im Portal einmalig aktiviert wird. Andernfalls kann die API-Anfrage zur Erzeugung eines Tokenpaars nicht erfolgreich verarbeitet werden, auch wenn ein gültiger API-Key verwendet wird.

API-Keys bilden die Grundlage für die Integration externer Systeme und die Nutzung der in den folgenden Kapiteln beschriebenen Schnittstellen.

5 API-Spezifikation und Versionierung

Die API des UIC-Sicherheitsportals wird durch eine OpenAPI-Spezifikation beschrieben. Diese Spezifikation ermöglicht es, für die meisten Programmiersprachen Quellcode zu generieren und beschreibt detailliert, wie die einzelnen Endpunkte verwendet werden und welche technischen Einschränkungen gelten. Die Spezifikation bildet damit die technische Grundlage für die Integration der API in bestehende Systeme.

Die Spezifikation kann unter folgender URL eingesehen und heruntergeladen werden:

https://sicherheitsportal.deutschlandtarifverbund.de/scalar/v1

Eine benutzerfreundliche UI zum Testen der Endpunkte ist unter demselben Link ebenfalls verfügbar.

Die OpenAPI-Spezifikation beschreibt ausschließlich die technische Sicht auf die API. Sie definiert, wie Endpunkte aufgerufen werden und welche Daten dabei übertragen werden.

Für das Verständnis der fachlichen Zusammenhänge – insbesondere im Hinblick auf den Ticket-Lebenszyklus (Ausgabe, Kontrolle, Statusänderung), die Rolle der beteiligten Systeme sowie die Bedeutung einzelner Datenfelder im Kontrollprozess – ist eine ergänzende fachliche Einordnung erforderlich. Diese erfolgt in den nachfolgenden Kapiteln dieses Dokuments.

Die API fungiert als zentrale Schnittstelle zwischen ausgebenden Systemen (z. B. Vertriebssystemen), kontrollierenden Systemen (z. B. Kontrollgeräten oder Apps) und dem UIC-Sicherheitsportal als zentraler Sicherheits- und Referenzplattform. Die OpenAPI-Spezifikation stellt sicher, dass diese Systeme auf einer einheitlichen technischen Grundlage miteinander kommunizieren können, unabhängig von ihrer konkreten Implementierung.

Weiterentwicklung der API

Die API ist versioniert, so dass auch bei Weiterentwicklungen sichergestellt ist, dass die angebundenen Systeme weiterhin die Funktionen aufrufen können. Die Verpflichtung zur Umsetzung von Anpassungen an den Systemanbindungen ergibt sich aus den jeweiligen Verträgen.

In der Praxis werden relevante Anpassungen an bestehenden Schnittstellen mit großem Vorlauf von mehreren Monaten in der technischen Arbeitsgruppe UIC sowie den Gremien der DTVG vorgestellt, diskutiert und abgestimmt. Nach Abschluss der Abstimmung wird den nutzenden Organisationen die neue Dokumentation im Entwurfsstadium zur Verfügung gestellt sowie das geplante Änderungsdatum kommuniziert. Die neue Version des Dokuments wird dann im zuständigen Gremium der DTVG freigegeben und offiziell an die Vertragspartner verteilt.

Die im Gegensatz zu diesem Vorgehen festgehaltene kurze Frist im Vertrag ist gewählt, um schnell Erweiterungen des Portals und seiner Funktionen produktiv nehmen zu können. Ein Beispiel hierfür ist ein neuer Barcode-Endpunkt für eine neue Partner-Tariforganisation.

6 Authentifizierung

Der Zugriff auf die API des UIC-Sicherheitsportals erfordert eine eindeutige Authentifizierung des aufrufenden Systems. Ziel ist es, sicherzustellen, dass jede Anfrage einer Organisation und einem konkreten technischen Zugang zugeordnet werden kann.

Die Authentifizierung bildet damit die Grundlage für eine sichere Nutzung der API und stellt sicher, dass nur berechtigte Systeme auf Funktionen wie Ticketausstellung, Kontrolle oder Statusänderungen zugreifen können.

Gleichzeitig ist die Authentifizierung eng mit der Vergabe von Berechtigungen verknüpft. Erst durch das Zusammenspiel beider Aspekte wird festgelegt, welche Organisation welche Aktionen im System durchführen darf.

Das UIC-Sicherheitsportal unterstützt drei unterschiedliche Verfahren zur Authentifizierung, um verschiedenen technischen Anforderungen und Einsatzszenarien gerecht zu werden: API-Keys, JWT und OAuth2.

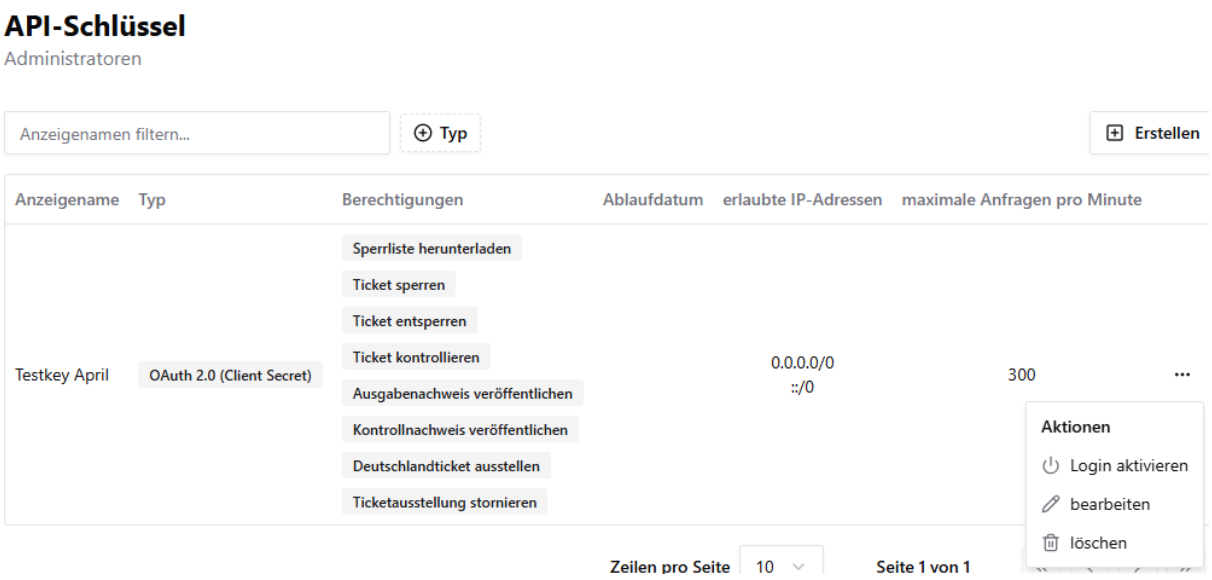

Abbildung 8: API-Key Verwaltung im Frontend

Mit OAuth2 können alle Endpunkte authentifiziert werden, während die möglichen Endpunkte mit JWT und API-Key eingeschränkt sind.

Tabelle 2: Überblick über die Endpunkte und die jeweils erforderlichen Rollen und Berechtigungen sowie die möglichen Authentifizierungsmethoden

HTTP-Methode	Pfad	Beschreibung	Authentifizierung	Nötige Rolle	Erforderliche Berechtigung
POST	/api/v1/auth/token	OAuth2-Endpunkt zur Anforderung eines Tokens über Client-Credentials, mit einem Refresh-Token oder einem signierten JSON Web Token.	Request Parameter im Body: • client_id • client_secret • grant_type • assertion	Keine	Keine
GET	/api/v1/blacklist	Gibt eine Liste aller verfügbaren Sperrlisten zurück (üblicherweise die letzten zwei Wochen).	API-Key / JWT / OAuth2	Kontrolleur	Sperrliste herunterladen
GET	/api/v1/blacklist/{id}	Gibt den Inhalt einer bestimmten Sperrliste zurück.	API-Key / JWT / OAuth2	Kontrolleur	Sperrliste herunterladen
GET	/api/v1/blacklist/latest	Gibt den Inhalt der neuesten Sperrliste zurück.	API-Key / JWT / OAuth2	Kontrolleur	Sperrliste herunterladen
POST	/api/v1/ticket/lock	Startet eine Anfrage zum Sperren einer Sammlung von Tickets. Die Tickets werden kurz darauf zur Sperrliste hinzugefügt.	API-Key / JWT / OAuth2	Ausgeber	Ticket sperren
POST	/api/v1/ticket/unlock	Startet eine Anfrage zum Entsperren einer Sammlung von Tickets. Die Tickets werden kurz darauf von der Sperrliste entfernt.	JWT / OAuth2	Ausgeber	Ticket entsperren
POST	/api/v1/ticket/cancel	Mit diesem Endpunkt kann eine Ticketstornierung angefordert werden. Die Stornierung kann nicht rückgängig gemacht werden. Die übergebenen Tickets werden gesperrt und auf die Sperrliste geschrieben.	JWT / OAuth2	Ausgeber	Ticketausstellung stornieren

POST	/api/v1/ticket/issuance-record	Mit Aufruf dieses Endpunkts werden ein oder mehrere Ausgabenachweise im Sicherheitsportal erfasst.	JWT / OAuth2	Ausgeber	Ausgabenachweis veröffentlichen
POST	/api/v1/ticket/issuance/dticket	Mit Aufruf dieses Endpunkts wird ein Deutschlandticket ausgestellt.	OAuth2	Ausgeber	Deutschlandticket ausstellen
POST	/api/v1/ticket/issuance/dtv	Mit Aufruf dieses Endpunkts wird ein D-TARIF-Ticket ausgestellt.	OAuth2	Ausgeber	Ticket ausstellen
POST	/api/v1/ticket/issuance/kvv	Mit Aufruf dieses Endpunkts wird ein KVV-Tarif-Ticket ausgestellt.	OAuth2	Ausgeber	Ticket ausstellen
POST	/api/v1/ticket/issuance/nox	Mit Aufruf dieses Endpunkts wird ein NOX-Ticket ausgestellt.	OAuth2	Ausgeber	Ticket ausstellen
POST	/api/v1/ticket/issuance/nds	Mit Aufruf dieses Endpunkts wird ein Niedersachsen-Tarif-Ticket ausgestellt.	OAuth2	Ausgeber	Ticket ausstellen
POST	/api/v1/ticket/issuance/sht	Mit Aufruf dieses Endpunkts wird ein Schleswig-Holstein-Tarif-Ticket ausgestellt.	OAuth2	Ausgeber	Ticket ausstellen
POST	/api/v1/validation/validate	Mit Aufruf dieses Endpunkts kann geprüft werden, ob das entsprechende Ticket gesperrt ist und infolgedessen auf der Sperrliste steht.	API-Key / JWT / OAuth2	Kontrolleur	Ticket kontrollieren
POST	/api/v1/validation/records	Mit Aufruf dieses Endpunkts werden ein oder mehrere Kontrollnachweise im Sicherheitsportal erfasst.	API-Key / JWT / OAuth2	Kontrolleur	Kontrollnachweis veröffentlichen

6.1 API-Key

Die Authentifizierung über API-Keys stellt eine einfache Möglichkeit dar, API-Aufrufe eindeutig einem technischen Zugang zuzuordnen. Die grundlegende Authentifizierung erfolgt dabei über einen API-Key, der bei jeder Anfrage im HTTP-Header „Authorization“ mitgeführt wird. Der API-Key wird direkt als Wert übergeben, ohne Präfix oder Suffix.

Auf diese Weise kann das System jede Anfrage eindeutig einem technischen Zugang und damit einer Organisation zuordnen. Auf dieser Basis werden die hinterlegten Berechtigungen geprüft. Fachlich eignet sich dieses Verfahren insbesondere für klar abgegrenzte Anwendungsfälle, bei denen ein System dauerhaft und ohne komplexen Authentifizierungsprozess auf bestimmte Funktionen zugreifen soll. Dabei ist zu beachten, dass der API-Key stets im Kontext der zugehörigen Organisation verwendet wird und alle darüber ausgeführten Aktionen dieser Organisation zugerechnet werden.

Die API-Keys werden im UIC-Sicherheitsportal für eine Organisation erzeugt und verwaltet. Die Einrichtung erfolgt in der Regel durch berechtigte Administratoren der jeweiligen Organisation. Fachlich ist der API-Key damit immer einem konkreten technischen Zugang innerhalb einer Organisation zugeordnet. Über diesen Zugang werden die zugehörigen Berechtigungen gesteuert. Die Organisation trägt die Verantwortung für die sichere Verwaltung ihrer API-Keys sowie für alle Aktionen, die über diese Zugänge durchgeführt werden.

6.2 OAuth2 mit Client Credentials

Bei der Authentifizierung über OAuth2 erfolgt der Zugriff mithilfe sogenannter Client Credentials, bestehend aus einer ClientId und einem ClientSecret. Mit diesen Zugangsdaten wird zunächst ein Zugriffstoken erzeugt, das anschließend für die eigentlichen API-Aufrufe verwendet wird.

Voraussetzung für die Nutzung dieses Verfahrens ist eine einmalige manuelle Freischaltung des Logins im Frontend des UIC-Sicherheitsportals. Nach dieser initialen Aktivierung kann die Authentifizierung mit Client Credentials einmalig durchgeführt werden. Im Anschluss ist der Login wieder deaktiviert, sodass ein erneutes Anmelden mit den Client Credentials im laufenden Betrieb nicht vorgesehen ist.

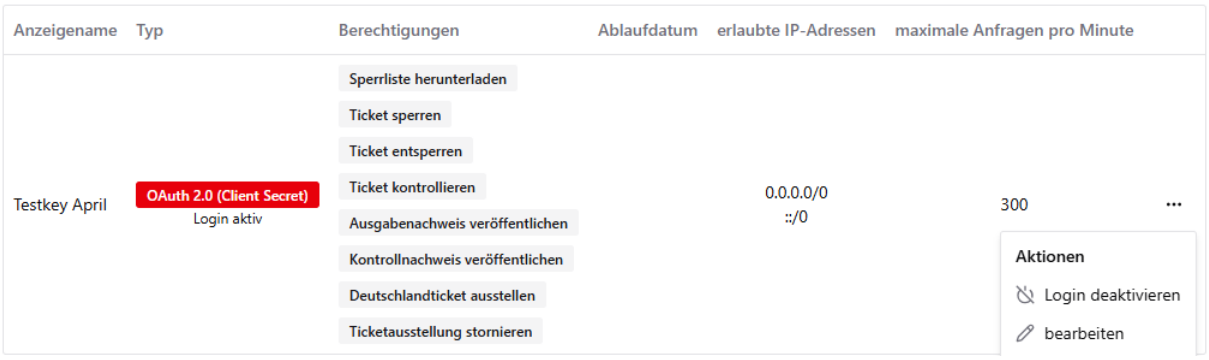

Abbildung 9: OAuth-Schlüssel mit aktivem Login

Das im Rahmen der Authentifizierung erzeugte Access-Token wird bei API-Aufrufen im HTTP-Header „Authorization“ mitgeführt. Dabei wird das Token mit dem Präfix „Bearer“ übergeben. Auf diese Weise kann das System die Anfrage eindeutig einem autorisierten technischen Zugang zuordnen.

Neben dem Access-Token wird auch ein Refresh-Token bereitgestellt. Dieser dient dazu, neue Zugriffstoken zu erzeugen, ohne dass eine erneute Authentifizierung mit den ursprünglichen Zugangsdaten erforderlich ist.

Fachlich bedeutet dies, dass Systeme nach der initialen Anmeldung nicht dauerhaft mit ihren Client Credentials arbeiten, sondern stattdessen auf Basis des Refresh-Tokens neue Access-Tokens beziehen. Dies erhöht die Sicherheit und reduziert die Notwendigkeit, sensible Zugangsdaten regelmäßig zu verwenden.

Access-Tokens sind nur für eine kurze Zeit gültig und haben eine Gültigkeitsdauer von 10 Minuten. Der Refresh-Token hingegen besitzt eine deutlich längere Gültigkeit und kann über einen Zeitraum von 30 Tagen hinweg verwendet werden, um einmalig ein neues Tokenpaar zu erzeugen. Dabei verliert der Refresh-Token erst dann seine Gültigkeit, wenn ein damit erstellter Access-Token erstmalig verwendet wurde.

Ein wesentliches Prinzip dieses Verfahrens ist, dass für jeden technischen Zugang immer nur ein gültiges Tokenpaar existiert. Sobald ein neues Tokenpaar erzeugt und verwendet wird, verlieren zuvor ausgestellte Tokens ihre Gültigkeit. Dies stellt sicher, dass Zugriffe eindeutig steuerbar bleiben und keine parallelen oder veralteten Sitzungen bestehen.

6.3 OAuth2 mit JWT Bearer Assertion

Eine weitere Möglichkeit zur Authentifizierung besteht darin, dass sich ein System nicht mit fest hinterlegten Zugangsdaten anmeldet, sondern sich durch ein signiertes Token gegenüber dem UIC-Sicherheitsportal ausweist.

Fachlich basiert dieses Verfahren auf einem Vertrauensverhältnis zwischen dem aufrufenden System und dem Sicherheitsportal. Das System erzeugt eigenständig ein sogenanntes Assertion Token und signiert dieses mit einem zuvor im Sicherheitsportal hinterlegten Schlüssel. Das Sicherheitsportal prüft diese Signatur und stellt bei erfolgreicher Prüfung ein Zugriffstoken aus. Voraussetzung hierfür ist, dass im Sicherheitsportal entsprechende Zugangsdaten eingerichtet sind und mindestens ein Public Key hinterlegt wurde, mit dem die Signatur überprüft werden kann. Der Prozess basiert auf dem Standard RFC 7523 und läuft in vereinfachter Form wie folgt ab:

Das aufrufende System erzeugt ein signiertes JWT und übermittelt dieses an den Token-Endpunkt. Nach erfolgreicher Prüfung stellt das Sicherheitsportal einen Access-Token aus, das für die anschließenden API-Aufrufe verwendet wird. Im Gegensatz zum Client-Credentials-Verfahren wird in diesem Fall kein Refresh-Token ausgegeben. Für jeden neuen Zugriff muss ein neues Assertion Token erzeugt und erneut ein Access-Token angefordert werden.

Fachlich eignet sich dieses Verfahren insbesondere für System-zu-System-Kommunikation, bei der bereits etablierte Sicherheitsmechanismen und Schlüsselverwaltung eingesetzt werden.

Tabelle 3: Inhalt des Assertion Token

Feld	Beschreibung	Datentyp	Pflichtfeld	Wertebereich
Header – kid	KeyId des verwendeten Schlüssels zur Erstellung der Signatur	Text	Ja	
Header – typ	Type – immer „JWT“	Text	Ja	JWT
Header – alg	Verwendeter Signatur Algorithmus	Text	Ja	PS256 oder RS256
Body – sub	Identifiziert das Credential im Sicherheitsportal.	UUID	Ja	

Body – aud	Beschreibt den Tokenendpunkt, für welchen dieser Token ausgestellt wurde.	URI	Ja	Je nach System die volle URL des Token-Endpunkts
Body – nbf	Gültigkeitsbeginn des Tokens	Timestamp	Ja	
Body – iat	Ausstellungszeitpunkt des Tokens	Timestamp	Ja	
Body – exp	Ablaufdatum des Tokens	Timestamp		
Body – iss	Issuer des Tokens. Kann bei der Konfiguration des Credentials frei festgelegt werden	Text	Ja	
Body – jti	Token Identifier. Wenn vorhanden, kann dieser Token nur genau einmal verwendet werden	UUID	Nein	

Tabelle 4: Beispiel – JSON/JWT


```
{
  "kid": "2e6b73d1-5951-418d-a24e-54db43ed1e6d",
  "typ": "JWT",
  "alg": "PS256"
}.
{
  "sub": "aa7a0db6-13fd-43e2-bb30-1866158bf4e7",
  "aud": "https://test.sicherheitsportal.deutschlandtarifverbund.de/api/v1/auth/token",
  "nbf": 1756996034,
  "iss": "aa7a0db6-13fd-43e2-bb30-1866158bf4e7",
  "exp": 1756996454,
  "iat": 1756996154,
  "jti": "deb4a7ec-471a-4489-92d0-64811c59d227"
}.$signature
```


Tabelle 5: Beispiel – Token-Endpunkt Request mit Assertion JWT

```
POST /api/v1/auth/token
(Form URL Encoded)

grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer&
assertion=eyJhb….
```

7 Datenschemata

Nachfolgend werden die Schemata der Daten abgebildet, die zwischen den angebundenen Systemen und dem UIC-Sicherheitsportal ausgetauscht werden. Dabei handelt es sich nicht um isolierte technische Objekte, sondern um Abbildungen der fachlichen Prozesse rund um die Ausstellung, Verwaltung und Kontrolle von Tickets.

Die einzelnen Schemata sind jeweils bestimmten Anwendungsfällen zugeordnet. Sie kommen beispielsweise bei der Ticketausstellung, der Validierung im Kontrollprozess oder bei der Verarbeitung von Kontroll- und Ausgabedaten zum Einsatz. Für das Verständnis ist entscheidend, dass die enthaltenen Felder nicht unabhängig voneinander betrachtet werden, sondern immer im jeweiligen fachlichen Kontext stehen. Die Bedeutung einzelner Werte ergibt sich daher häufig erst aus ihrem Zusammenspiel innerhalb eines Prozesses.

Die nachfolgenden Abschnitte unterscheiden zwischen Anfrage- und Antwortstrukturen. Während die Anfrageschemata beschreiben, welche Informationen an das UIC-Sicherheitsportal übermittelt werden, definieren die Antwortschemata die Rückmeldungen des Systems und bilden insbesondere die Grundlage für die fachliche Bewertung im Kontrollprozess.

7.1 Wiederkehrende Basistypen

Einige Felder werden in mehreren Datenschemata verwendet und sind hier einmalig detailliert erläutert. In den jeweiligen Datenschemata finden sich nur noch der Feldname und der Verweis auf diesen Darstellung bzw. eventuelle Besonderheiten im jeweiligen Schema.

Tabelle 6: Wiederkehrende Basistypen

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Wertebereich	Beispiel
uniqueOrderId	-	Eindeutige Kennung aus dem aufrufenden System. Falls zum übergebenen Wert ein Ticket existiert, wird das bereits vorhandene Ticket storniert.	string	UUID	"28d07a4f-02ee-4c6f-996d-89e47670157b"
ticketId	issuerPNR	IssuerPNR aus dem UIC-Barcode	string	Max. 32 Zeichen	"A0815BF0"
ticketReference	-	Benutzerdefinierte Referenz zum Ticket im UIC-Sicherheitsportal. Kann später	string	Max. 16 Zeichen	„ABO31651756“

		zur Identifikation/Suche nach dem Ticket verwendet werden. Der Wert muss nicht eindeutig sein.			
issuerAuthorizationId	issuerAuthorizationId	VDV KA Autorisations-ID des Tickets (Berechtigungsnummer)	int	0 – 2.147.483.647	15312312
issuerKvpId	extIssuerId	VDV KA Kundenvertragspartner-ID (KVP-ID)	int	4 Zeichen	6321
issuedAt	issuingDateTime (berechneter Wert aus issuingYear, issuingDay und issuingTime)	Ausgabezeitpunkt des Tickets	string	ISO 8601 Date	"2025-03-01T03:00:00+01:00"
validFrom **oder** monthOfValidity	validFromDateTime (berechneter Wert aus issuingYear, issuingDay, validFromDay)	Gültigkeitsbeginn des Tickets	string	ISO 8601 Date	"2025-03-01T03:00:00+01:00"
validTo **oder** validUntil	Berechneter Wert aus issuingYear, issuingDay, validFromDay und validUntilDay	Gültigkeitsende des Tickets	string	ISO 8601 Date	"2025-03-01T03:00:00+01:00"
rics **oder** issuerRics	issuer (Num oder IA5)	RICS- bzw. Organisationscode des Ticketausstellers	int oder string	4 Zeichen	
securityProviderRics	securityProvider (Num oder IA5)	RICS- oder Organisationscode des Barcodeerstellers	int oder string	4 Zeichen	„3634“
keyId	keyId	ID des verwendeten Signaturschlüssels	string	5 Zeichen	„DTX04“
firstName	firstName	Vorname des Ticketinhabers	string		„Max“
lastName	lastName	Nachname des Ticketinhabers	string		„Mustermann“

dateOfBirth	Zusammengesetzter Wert aus yearOfBirth, monthOfBirth, dayOfBirthInMonth	Geburtsdatum des Ticketinhabers	string	ISO 8601 Date	„1999-12-31“
gender	gender	Geschlecht des Ticketinhabers. Defaultwert = 0	int	0 = undefined, 1 = female, 2 = male, 3 = other	
passengerType	passengerType	Fahrgasttyp. Defaultwert = 0	int	0 = adult, 1 = senior, 2 = child, 3 = youth, 4 = dog, 5 = bike, 6 = freeAddOnPassenger, 7 = freeAddOnChild	
numberOfPassengers	numberOfPassengers	Anzahl Fahrgäste des Fahrgasttyps	int	1 - 200	
reductionCard	reductionCard	Angewendete Ermäßigungskarte für das Ticket	string		„BC25“, „Schülerausweis“
price **oder** priceInCents	price	Ticketpreis in Cent	int		6300
currency	currency	Währung des Ticketpreises	string	3 Zeichen	„EUR“
classCode	classCode	Wagenklasse des Ticket. Defaultwert = „second“	string	„first“, „second“	
productOwner	productOwner (Num oder IA5)	Produktverantwortliche Tariforganisation. Beim D-Ticket: RICS-Code	int oder string	4 Zeichen	“3634”, “KVV”

		Ber Verbund-tickets: Ver-bundkürzel			
productId	productId (Num oder IA5)	ProduktID des Tickets. Stamm-daten beachten! Beim D-Ticket gültige Werte: 9999 (D-Ticket), 9998 (D-Job-ticket), 9996 (D-Semesterticket), 9995 (D-Schülerticket)	int oder string	Max. 16 Zeichen	
tariff-Descrip-tion	tariffDesc	Name des Tarif-produktes. Wird abwei-chend vom ASN.1 in der Schnittstelle nur einmalig ange-geben.	string	Max. 32 Zeichen	„Einzelfahrt“, „Deutschlandticket“
tariffId	tariffId (Num oder IA5)	Preisstufe bzw. ergänzende In-formationen zum Tarifpro-dukt. Wird abwei-chend vom ASN.1 in der Schnittstelle nur einmalig ange-geben.	string	Max. 32 Zeichen	„5 Waben“, „inkl. Kostenloser Mit-nahme“, „Preisstufe 3“
returnIn-cluded	returnIncluded	Hin-/Rück-Indi-kator	boolea n		
isSecure-PaperTi-cket	securePaperTicket	Sicherheitspa-pier-Indikator. Muss true sein, wenn der Bar-code auf Sicher-heitspapier aus-gegeben wird. Defaultwert = false	boolea n		

Zusätzliche Erläuterungen

Für das Feld „**uniqueOrderId**“ gilt folgende fachliche Bedeutung:

- Die Kennung dient der eindeutigen Zuordnung eines Tickets im aufrufenden Fremdsystem.
- Wird bei einer erneuten Anfrage dieselbe uniqueOrderId verwendet und existiert hierzu bereits ein Ticket mit identischem Gültigkeitsende, wird das existierende Ticket storniert und durch das neu erzeugte Ticket ersetzt.
- Das Feld unterstützt damit die konsistente Fortschreibung von Vorgängen im Fremdsystem und verhindert doppelte oder widersprüchliche Ticketstände.
- Diese Angabe wird nicht in den Barcode geschrieben.

Für das Feld „**ticketReference**“ gilt folgende fachliche Regel:

- Es dient der optionalen Referenzierung eines Tickets im aufrufenden System.
- Es wird vom UIC-Sicherheitsportal nicht fachlich interpretiert und hat keinen Einfluss auf die Verarbeitung oder Bewertung des Tickets.
- Diese Angabe wird ebenfalls nicht in den Barcode geschrieben.

Für das Feld „**dateOfBirth**“ gelten folgende fachliche und technische Einschränkungen:

- Das Feld muss ein gültiges Datum enthalten.
- Das Datum muss mindestens einen Tag in der Vergangenheit liegen.
- Das Datum darf nicht vor dem 01.01.1901 liegen.

7.2 Anfrage

Die Anfrageschemata beschreiben die Daten, die von angebundenen Systemen an das UIC-Sicherheitsportal übermittelt werden, um eine konkrete Aktion auszuführen. Fachlich entsprechen diese Daten den Informationen, die im jeweiligen Prozessschritt benötigt werden, beispielsweise bei der Ticketausstellung oder der Kontrolle. Dabei ist entscheidend, dass die übermittelten Werte den tatsächlichen Ticketdaten entsprechen und unverändert aus dem UIC-Ticket übernommen werden. Abweichungen können dazu führen, dass Tickets nicht korrekt verarbeitet oder zugeordnet werden.

Die nachfolgenden Schemata definieren die Struktur der jeweiligen Anfragen und sind den einzelnen Anwendungsfällen zugeordnet.

7.2.1 Ticket

Das Schema „Ticket“ beschreibt die grundlegenden Identifikationsmerkmale eines Tickets innerhalb des UIC-Sicherheitsportals.

Fachlich dient es dazu, ein Ticket eindeutig zu referenzieren, beispielsweise im Rahmen einer Kontrolle, einer Sperrung oder einer Statusabfrage. Die enthaltenen Informationen entsprechen den im UIC-Barcode gespeicherten Werten und müssen unverändert übernommen werden. Nur so ist sichergestellt, dass das Ticket korrekt identifiziert und verarbeitet werden kann.

Das Schema bildet damit die Grundlage für alle Anfragen, bei denen ein konkretes Ticket adressiert wird.

Tabelle 7: Datenschema Ticket

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
rics	Besonderheit: Falls das Feld issuer im Barcode nicht belegt ist, muss stattdessen securityProvider ausgelesen werden.			Ja		
ticketId				Ja		
validTo				Ja		

Tabelle 8: Beispiel JSON Ticket

```
{
  "rics":"5143",
  "ticketId":"A0815BF0",
  "validTo":"2025-03-01T03:00:00+01:00"
}
```

7.2.2 Tickets

Das Schema „Tickets“ erweitert das einzelne Ticket auf eine Liste mehrerer Tickets und ermöglicht die gleichzeitige Verarbeitung mehrerer Vorgänge innerhalb einer Anfrage.

Tabelle 9: Datenschema Tickets

Feld	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
tickets	Liste der Tickets, welche in der jeweiligen Anfrage verarbeitet werden sollen	Array Typ: 7.2.1	Ja	1 – 10.000 Elemente innerhalb des Arrays	Siehe Beispiel – JSON

Tabelle 10: Beispiel JSON Tickets

```
{
  "tickets":[
    {
      "rics":"9999",
      "ticketId":"A0815BF0",
      "validTo":"2025-03-01T03:00:00+01:00"
    },
    {
      "rics":"9999",
      "ticketId":"BC93DE99",
      "validTo":"2025-03-01T03:00:00+01:00"
    }
  ]
}
```

7.2.3 Kontrolle

Das Schema „Kontrolle“ beschreibt die Daten, die im Rahmen einer Ticketprüfung an das UIC-Sicherheitsportal übermittelt werden. Fachlich wird dieses Schema immer dann verwendet, wenn ein Ticket aktiv geprüft wird, beispielsweise durch ein Kontrollgerät oder eine Kontroll-App im Fahrgastkontakt. Dabei bildet die Anfrage die konkrete Situation der Kontrolle ab. Neben den Ticketdaten werden Informationen wie Gültigkeitszeitraum, Produktbezug, Kontrollzeitpunkt und -ort übermittelt, damit das Ticket im richtigen Kontext bewertet werden kann.

Ein zentrales Unterscheidungsmerkmal ist, ob es sich um eine tatsächliche Kontrolle oder lediglich um eine informatorische Abfrage handelt. Bei einer echten Kontrolle wird der Vorgang im System gespeichert und fließt in die Kontrollhistorie des Tickets ein. Diese Informationen können später beispielsweise zur Erkennung von Mehrfachkontrollen oder auffälligen Nutzungsmustern herangezogen werden. Bei einer rein informatorischen Abfrage erfolgt zwar ebenfalls eine Speicherung, die Daten fließen aber nicht in die automatischen Analysen ein. Solche Abfragen werden typischerweise in administrativen Anwendungen oder bei der Analyse von Einzelfällen genutzt.

Die korrekte Verwendung dieses Schemas ist daher entscheidend dafür, ob Kontrollereignisse im System nachvollziehbar sind und für weitere Auswertungen zur Verfügung stehen.

Tabelle 11: Datenschema Kontrolle

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
rics	Besonderheit: Falls das Feld issuer im Barcode nicht belegt ist, muss stattdessen securityProvider ausgelesen werden.			Ja		
ticketId				Ja		
validFrom				Ja		
validTo				Ja		
productId				Ja		
tariffDescription				Ja		
issuedAt				Ja		
validatedAt	-	Kontrollzeitpunkt. Wenn leer, wird der aktuelle Zeitpunkt angenommen.	string	Nein	ISO 8601 Date	"2025-02-15T10:30:00+01:00"
keyId				Ja		
securityProviderRics				Ja		

location	-	Kontrollort	7.2.28	Nein		Siehe Beispiel-JSON
writeValida-tionRecord	-	Default = true	boolea n	Nein		

Tabelle 12: Beispiel JSON Kontrolle


```
{
  "rics":"5143",
  "ticketId":"A0815BF0",
  "validFrom":"2025-02-01T00:00:00+01:00",
  "validTo":"2025-03-01T03:00:00+01:00",
  "productId":9999,
  "tariffDescription":"Deutschlandticket",
  "issuedAt":"2025-01-25T02:00:00+01:00",
  "validatedAt":"2025-02-15T10:30:00+01:00",
  "keyId":"31A33",
  "securityProviderRics":"3634",
  "location":
   {
      "dhid":"de:01056:17016",
      "latitude":null,
      "longitude":null
   },
  "writeValidationRecord":true
}
```


Zusätzliche Erläuterungen

Es gilt für das Feld „**writeValidationRecord**“ folgende fachliche Bedeutung:

- Wenn der Wert auf „true“ (Default) gesetzt ist, wird der Vorgang als echte Kontrolle behandelt und als Kontrollereignis im System gespeichert.
- Wenn der Wert auf „false“ gesetzt ist, erfolgt lediglich eine informatorische Abfrage ohne Speicherung in der Kontrollhistorie.
- Die korrekte Verwendung dieses Feldes ist insbesondere für die Nachvollziehbarkeit von Kontrollen sowie für die Erkennung auffälliger Nutzungsmuster relevant.

7.2.4 Kontrollnachweise

Das Schema „Kontrollnachweise“ beschreibt die Übermittlung von bereits durchgeführten Kontrollen an das UIC-Sicherheitsportal. Dieses Schema wird verwendet, wenn Kontrollen nicht direkt im Moment der Prüfung an das System übermittelt werden, sondern gesammelt und zu einem späteren Zeitpunkt nachgereicht werden. Dies ist insbesondere bei nicht dauerhaft onlinefähigen Kontrollgeräten relevant.

Fachlich ermöglicht dies, dass auch nachträglich übermittelte Kontrollen in die Kontrollhistorie eines Tickets einfließen und für Auswertungen zur Verfügung stehen.

Die übermittelten Daten entsprechen inhaltlich einzelnen Kontrollvorgängen und werden im System wie reguläre Kontrollen verarbeitet.

Tabelle 13: Datenschema Kontrollnachweise

Feld	Beschreibung	Datentyp	Pflicht-feld	Wertebereich	Beispiel
Validation-Records	Liste der Tickets, für die Kontrollnachweise veröffentlicht werden sollen.	Array Typ: 7.2.3	Ja	1 – 10.000 Elemente innerhalb des Arrays	Siehe Beispiel – JSON

Tabelle 14: Beispiel JSON Kontrollnachweise


```
{
  "validationRecords":[
    {
      "rics":"5143",
      "ticketId":"A0815BF0",
      "validFrom":"2025-02-01T00:00:00+01:00",
      "validTo":"2025-03-01T03:00:00+01:00",
      "productId":9999,
      "tariffDescription":"Deutschlandticket",
      "issuedAt":"2025-01-25T02:00:00+01:00",
      "validatedAt":"2025-02-15T10:30:00+01:00",
      "keyId":"31A33",
      "securityProviderRics":"3634",
      "location":
       {
          "dhid":"de:01056:17016",
          "latitude":null,
          "longitude":null
       },
      "writeValidationRecord":true
    },
    {
      "rics":"5143",
```

```
      "ticketId":"BC93DE99",
      "validFrom":"2025-02-01T00:00:00+01:00",
      "validTo":"2025-03-01T03:00:00+01:00",
      "productId":9999,
      "tariffDescription":"Deutschlandticket",
      "issuedAt":"2025-01-25T02:00:00+01:00",
      "validatedAt":"2025-02-15T10:30:00+01:00",
      "keyId":"31A33",
      "securityProviderRics":"3634",
      "location":
       {
          "dhid":null,
          "latitude":"53.7",
          "longitude":"9.6"
       },
      "writeValidationRecord":true
    }
  ]
}
```

7.2.5 Ausgabenachweis

Das Schema „Ausgabenachweis“ beschreibt die Übermittlung von Informationen über die Ausstellung eines Tickets an das UIC-Sicherheitsportal. Dieses Schema wird verwendet, um ein selbst ausgestelltes Ticket nachträglich im System zu registrieren. Dies ist nur in Fällen relevant, in denen die Barcodeerstellung nicht über das UIC-Sicherheitsportal erfolgt, sondern in einem externen Vertriebssystem.

Fachlich dient der Ausgabenachweis dazu, die Existenz und die wesentlichen Merkmale eines Tickets im System bekannt zu machen. Dadurch kann das Ticket im Rahmen von Kontrollen korrekt zugeordnet und bewertet werden.

Der Ausgabenachweis bildet somit die Grundlage dafür, dass ein Ticket im Kontrollprozess vollständig berücksichtigt werden kann, auch wenn es außerhalb des Systems erzeugt wurde.

Tabelle 15: Datenschema Ausgabenachweis

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
rics				Ja		
ticketId				Ja		
validFrom				Ja		
validTo				Ja		
productId				Ja		
productOwner				Ja		
issuedAt				Ja		
keyId				Nein		
securityProviderRics				Ja		
postcode	-	Postleitzahl des Ticketinhabers	string	**Nein**	Min.: 5 Zeichen Max.: 5 Zeichen	"49661"
issuerAuthorizationId				Nein		
issuerKvpId				Nein		
ticketReference				Nein		
uniqueOrderId				Ja		

Tabelle 16: Beispiel JSON Ausgabenachweis


```
{
  "rics":"5143",
  "ticketId":"A0815BF0",
  "validFrom":"2025-02-01T00:00:00+01:00",
  "validTo":"2025-03-01T03:00:00+01:00",
  "productId":9999,
  "productOwner":"3634",
  "tariffDescription":"Deutschlandticket",
  "issuedAt":"2025-01-25T02:00:00+01:00",
  "keyId":"31A33",
  "securityProviderRics":"3634",
  "issuerAuthorizationId":15312312,
  "issuerKvpId":6321,
  "ticketReference":"ABO121618”
  "uniqueOrderId":"6e25585b-2e7e-4ea1-99c3-a5f743693f21"
}
```

7.2.6 Ausgabenachweise

Das Schema „Ausgabenachweise“ erweitert den einzelnen Ausgabenachweis auf eine Liste mehrerer Datensätze und ermöglicht die gebündelte Übermittlung mehrerer ausgestellter Tickets in einer Anfrage.

Tabelle 17: Datenschema Ausgabenachweis

Feld	Beschreibung	Datentyp	Pflicht-feld	Wertebereich	Beispiel
issuanceRecords	Liste der Tickets, für die Ausgabe-nachweise veröf-fentlicht werden sollen.	Array Typ: 7.2.5	Ja	1 – 10.000 Elemente inner-halb des Arrays	Siehe Beispiel – JSON

Tabelle 18: Beispiel JSON Ausgabenachweise


```
{
  "issuanceRecords":[
    {
      "rics":"5143",
      "ticketId":"A0815BF0",
      "validFrom":"2025-02-01T00:00:00+01:00",
      "validTo":"2025-03-01T03:00:00+01:00",
      "productId":9999,
      "productOwner":"3634",
      "tariffDescription":"Deutschlandticket",
      "issuedAt":"2025-01-25T02:00:00+01:00",
      "keyId":"31A33",
      "securityProviderRics":"3634",
      "postcode":"49661",
      "issuerAuthorizationId":15312312,
      "issuerKvpId":6321,
      "uniqueOrderId":"6e25585b-2e7e-4ea1-99c3-a5f743693f21"
    },
    {
      "rics":"5143",
      "ticketId":"B33041F0",
      "validFrom":"2025-02-01T00:00:00+01:00",
      "validTo":"2025-03-01T03:00:00+01:00",
      "productId":9999,
      "productOwner":"3634",
      "tariffDescription":"Deutschlandticket",
      "issuedAt":"2025-01-25T02:00:00+01:00",
      "keyId":"31A33",
      "securityProviderRics":"3634",
```

```
      "postcode":"49681",
      "issuerAuthorizationId":15312313,
      "issuerKvpId":6321,
      "uniqueOrderId":"fe97b98f-b3c6-40bb-acef-d78a72297dce"
    }
  ]
}
```

7.2.7 Deutschlandticket

Das Schema „Deutschlandticket“ beschreibt die Daten, die für die Ausstellung eines Deutschlandtickets an das UIC-Sicherheitsportal übermittelt werden müssen. Dieses Schema wird verwendet, wenn ein Deutschlandticket über die API des UIC-Sicherheitsportals erstellt wird.

Fachlich werden hier alle für die Barcodeerzeugung erforderlichen Informationen zur Fahrtberechtigung, zum Fahrgast sowie zum Gültigkeitszeitraum übergeben. Dazu gehören insbesondere personenbezogene Daten wie Name und Geburtsdatum, die bei personengebundenen Tickets im Kontrollprozess eine zentrale Rolle spielen. Die übermittelten Daten bilden die Grundlage für die Erstellung des Barcodes sowie für die spätere Prüfung im Rahmen der Kontrolle. Insbesondere bei der Validierung wird überprüft, ob die im Barcode hinterlegten Informationen mit den vorliegenden Angaben übereinstimmen.

Das Schema stellt damit sicher, dass Deutschlandtickets systemweit einheitlich erzeugt und im Kontrollprozess korrekt interpretiert werden können.

Tabelle 19: Datenschema Deutschlandticket

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
firstName				Ja		
lastName				Ja		
dateOfBirth				Ja		
gender				Nein		
productId				Ja		
productOwner				Ja		
priceInCents				Ja		
monthOfValidity				Ja		
postcode	-	Postleitzahl des Ticketinhabers	string	**Nein**	Min.: 5 Zeichen Max.: 5 Zeichen	"49661"
uniqueOrderId				Ja		
ticketReference				Nein		

Tabelle 20: Beispiel JSON Deutschlandticket

```
{
  "firstName":"Maxima",
  "lastName":"Musterfrau",
  "dateOfBirth":"1990-05-30",
  "productId":9999,
  "productOwner":"3634",
  "priceInCents":6300,
  "monthOfValidity":"2026-06-01",
  "uniqueOrderId":"28d07a4f-02ee-4c6f-996d-89e47670157b",
  "ticketReference":"5GFHt5Q9TXhvvWWH"
}
```

Zusätzliche Erläuterungen

Hinweis zum **Erstellungszeitraum**: Deutschlandtickets können bis zu 6 Monate im Voraus erstellt werden.

Für das Feld „**monthOfValidity**“ gilt folgende fachliche Regel:

- Maßgeblich ist ausschließlich der angegebene Gültigkeitsmonat. Eine etwaig übermittelte Tagesangabe wird bei der Verarbeitung nicht berücksichtigt.
- Das Feld dient somit der monatlichen Zuordnung des Deutschlandtickets und nicht der Angabe eines konkreten Kalendertags innerhalb dieses Monats.

Für das Feld „**postcode**“ gilt:

- Feld kann leer bleiben, da angedachter Verwendungszweck obsolet ist.
- Diese Angabe wird nicht in den Barcode geschrieben.

Für das Feld „**gender**“ gilt:

- Feld kann leer bleiben, da der Defaultwert für die Barcodeerzeugung ausreicht.

7.2.8 D-Tarif-Ticket

Das Schema „D-Tarif-Ticket“ beschreibt die Daten, die für die Ausstellung eines Tickets im Deutschlandtarif an das UIC-Sicherheitsportal übermittelt werden. Dieses Schema wird verwendet, wenn ein Barcode für ein Ticket des Deutschlandtarifs erstellt wird, beispielsweise für relationsbasierte oder relationslose Fahrkarten.

Fachlich werden hierbei sowohl die Produktinformationen als auch die Angaben zum Fahrgast übergeben. Im Gegensatz zum Deutschlandticket, das als standardisiertes Abo-Produkt ausgestaltet ist, können D-Tarif-Tickets unterschiedliche Ausprägungen haben, etwa hinsichtlich Strecke, Gültigkeitszeitraum oder Tarifprodukt. Die übermittelten Daten definieren die konkrete Ausgestaltung des Tickets und bilden die Grundlage für dessen spätere Nutzung und Kontrolle. Dabei ist insbesondere relevant, dass die tariflichen Informationen eindeutig und vollständig angegeben werden, um eine korrekte Interpretation im Kontrollprozess zu ermöglichen.

Das Schema ermöglicht somit die flexible Abbildung verschiedener Angebote des Deutschlandtarifs bei gleichzeitig einheitlicher Verarbeitung im UIC-Sicherheitsportal.

Tabelle 21: Datenschema D-Tarif-Ticket

Feld	Feld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
OpenTicket	-	Angaben zum Produkt	7.2.9	Ja		Siehe Beispiel – JSON
traveler	-	Angaben zum Fahrgast	7.2.10	Ja		Siehe Beispiel – JSON
uniqueOrderId				Ja		
ticketReference				Nein		

Tabelle 22: Beispiel JSON D-Tarif-Ticket


```
{
  "openTicket": {
    "classCode":"first",
    "price":1800,
    "productId":12345,
    “returnIncluded”:false
    "tariffDescription":"Einzelfahrschein",
    "tariffs":[ { "numberOfPassengers":1, "passengerType":0, "reductionCard":"BC25" } ],
    "validFrom":"2026-03-12T00:00:00.000Z",
    "validUntil":"2026-03-12T23:59:59.999Z",
    "currency":"EUR",
    "carrier":["1080", "3122"],
    "isSecurePaperTicket":false,
    "validRegion":[ { "nutsCode":["de"] } ],
    "fromStation":"de:02000:10950",
    "fromStationName":"Hamburg Hbf",
```

```
    "toStation":"de:03404:71612",
    "toStationName":"Osnabrück Hbf",
    "validRegionDescription":"Gültig für Strecke: Hamburg Hbf - Osnabrück Hbf",
……"includedAddOns“:{ "zoneId":["1"] }

},
  "traveler": {
    "firstName":"Maxima",
    "lastName":"Musterfrau",
    "passengerType":0,
    "dateOfBirth":"1990-05-30",
    "gender":1,
    "secondName":"Susanne",
    "title":null
  },
  "uniqueOrderId":"28d07a4f-02ee-4c6f-996d-89e47670157b",
  "ticketReference":"5GFHt5Q9TXhvvWWH"
}
```

7.2.9 D-Tarif-OpenTicket

Das Schema „D-Tarif-OpenTicket“ beschreibt die tariflichen Eigenschaften eines Tickets im Deutschlandtarif. Dieses Schema wird verwendet, um die konkreten Rahmenbedingungen eines Tickets im Zusammenhang mit der Ausstellung eines D-Tarif-Tickets festzulegen. Fachlich werden hier alle Informationen zusammengefasst, die das eigentliche Produkt definieren. Dazu gehören beispielsweise die Gültigkeit, die tarifliche Einordnung, enthaltene Beförderer sowie gegebenenfalls eine konkrete Relation zwischen Start- und Zielpunkt. Die Angaben bestimmen, unter welchen Bedingungen ein Ticket genutzt werden darf und bilden damit die Grundlage für die fachliche Bewertung im Kontrollprozess.

Das Schema ermöglicht es, unterschiedliche Tarifausprägungen strukturiert abzubilden und sorgt dafür, dass Tickets im Deutschlandtarif eindeutig interpretiert werden können.

Tabelle 23: Datenschema D-Tarif-OpenTicket

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
classCode				Ja		
price				Ja		
productId				Ja		
returnIncluded				Ja		
tariffDescription				Ja		
tariffs	-	Tarifkonfiguration im Ticket	Array Typ: 7.2.11	Ja		Siehe Beispiel – JSON
validFrom				Ja		
validUntil				Ja		
carrier	carrier	Im Ticket enthaltene Beförderer als RICS-/Org. Code (optional), Stammdaten beachten	Array Typ: string	Nein		["1080", "3122"]
isSecurePaperTicket				Nein		
validRegion	validRegion	Gültigkeitsregion des Tickets als Menge von NUTS-Codes	Array Typ: string	Nein		["de"]

fromStation	fromStation	dhid der Starthaltestelle	string	Nein		"de:02000:10950"
fromStationName	fromStationName	Name der Starthaltestelle	UTF8string	Nein		"Hamburg Hbf"
toStation	toStation	dhid der Zielhaltestelle	string	Nein		"de:03404:71612"
toStationName	toStationName	Name der Zielhaltestelle	UTF8string	Nein		"Osnabrück Hbf"
validRegionDescription	validRegionDesc	Menschenlesbare Gültigkeitsregion des Tickets	string	Nein		„ohne Umweg“, "Gültig für Strecke: Hamburg Hbf - Osnabrück Hbf"
includedAddOns	includedAddOns	Gültigkeitsregion nach VDV, steuert die Erzeugung der includedAddOns im Barcode	Array Typ: string	Nein		["1"]

Tabelle 24: Beispiel JSON D-Tarif-OpenTicket


```
{
  "classCode":"first",
  "price":1800,
  "productId":12345,
  “returnIncluded”:false
  "tariffDescription":"Einzelfahrschein",
  "tariffs":[ { "numberOfPassengers":1, "passengerType":0, "reductionCard":"BC25" } ],
  "validFrom":"2026-03-12T00:00:00.000Z",
  "validUntil":"2026-03-12T23:59:59.999Z",
  "currency":"EUR",
  "carrier":["1080", "3122"],
  "isSecurePaperTicket":false,
  "validRegion":[ { "nutsCode":"de" } ],
  "fromStation":"de:02000:10950",
  "fromStationName":"Hamburg Hbf",
  "toStation":"de:03404:71612",
  "toStationName":"Osnabrück Hbf",
```

"validRegionDescription":"Gültig für Strecke: Hamburg Hbf - Osnabrück Hbf",
…"includedAddOns“:{ "zoneId":["1"] }
}

Zusätzliche Erläuterungen

Für die Felder „**fromStation**“ und „**toStation“** gelten folgende fachliche und technische Einschränkungen:

- Haltestellen werden im Rahmen der Streckenbeschreibung über standardisierte Identifikatoren referenziert, wie beispielsweise die deutschlandweit verwendete DHID (Deutschlandweite Haltestellen-ID).
- Diese Kennungen ermöglichen eine eindeutige Zuordnung von Start- und Zielpunkten und bilden die Grundlage für die fachliche Interpretation von relationsbasierten Tickets.

Für das Feld „**validRegionDescription“** gelten folgende fachliche und technische Einschränkungen:

- Es dient der menschenlesbaren Beschreibung des Gültigkeitsbereichs eines Tickets.
- Die fachliche Bewertung erfolgt ausschließlich auf Basis der strukturierten Daten (z. B. Haltestellen-IDs). Die Beschreibung dient ausschließlich der Anzeige und hat keinen Einfluss auf die Validierung.

7.2.10 D-Tarif-Fahrgast

Das Schema „D-Tarif-Fahrgast“ beschreibt die personenbezogenen Angaben zu einem Ticket im Deutschlandtarif. Dieses Schema wird im Rahmen der Ticketausstellung verwendet und ist Bestandteil des D-Tarif-Tickets. Es wird immer dann übermittelt, wenn ein Ticket für einen konkreten Fahrgast erzeugt wird.

Fachlich werden hier die Daten erfasst, die zur Identifikation des Fahrgasts und zur tariflichen Einordnung erforderlich sind. Dazu gehören insbesondere Name, Geburtsdatum sowie der Fahrgasttyp, beispielsweise Erwachsener, Kind oder weitere tarifliche Kategorien. Diese Angaben sind insbesondere bei personengebundenen Tickets von Bedeutung, da sie im Kontrollprozess zur Überprüfung der Berechtigung herangezogen werden. Die Daten werden im Ticket hinterlegt und können bei der Kontrolle mit einem Ausweisdokument abgeglichen werden.

Das Schema ist somit ein Bestandteil der Ticketdefinition und wird bei der Erstellung eines D-Tarif-Tickets gemeinsam mit den Produktdaten übermittelt.

Tabelle 25: Datenschema D-Tarif-Fahrgast

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
firstName				Ja		
lastName				Ja		
passengerType				Ja		
dateOfBirth				Nein		
gender				**Nein**		
secondName	secondName	Optionaler zweiter Name des Ticketinhabers	string	Nein		null, "Susanne"
title	title	Titel des Ticketinhabers	string	Nein	Max.: 3 Zeichen	null, "Dr."

Tabelle 26: Beispiel JSON D-Tarif-Fahrgast

```
{
  "firstName":"Maxima",
  "lastName":"Musterfrau",
  "passengerType":0,
  "dateOfBirth":"1990-05-30",
  "gender":1,
  "secondName":"Susanne",
  "title":null
}
```

7.2.11 D-Tarif-Tariftyp

Das Schema „D-Tarif-Tariftyp“ beschreibt die tarifliche Zusammensetzung eines Tickets im Deutschlandtarif. Dieses Schema wird im Rahmen der Ticketausstellung als Bestandteil des D-Tarif-Produkts verwendet. Es kommt immer dann zum Einsatz, wenn ein Ticket mehrere Fahrgäste oder unterschiedliche tarifliche Ausprägungen abbildet.

Fachlich wird hier festgelegt, wie sich das Ticket zusammensetzt, beispielsweise in Bezug auf die Anzahl der Fahrgäste, deren Fahrgasttypen und mögliche Ermäßigungen.

Diese Informationen sind insbesondere für die korrekte Preisbildung sowie für die Interpretation des Tickets im Kontrollprozess relevant. Sie geben Aufschluss darüber, für wen und unter welchen Bedingungen das Ticket gültig ist.

Das Schema ermöglicht somit die strukturierte Abbildung auch komplexerer tariflicher Konstellationen innerhalb eines Tickets.

Tabelle 27: Datenschema D-Tarif-Tariftyp

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
numberOfPassengers				Ja		
passengerType				Ja		
reductionCard				Nein		

Tabelle 28: Beispiel JSON D-Tarif-Tariftyp

```
{
  "numberOfPassengers":1,
  "passengerType":0,
  "reductionCard":"BC25"
}
```

7.2.12 KVV-Ticket

Das Schema „KVV-Ticket“ beschreibt die Daten, die für die Ausstellung eines Tickets im KVV-Tarif an das UIC-Sicherheitsportal übermittelt werden. Dieses Schema wird verwendet, wenn ein Barcode für ein Ticket des KVV-Tarifs erstellt wird.

Fachlich werden hierbei sowohl die Produktinformationen als auch die Angaben zum Fahrgast übergeben. Wie auch im D-TARIF, können KVV-Tickets unterschiedliche Ausprägungen haben, etwa hinsichtlich Strecke, Gültigkeitszeitraum oder Tarifprodukt. Die übermittelten Daten definieren die konkrete Ausgestaltung des Tickets und bilden die Grundlage für dessen spätere Nutzung und Kontrolle. Dabei ist insbesondere relevant, dass die tariflichen Informationen eindeutig und vollständig angegeben werden, um eine korrekte Interpretation im Kontrollprozess zu ermöglichen.

Das Schema ermöglicht somit die flexible Abbildung verschiedener Angebote des KVV-Tarifs bei gleichzeitig einheitlicher Verarbeitung im UIC-Sicherheitsportal.

Tabelle 29: Datenschema KVV-Ticket

Feld	Feld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
OpenTicket	-	Angaben zum Produkt	7.2.13	Ja		Siehe Beispiel – JSON
traveler	-	Angaben zum Fahrgast	7.2.14	Ja		Siehe Beispiel – JSON
UniqueOrderId				Ja		
ticketReference				Ja		

Tabelle 30: Beispiel JSON KVV-Ticket


```
{
  "openTicket": {
    "classCode":"second",
    "price":500,
    "productId":12345,
    "returnIncluded":false
    "tariffId":"Preisstufe 5"
    "tariffDescription":"Tageskarte Gruppe",
    "tariffs":[ { "numberOfPassengers":3, "passengerType":0, "reductionCard":"BC25" } ],
    "validFrom":"2026-03-12T00:00:00.000Z",
    "validUntil":"2026-03-12T23:59:59.999Z",
    "currency":"EUR",
    "isSecurePaperTicket":false,
    "validRegion":[ { "zoneID":["100","235"] } ],
    "fromStation":"de:08212:90",
    "fromStationName":"Karlsruhe Hbf",
    "toStation":"de:08215:1856",
    "toStationName":"Bruchsal",
```

```
    "validRegionDescription":"über Karlsruhe-Durlach",
},
  "traveler": {
    "firstName":"Maxima",
    "lastName":"Musterfrau",
    "passengerType":0,
    "dateOfBirth":"1990-05-30",
  },
  "uniqueOrderId":"28d07a4f-02ee-4c6f-996d-89e47670157b",
  "ticketReference":"5GFHt5Q9TXhvvWWH"
}
```

7.2.13 KVV-OpenTicket

Das Schema „KVV-OpenTicket“ beschreibt die tariflichen Eigenschaften eines Tickets im KVV-Tarif. Dieses Schema wird verwendet, um die konkreten Rahmenbedingungen eines Tickets im Zusammenhang mit der Ausstellung eines KVV-Tickets festzulegen. Fachlich werden hier alle Informationen zusammengefasst, die das eigentliche Produkt definieren. Dazu gehören beispielsweise die Gültigkeit, die tarifliche Einordnung sowie gegebenenfalls eine konkrete Relation zwischen Start- und Zielpunkt. Die Angaben bestimmen, unter welchen Bedingungen ein Ticket genutzt werden darf und bilden damit die Grundlage für die fachliche Bewertung im Kontrollprozess.

Das Schema ermöglicht es, unterschiedliche Tarifausprägungen strukturiert abzubilden und sorgt dafür, dass Tickets im KVV-Tarif eindeutig interpretiert werden können.

Tabelle 31: Datenschema KVV-OpenTicket

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
classCode				Ja		
price				Ja		
productId				Ja		
returnIncluded				Ja		
tariffDescription				Ja		
tariffId				Nein		
tariffs	-	Tariftypen des Tickets	Array Typ: 7.2.15	Ja		Siehe Beispiel – JSON
validFrom				Ja		
validUntil				Ja		
currency				Nein		
isSecurePaperTicket				Nein		
validRegion	validRegion	Gültigkeitsregion des Tickets als Menge von ZonenIds	Array Typ: string	Nein		["100","235"]
fromStation	fromStation	dhid der Starthaltestelle	string	Nein		"de:08212:90"

fromStationName	fromStation-Name	Name der Starthaltestelle + Wabe	UTF8string	Nein		"Karlsruhe Hbf (100)"
toStation	toStation	dhid der Zielhaltestelle	string	Nein		"de:08215:1856"
toStation-Name	toStation-Name	Name der Zielhaltestelle + Wabe	UTF8string	Nein		"Bruchsal (236)"
validRegion-Description	validRegion-Desc	Menschenlesbare Gültigkeitsregion des Tickets	string	Nein		„über Karlsruhe-Durlach“

Tabelle 32: Beispiel JSON KVV-OpenTicket


```
{
    "classCode":"first",
    "price":500,
    "productId":12345,
    "returnIncluded":false
    "tariffId":"Preisstufe 5"
    "tariffDescription":"Tageskarte Gruppe",
    "tariffs":[ { "numberOfPassengers":3, "passengerType":0, "reductionCard":"BC25" } ],
    "validFrom":"2026-03-12T00:00:00.000Z",
    "validUntil":"2026-03-12T23:59:59.999Z",
    "currency":"EUR",
    "isSecurePaperTicket":false,
    "validRegion":[ { "zoneID":["100","235"] } ],
    "fromStation":"de:08212:90",
    "fromStationName":"Karlsruhe Hbf (100)",
    "toStation":"de:08215:1856",
    "toStationName":"Bruchsal (236)",
    "validRegionDescription":"über Karlsruhe-Durlach",
},
```

Zusätzliche Erläuterungen

Für die Felder „**fromStation**“ und „**toStation“** gelten folgende fachliche und technische Einschränkungen:

- Haltestellen werden im Rahmen der Streckenbeschreibung über standardisierte Identifikatoren referenziert, wie beispielsweise die deutschlandweit verwendete DHID (Deutschlandweite Haltestellen-ID).
- Diese Kennungen ermöglichen eine eindeutige Zuordnung von Start- und Zielpunkten und bilden die Grundlage für die fachliche Interpretation von relationsbasierten Tickets.

Für das Feld „**validRegionDescription“** gelten folgende fachliche und technische Einschränkungen:

- Es dient der menschenlesbaren Beschreibung des Gültigkeitsbereichs eines Tickets.
- Die fachliche Bewertung erfolgt ausschließlich auf Basis der strukturierten Daten (z. B. Haltestellen-IDs). Die Beschreibung dient ausschließlich der Anzeige und hat keinen Einfluss auf die technische Validierung.

7.2.14 KVV-Fahrgast

Das Schema „KVV-Fahrgast“ beschreibt die personenbezogenen Angaben zu einem Ticket im KVV-Tarif. Dieses Schema wird im Rahmen der Ticketausstellung verwendet und ist Bestandteil des KVV-Tickets. Es wird immer dann übermittelt, wenn ein Ticket für einen konkreten Fahrgast erzeugt wird.

Fachlich werden hier die Daten erfasst, die zur Identifikation des Fahrgasts und zur tariflichen Einordnung erforderlich sind. Dazu gehören insbesondere Name, Geburtsdatum sowie der Fahrgasttyp, beispielsweise Erwachsener, Kind oder weitere tarifliche Kategorien. Diese Angaben sind insbesondere bei personengebundenen Tickets von Bedeutung, da sie im Kontrollprozess zur Überprüfung der Berechtigung herangezogen werden. Die Daten werden im Ticket hinterlegt und können bei der Kontrolle mit einem Ausweisdokument abgeglichen werden.

Das Schema ist somit ein Bestandteil der Ticketdefinition und wird bei der Erstellung eines KVV-Tickets gemeinsam mit den Produktdaten übermittelt.

Tabelle 33: Datenschema KVV-Fahrgast

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
firstName				Ja		
lastName				Ja		
passengerType				Ja		
dateOfBirth				Nein		

Tabelle 34: Beispiel JSON KVV-Fahrgast

```
{
  "firstName":"Maxima",
  "lastName":"Musterfrau",
  "passengerType":0,
  "dateOfBirth":"1990-05-30",
}
```

7.2.15 KVV-Tariftyp

Das Schema „KVV-Tariftyp“ beschreibt die tarifliche Zusammensetzung eines Tickets im KVV-Tarif. Dieses Schema wird im Rahmen der Ticketausstellung als Bestandteil des KVV-Produkts verwendet. Es kommt immer dann zum Einsatz, wenn ein Ticket mehrere Fahrgäste oder unterschiedliche tarifliche Ausprägungen abbildet.

Fachlich wird hier festgelegt, wie sich das Ticket zusammensetzt, beispielsweise in Bezug auf die Anzahl der Fahrgäste, deren Fahrgasttypen und mögliche Ermäßigungen.

Diese Informationen sind insbesondere für die korrekte Preisbildung sowie für die Interpretation des Tickets im Kontrollprozess relevant. Sie geben Aufschluss darüber, für wen und unter welchen Bedingungen das Ticket gültig ist.

Das Schema ermöglicht somit die strukturierte Abbildung auch komplexerer tariflicher Konstellationen innerhalb eines Tickets.

Tabelle 35: Datenschema KVV-Tariftyp

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
numberOfPassengers				Ja		
passengerType				Ja		
reductionCard				Nein		

Tabelle 36: Beispiel JSON KVV-Tariftyp

```
{
  "numberOfPassengers":1,
  "passengerType":0,
  "reductionCard":"BC25"
}
```

7.2.16 NOX-Ticket

Das Schema „NOX-Ticket“ beschreibt die Daten, die für die Ausstellung eines Ticketbarcodes im Haustarif von NOX mobility an das UIC-Sicherheitsportal übermittelt werden.

Tabelle 37: Datenschema NOX-Ticket

Feld	Feld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
OpenTicket	-	Angaben zum Produkt	7.2.9	Ja		Siehe Beispiel – JSON
traveler	-	Angaben zum Fahrgast	7.2.18	Ja		Siehe Beispiel – JSON
uniqueOrderId				Ja		
ticketReference				Nein		

Tabelle 38: Beispiel JSON NOX-Ticket


```
{
  "openTicket": {
    "classCode":"second",
    "price":12300,
    "productId":12345,
    “returnIncluded”:false
    "tariffDescription":"Einzelfahrt",
    "tariffs":[ { "numberOfPassengers":1, "passengerType":0} ],
    "validFrom":"2026-03-12T21:00:00.000Z",
    "validUntil":"2026-03-13T09:59:59.999Z",
    "currency":"EUR",
    "fromStation":"de:02000:10950",
    "fromStationName":"Hamburg Hbf",
    "toStation":" de:09162:100",
    "toStationName":"München Hbf"
},
  "traveler": {
    "firstName":"Maxima",
    "lastName":"Musterfrau",
    "passengerType":0,
    "dateOfBirth":"1990-05-30",
  },
  "uniqueOrderId":"28d07a4f-02ee-4c6f-996d-89e47670157b",
  "ticketReference":"5GFHt5Q9TXhvvWWH"
}
```

7.2.17 NOX-OpenTicket

Das Schema „NOX-OpenTicket“ beschreibt die tariflichen Eigenschaften eines Tickets im Haustarif von NOX mobility. Fachlich werden hier alle Informationen zusammengefasst, die das eigentliche Produkt definieren. Dazu gehören beispielsweise die Gültigkeit, die tarifliche Einordnung, enthaltene Beförderer sowie die konkrete Relation zwischen Start- und Zielpunkt. Die Angaben bestimmen, unter welchen Bedingungen ein Ticket genutzt werden darf und bilden damit die Grundlage für die fachliche Bewertung im Kontrollprozess.

Tabelle 39: Datenschema NOX-OpenTicket

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
classCode				Ja		
price				Ja		
productId				Ja		
tariffDescription				Ja		
tariffs	-	Tarifkonfiguration im Ticket	Array Typ: 7.2.19	Ja		Siehe Beispiel – JSON
validFrom				Ja		
validUntil				Ja		
isSecurePaperTicket				Nein		
fromStation	fromStation	dhid der Starthaltestelle	string	Nein		"de:02000:10950"
fromStationName	fromStationName	Name der Starthaltestelle	UTF8string	Nein		"Hamburg Hbf"
toStation	toStation	dhid der Zielhaltestelle	string	Nein		" de:09162:100"
toStationName	toStationName	Name der Zielhaltestelle	UTF8string	Nein		"München Hbf"

Tabelle 40: Beispiel JSON NOX-OpenTicket

```
{
  "classCode":"second",
  "price":12300,
  "productId":12345,
  “returnIncluded”:false
  "tariffDescription":"Einzelfahrt",
```

```
  "tariffs":[ { "numberOfPassengers":1, "passengerType":0} ],
  "validFrom":"2026-03-12T21:00:00.000Z",
  "validUntil":"2026-03-13T09:59:59.999Z",
  "currency":"EUR",
  "fromStation":"de:02000:10950",
  "fromStationName":"Hamburg Hbf",
  "toStation":" de:09162:100",
  "toStationName":"München Hbf"
}
```


Zusätzliche Erläuterungen

Für die Felder „**fromStation**“ und „**toStation“** gelten folgende fachliche und technische Einschränkungen:

- Haltestellen werden im Rahmen der Streckenbeschreibung über standardisierte Identifikatoren referenziert, wie beispielsweise die deutschlandweit verwendete DHID (Deutschlandweite Haltestellen-ID).
- Diese Kennungen ermöglichen eine eindeutige Zuordnung von Start- und Zielpunkten und bilden die Grundlage für die fachliche Interpretation von relationsbasierten Tickets.

7.2.18 NOX-Fahrgast

Das Schema „NOX-Fahrgast“ beschreibt die personenbezogenen Angaben zu einem Ticket von NOX mobility. Dieses Schema wird im Rahmen der Ticketausstellung verwendet und ist Bestandteil des NOX-Tickets.

Fachlich werden hier die Daten erfasst, die zur Identifikation des Fahrgasts und zur tariflichen Einordnung erforderlich sind. Dazu gehören insbesondere Name, Geburtsdatum sowie der Fahrgasttyp, beispielsweise Erwachsener, Kind oder weitere tarifliche Kategorien. Die Daten werden im Ticket hinterlegt und können bei der Kontrolle mit einem Ausweisdokument abgeglichen werden.

Das Schema ist somit ein Bestandteil der Ticketdefinition und wird bei der Erstellung eines D-Tarif-Tickets gemeinsam mit den Produktdaten übermittelt.

Tabelle 41: Datenschema NOX-Fahrgast

Feld	Datenfeld im Barcode	Beschreibung	Daten-typ	Pflicht-feld	Wertebe-reich	Beispiel
firstName				Ja		
lastName				Ja		
passenger-Type				Ja		
dateOfBirth				Nein		
gender				**Nein**		
secondName	secondName	Optionaler zweiter Name des Ticketinhabers	string	Nein		null, "Susanne"

Tabelle 42: Beispiel JSON NOX-Fahrgast

```
{
  "firstName":"Maxima",
  "lastName":"Musterfrau",
  "passengerType":0,
  "dateOfBirth":"1990-05-30",
  "gender":1,
  "secondName":"Susanne"
}
```

7.2.19 NOX-Tariftyp

Das Schema „NOX-Tariftyp“ beschreibt die tarifliche Zusammensetzung eines Tickets im Tarif von NOX mobility.

Fachlich wird hier festgelegt, wie sich das Ticket in Bezug auf die Anzahl der Fahrgäste, deren Fahrgasttypen und mögliche Ermäßigungen zusammensetzt.

Diese Informationen sind insbesondere für die korrekte Preisbildung sowie für die Interpretation des Tickets im Kontrollprozess relevant. Sie geben Aufschluss darüber, für wen und unter welchen Bedingungen das Ticket gültig ist.

Tabelle 43: Datenschema NOX-Tariftyp

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
numberOfPassengers				Ja		
passengerType				Ja		

Tabelle 44: Beispiel JSON NOX-Tariftyp

```
{
  "numberOfPassengers":1,
  "passengerType":0
}
```

7.2.20 NDS-Ticket

Das Schema „NDS-Ticket“ beschreibt die Daten, die für die Ausstellung eines Tickets im Niedersachsen-Tarif an das UIC-Sicherheitsportal übermittelt werden. Dieses Schema wird verwendet, wenn ein Barcode für ein Ticket des Niedersachsen-Tarifs erstellt wird.

Fachlich werden hierbei sowohl die Produktinformationen als auch die Angaben zum Fahrgast übergeben. Wie auch im D-TARIF, können Niedersachsentarif-Tickets unterschiedliche Ausprägungen haben, etwa hinsichtlich Strecke, Gültigkeitszeitraum oder Tarifprodukt. Die übermittelten Daten definieren die konkrete Ausgestaltung des Tickets und bilden die Grundlage für dessen spätere Nutzung und Kontrolle. Dabei ist insbesondere relevant, dass die tariflichen Informationen eindeutig und vollständig angegeben werden, um eine korrekte Interpretation im Kontrollprozess zu ermöglichen.

Das Schema ermöglicht somit die flexible Abbildung verschiedener Angebote des Niedersachsen-Tarifs bei gleichzeitig einheitlicher Verarbeitung im UIC-Sicherheitsportal.

Tabelle 45: Datenschema NDS-Ticket

Feld	Feld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
OpenTicket	-	Angaben zum Produkt	7.2.21	Ja		Siehe Beispiel – JSON
traveler	-	Angaben zum Fahrgast	7.2.22	Ja		Siehe Beispiel – JSON
UniqueOrderId				Ja		
ticketReference				Ja		

Tabelle 46: Beispiel JSON NDS-Ticket


```
{
  "openTicket": {
    "classCode":"second",
    "price":6180,
    "productId":55021000,
    "returnIncluded":false
    "tariffId":" inkl. Kostenlose Mitnahme am WE"
    "tariffDescription":" SCHÜLER Abo-Monatskarte 2. Klasse",
    "tariffs":[ { "numberOfPassengers":1, "passengerType":3, "reductionCard":"" } ],
    "validFrom":"2026-03-12T00:00:00.000Z",
    "validUntil":"2026-03-12T23:59:59.999Z",
    "currency":"EUR",
    "isSecurePaperTicket":false,
    "validRegion":[ { "nutsCode":["DE9"] } ],
    "fromStation":"33582884",
    "fromStationName":"Hodenhagen",
```

```
    "toStation":" 33586193",
    "toStationName":"Walsrode",
    "validRegionDescription":"direkter Weg",
},
  "traveler": {
    "firstName":"Maxima",
    "lastName":"Musterfrau",
    "passengerType":0,
    "dateOfBirth":"1990-05-30",
  },
  "uniqueOrderId":"28d07a4f-02ee-4c6f-996d-89e47670157b",
  "ticketReference":"5GFHt5Q9TXhvvWWH"
}
```

7.2.21 NDS-OpenTicket

Das Schema „NDS-OpenTicket“ beschreibt die tariflichen Eigenschaften eines Tickets im Niedersachsen-Tarif. Dieses Schema wird verwendet, um die konkreten Rahmenbedingungen eines Tickets im Zusammenhang mit der Ausstellung eines Niedersachsentarif-Tickets festzulegen. Fachlich werden hier alle Informationen zusammengefasst, die das eigentliche Produkt definieren. Dazu gehören beispielsweise die Gültigkeit, die tarifliche Einordnung sowie gegebenenfalls eine konkrete Relation zwischen Start- und Zielpunkt. Die Angaben bestimmen, unter welchen Bedingungen ein Ticket genutzt werden darf und bilden damit die Grundlage für die fachliche Bewertung im Kontrollprozess.

Das Schema ermöglicht es, unterschiedliche Tarifausprägungen strukturiert abzubilden und sorgt dafür, dass Tickets im Niedersachsen-Tarif eindeutig interpretiert werden können.

Tabelle 47: Datenschema NDS-OpenTicket

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
classCode				Ja		
price				Ja		
productId				Ja		
returnIncluded				Ja		
tariffDescription				Ja		
tariffId				Nein		
tariffs	-	Tariftypen des Tickets	Array Typ: 7.2.23	Ja		Siehe Beispiel – JSON
validFrom				Ja		
validUntil				Ja		
currency				Nein		
isSecurePaperTicket				Nein		
validRegion	validRegion	Gültigkeitsregion des Tickets als Menge von NutsCodes	Array Typ: string	Nein		["DE9"]
fromStation	fromStation	dhid der Starthaltestelle	string	Nein		"33582884"

fromStationName	fromStationName	Name der Starthaltestelle + Wabe	UTF8string	Nein		"Hodenhagen"
toStation	toStation	dhid der Zielhaltestelle	string	Nein		"33586193"
toStationName	toStationName	Name der Zielhaltestelle + Wabe	UTF8string	Nein		"Walsrode"
validRegionDescription	validRegionDesc	Menschenlesbare Gültigkeitsregion des Tickets	string	Nein		„direkter Weg“

Tabelle 48: Beispiel JSON NDS-OpenTicket


```
{
    "classCode":"second",
    "price":6180,
    "productId":55021000,
    "returnIncluded":false
    "tariffId":" inkl. Kostenlose Mitnahme am WE"
    "tariffDescription":" SCHÜLER Abo-Monatskarte 2. Klasse",
    "tariffs":[ { "numberOfPassengers":1, "passengerType":3, "reductionCard":"" } ],
    "validFrom":"2026-03-12T00:00:00.000Z",
    "validUntil":"2026-03-12T23:59:59.999Z",
    "currency":"EUR",
    "isSecurePaperTicket":false,
    "validRegion":[ { "nutsCode":["DE9"] } ],
    "fromStation":"33582884",
    "fromStationName":"Hodenhagen",
    "toStation":" 33586193",
    "toStationName":"Walsrode",
    "validRegionDescription":"direkter Weg",
},
```

Zusätzliche Erläuterungen

Für die Felder „**fromStation**“ und „**toStation“** gelten folgende fachliche und technische Einschränkungen:

- Haltestellen werden im Rahmen der Streckenbeschreibung über standardisierte Identifikatoren referenziert. Im Niedersachsentarif wird hierfür die NITAG-ID verwendet.
- Diese Kennungen ermöglichen eine eindeutige Zuordnung von Start- und Zielpunkten und bilden die Grundlage für die fachliche Interpretation von relationsbasierten Tickets.

Für das Feld „**validRegionDescription“** gelten folgende fachliche und technische Einschränkungen:

- Es dient der menschenlesbaren Beschreibung des Gültigkeitsbereichs eines Tickets.
- Die fachliche Bewertung erfolgt ausschließlich auf Basis der strukturierten Daten (z. B. Haltestellen-IDs). Die Beschreibung dient ausschließlich der Anzeige und hat keinen Einfluss auf die technische Validierung.

7.2.22 NDS-Fahrgast

Das Schema „NDS-Fahrgast“ beschreibt die personenbezogenen Angaben zu einem Ticket im Niedersachsen-Tarif. Dieses Schema wird im Rahmen der Ticketausstellung verwendet und ist Bestandteil des Niedersachsentarif-Tickets. Es wird immer dann übermittelt, wenn ein Ticket für einen konkreten Fahrgast erzeugt wird.

Fachlich werden hier die Daten erfasst, die zur Identifikation des Fahrgasts und zur tariflichen Einordnung erforderlich sind. Dazu gehören insbesondere Name, Geburtsdatum sowie der Fahrgasttyp, beispielsweise Erwachsener, Kind oder weitere tarifliche Kategorien. Diese Angaben sind insbesondere bei personengebundenen Tickets von Bedeutung, da sie im Kontrollprozess zur Überprüfung der Berechtigung herangezogen werden. Die Daten werden im Ticket hinterlegt und können bei der Kontrolle mit einem Ausweisdokument abgeglichen werden.

Das Schema ist somit ein Bestandteil der Ticketdefinition und wird bei der Erstellung eines Niedersachsentarif-Tickets gemeinsam mit den Produktdaten übermittelt.

Tabelle 49: Datenschema NDS-Fahrgast

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
firstName				Ja		
lastName				Ja		
passengerType				Ja		
dateOfBirth				Nein		

Tabelle 50: Beispiel JSON NDS-Fahrgast

```
{
  "firstName":"Maxima",
  "lastName":"Musterfrau",
  "passengerType":0,
  "dateOfBirth":"1990-05-30",
}
```

7.2.23 NDS-Tariftyp

Das Schema „NDS-Tariftyp“ beschreibt die tarifliche Zusammensetzung eines Tickets im Niedersachsen-Tarif. Dieses Schema wird im Rahmen der Ticketausstellung als Bestandteil des Niedersachsentarif-Produkts verwendet. Es kommt immer dann zum Einsatz, wenn ein Ticket mehrere Fahrgäste oder unterschiedliche tarifliche Ausprägungen abbildet.

Fachlich wird hier festgelegt, wie sich das Ticket zusammensetzt, beispielsweise in Bezug auf die Anzahl der Fahrgäste, deren Fahrgasttypen und mögliche Ermäßigungen.

Diese Informationen sind insbesondere für die korrekte Preisbildung sowie für die Interpretation des Tickets im Kontrollprozess relevant. Sie geben Aufschluss darüber, für wen und unter welchen Bedingungen das Ticket gültig ist.

Das Schema ermöglicht somit die strukturierte Abbildung auch komplexerer tariflicher Konstellationen innerhalb eines Tickets.

Tabelle 51: Datenschema NDS-Tariftyp

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
numberOfPassengers				Ja		
passengerType				Ja		
reductionCard				Nein		

Tabelle 52: Beispiel JSON NDS-Tariftyp

```
{
  "numberOfPassengers":1,
  "passengerType":0,
  "reductionCard":"BC25"
}
```

7.2.24 SHT-Ticket

Das Schema „SHT-Ticket“ beschreibt die Daten, die für die Ausstellung eines Tickets im SH-Tarif an das UIC-Sicherheitsportal übermittelt werden. Dieses Schema wird verwendet, wenn ein Barcode für ein Ticket des SH-Tarifs erstellt wird.

Fachlich werden hierbei sowohl die Produktinformationen als auch die Angaben zum Fahrgast übergeben. Wie auch im D-TARIF, können SHT-Tickets unterschiedliche Ausprägungen haben, etwa hinsichtlich Strecke, Gültigkeitszeitraum oder Tarifprodukt. Die übermittelten Daten definieren die konkrete Ausgestaltung des Tickets und bilden die Grundlage für dessen spätere Nutzung und Kontrolle. Dabei ist insbesondere relevant, dass die tariflichen Informationen eindeutig und vollständig angegeben werden, um eine korrekte Interpretation im Kontrollprozess zu ermöglichen.

Das Schema ermöglicht somit die flexible Abbildung verschiedener Angebote des SH-Tarifs bei gleichzeitig einheitlicher Verarbeitung im UIC-Sicherheitsportal.

Tabelle 53: Datenschema SHT-Ticket

Feld	Feld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
OpenTicket	-	Angaben zum Produkt	7.2.25	Ja		Siehe Beispiel – JSON
traveler	-	Angaben zum Fahrgast	7.2.26	Ja		Siehe Beispiel – JSON
UniqueOrderId				Ja		
ticketReference				Ja		

Tabelle 54: Beispiel JSON SHT-Ticket


```
{
  "openTicket": {
    "classCode":"first",
    "price":500,
    "productOwner":5728,
    "productId":12345,
    "returnIncluded":false
    "tariffId":"Preisstufe R"
    "tariffDescription":"Tageskarte Gruppe",
    "tariffs":[ { "numberOfPassengers":3, "passengerType":0, "reductionCard":"BC25" } ],
    "validFrom":"2026-03-12T00:00:00.000Z",
    "validUntil":"2026-03-12T23:59:59.999Z",
    "currency":"EUR",
    "isSecurePaperTicket":false,
    "validRegion":[ { "nutsCode":["DEF"] } ],
    "fromStation":"de:05334:1008",
```

```
    "fromStationName":" Lübeck Hbf 1234",
    "toStation":"de:14524:41032",
    "toStationName":" Kiel Hbf 5678",
    "validRegionDescription":"direkter Weg",
},
  "traveler": {
    "firstName":"Maxima",
    "lastName":"Musterfrau",
    "passengerType":0,
    "dateOfBirth":"1990-05-30",
  },
  "uniqueOrderId":"28d07a4f-02ee-4c6f-996d-89e47670157b",
  "ticketReference":"5GFHt5Q9TXhvvWWH"
}
```

7.2.25 SHT-OpenTicket

Das Schema „SHT-OpenTicket“ beschreibt die tariflichen Eigenschaften eines Tickets im SH-Tarif. Dieses Schema wird verwendet, um die konkreten Rahmenbedingungen eines Tickets im Zusammenhang mit der Ausstellung eines SHT-Tickets festzulegen. Fachlich werden hier alle Informationen zusammengefasst, die das eigentliche Produkt definieren. Dazu gehören beispielsweise die Gültigkeit, die tarifliche Einordnung sowie gegebenenfalls eine konkrete Relation zwischen Start- und Zielpunkt. Die Angaben bestimmen, unter welchen Bedingungen ein Ticket genutzt werden darf und bilden damit die Grundlage für die fachliche Bewertung im Kontrollprozess.

Das Schema ermöglicht es, unterschiedliche Tarifausprägungen strukturiert abzubilden und sorgt dafür, dass Tickets im SH-Tarif eindeutig interpretiert werden können.

Tabelle 55: Datenschema SHT-OpenTicket

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
classCode				Ja		
price				Ja		
productId				Ja		
productOwner				Ja		
returnIncluded				Ja		
tariffDescription				Ja		
tariffId				Nein		
tariffs	-	Tariftypen des Tickets	Array Typ: 7.2.27	Ja		Siehe Beispiel – JSON
validFrom				Ja		
validUntil				Ja		
currency				Nein		
isSecurePaperTicket				Nein		
validRegion	validRegion	Gültigkeitsregion des Tickets als Menge von NutsCodes	Array Typ: string	Nein		["DEF"]

fromStation	fromStation	dhid der Starthaltestelle	string	Nein		"de:05334:1008"
fromStationName	fromStation-Name	Name der Starthaltestelle + Gemeindecode	UTF8string	Nein		"Lübeck Hbf 1234"
toStation	toStation	dhid der Zielhaltestelle	string	Nein		"de:14524:41032"
toStation-Name	toStation-Name	Name der Zielhaltestelle + Gemeindecode	UTF8string	Nein		"Kiel Hbf 5678"
validRegion-Description	validRegion-Desc	Menschenlesbare Gültigkeitsregion des Tickets	string	Nein		"direkter Weg"

Tabelle 56: Beispiel JSON SHT-OpenTicket


```
{
    "classCode":"first",
    "price":500,
    "productId":12345,
    "productId":5728,
    "returnIncluded":false
    "tariffId":"Preisstufe R"
    "tariffDescription":"Tageskarte Gruppe",
    "tariffs":[ { "numberOfPassengers":3, "passengerType":0, "reductionCard":"BC25" } ],
    "validFrom":"2026-03-12T00:00:00.000Z",
    "validUntil":"2026-03-12T23:59:59.999Z",
    "currency":"EUR",
    "isSecurePaperTicket":false,
    "validRegion":[ { "nutsCode":["DEF"] } ],
    "fromStation":"de:05334:1008",
    "fromStationName":" Lübeck Hbf 1234",
    "toStation":"de:14524:41032",
    "toStationName":" Kiel Hbf 5678",
    "validRegionDescription":"direkter Weg",
},
```

Zusätzliche Erläuterungen

Für die Felder „**fromStation**“ und „**toStation“** gelten folgende fachliche und technische Einschränkungen:

- Haltestellen werden im Rahmen der Streckenbeschreibung über standardisierte Identifikatoren referenziert, wie beispielsweise die deutschlandweit verwendete DHID (Deutschlandweite Haltestellen-ID).
- Diese Kennungen ermöglichen eine eindeutige Zuordnung von Start- und Zielpunkten und bilden die Grundlage für die fachliche Interpretation von relationsbasierten Tickets.

Für das Feld „**validRegionDescription“** gelten folgende fachliche und technische Einschränkungen:

- Es dient der menschenlesbaren Beschreibung des Gültigkeitsbereichs eines Tickets.
- Die fachliche Bewertung erfolgt ausschließlich auf Basis der strukturierten Daten (z. B. Haltestellen-IDs). Die Beschreibung dient ausschließlich der Anzeige und hat keinen Einfluss auf die technische Validierung.

7.2.26 SHT-Fahrgast

Das Schema „SHT-Fahrgast“ beschreibt die personenbezogenen Angaben zu einem Ticket im SH-Tarif. Dieses Schema wird im Rahmen der Ticketausstellung verwendet und ist Bestandteil des SHT-Tickets. Es wird immer dann übermittelt, wenn ein Ticket für einen konkreten Fahrgast erzeugt wird.

Fachlich werden hier die Daten erfasst, die zur Identifikation des Fahrgasts und zur tariflichen Einordnung erforderlich sind. Dazu gehören insbesondere Name, Geburtsdatum sowie der Fahrgasttyp, beispielsweise Erwachsener, Kind oder weitere tarifliche Kategorien. Diese Angaben sind insbesondere bei personengebundenen Tickets von Bedeutung, da sie im Kontrollprozess zur Überprüfung der Berechtigung herangezogen werden. Die Daten werden im Ticket hinterlegt und können bei der Kontrolle mit einem Ausweisdokument abgeglichen werden.

Das Schema ist somit ein Bestandteil der Ticketdefinition und wird bei der Erstellung eines SHT-Tickets gemeinsam mit den Produktdaten übermittelt.

Tabelle 57: Datenschema SHT-Fahrgast

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
firstName				Ja		
lastName				Ja		
passengerType				Ja		
dateOfBirth				Nein		

Tabelle 58: Beispiel JSON SHT-Fahrgast

```
{
  "firstName":"Maxima",
  "lastName":"Musterfrau",
  "passengerType":0,
  "dateOfBirth":"1990-05-30",
}
```

7.2.27 SHT-Tariftyp

Das Schema „SHT-Tariftyp“ beschreibt die tarifliche Zusammensetzung eines Tickets im SH-Tarif. Dieses Schema wird im Rahmen der Ticketausstellung als Bestandteil des SHT-Produkts verwendet. Es kommt immer dann zum Einsatz, wenn ein Ticket mehrere Fahrgäste oder unterschiedliche tarifliche Ausprägungen abbildet.

Fachlich wird hier festgelegt, wie sich das Ticket zusammensetzt, beispielsweise in Bezug auf die Anzahl der Fahrgäste, deren Fahrgasttypen und mögliche Ermäßigungen.

Diese Informationen sind insbesondere für die korrekte Preisbildung sowie für die Interpretation des Tickets im Kontrollprozess relevant. Sie geben Aufschluss darüber, für wen und unter welchen Bedingungen das Ticket gültig ist.

Das Schema ermöglicht somit die strukturierte Abbildung auch komplexerer tariflicher Konstellationen innerhalb eines Tickets.

Tabelle 59: Datenschema SHT-Tariftyp

Feld	Datenfeld im Barcode	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
numberOfPassengers				Ja		
passengerType				Ja		
reductionCard				Nein		

Tabelle 60: Beispiel JSON SHT-Tariftyp

```
{
  "numberOfPassengers":1,
  "passengerType":0,
  "reductionCard":"BC25"
}
```

7.2.28 Kontrollort

Das Schema „Kontrollort“ beschreibt den Ort, an dem eine Ticketkontrolle durchgeführt wird. Dieses Schema wird im Rahmen der Kontrolle verwendet und ist Bestandteil der Kontrollanfrage. Es kommt immer dann zum Einsatz, wenn der Ort der Prüfung für die Bewertung des Tickets relevant ist.

Fachlich liefert der Kontrollort den räumlichen Kontext der Kontrolle. Er gibt an, wo sich der Fahrgast zum Zeitpunkt der Prüfung befindet, beispielsweise an einer Haltestelle, in einem Verkehrsmittel oder auf einer Strecke. Diese Information ist insbesondere bei strecken- oder relationsgebundenen Tickets von Bedeutung. Sie wird genutzt, um zu prüfen, ob sich der Fahrgast innerhalb des zulässigen Geltungsbereichs des Tickets befindet. Diese Prüfung findet aktuell nicht im UIC-Sicherheitsportal statt, sondern muss stattdessen bei Bedarf im Kontrollgerät implementiert werden. Der Kontrollort kann damit dazu beitragen, die Ticketgültigkeit im konkreten Nutzungskontext korrekt zu bewerten.

Zur eindeutigen Referenzierung eines Kontrollorts wird die Deutschlandweite Haltestellen-ID (DHID) verwendet (siehe Glossar). Alternativ kann der Kontrollort über geografische Koordinaten (Breiten- und Längengrad) angegeben werden, wenn keine DHID verfügbar ist.

Tabelle 61: Datenschema Kontrollort

Feld	Beschreibung	Datentyp	Pflichtfeld	Wertebereich	Beispiel
dhid	Deutschlandweite Haltestellen-Identifikation (DHID) des nächstgelegenen Bahnhofs	string	Ja	Max.: 20 Zeichen	"de:01056:17016", null
latitude	Ungefährer Breitengrad während Kontrolle (gekürzt auf erste Dezimalstelle, ohne vorherige Rundung)	string	Ja	Min.: -90 Max.: 90	"53.7", null
longitude	Ungefährer Längengrad während Kontrolle (gekürzt auf erste Dezimalstelle, ohne vorherige Rundung)	string	Ja	Min.: -180 Max.: 180	"9.6", null

Tabelle 62: Beispiel JSON Kontrollort Variante 1

```
{
  "dhid":"de:01056:17016",
  "latitude":null,
  "longitude":null
}
```

Tabelle 63: Beispiel JSON Kontrollort Variante 2

```
{
  "dhid":null,
  "latitude":"53.7",
  "longitude":"9.6"
}
```

7.3 Antwort

Die Antwortschemata beschreiben die Daten, die das UIC-Sicherheitsportal nach der Verarbeitung einer Anfrage an das aufrufende System zurückliefert.

Fachlich stellen diese Antworten die Rückmeldung des Systems auf eine durchgeführte Aktion dar oder liefern strukturierte Informationen zur weiteren Verarbeitung im angebundenen System. Im Kontext der Ticketkontrolle bilden die Antwortdaten die Grundlage für die fachliche Bewertung eines Tickets. Sie enthalten neben dem eigentlichen Kontrollergebnis auch ergänzende Informationen, die zur Einordnung der Situation herangezogen werden können.

Darüber hinaus werden Antwortdaten auch zur Synchronisation von Systemen genutzt, beispielsweise beim Abruf von Sperrlisten. In diesen Fällen dienen sie dazu, lokale Datenbestände im Kontrollsystem aktuell zu halten.

Im operativen Einsatz werden Antwortdaten beispielsweise in folgenden Situationen genutzt:

- Bei der Kontrolle eines Tickets erhält das Kontrollsystem eine Rückmeldung darüber, ob das Ticket gültig ist und ob zusätzliche Hinweise vorliegen, etwa bei mehrfacher Nutzung innerhalb kurzer Zeit.
- Beim Abruf von Sperrlisten erhält ein System eine Übersicht verfügbarer Daten sowie die zugehörigen Einträge, um lokale Sperrlisten zu aktualisieren.
- Bei der Ticketausstellung liefert die Antwort die Bestätigung, dass ein Ticket erfolgreich erzeugt wurde, und stellt die relevanten Ticketinformationen zur weiteren Verwendung bereit.
- Bei administrativen Abfragen, etwa zur Analyse oder Nachverfolgung, geben die Antworten Auskunft über den aktuellen Status eines Tickets oder bereits durchgeführte Kontrollen.

Die Bedeutung der einzelnen Felder ergibt sich dabei stets aus dem jeweiligen Anwendungsfall. Insbesondere im Kontrollprozess ist das Zusammenspiel mehrerer Informationen entscheidend für die Interpretation des Ergebnisses.

7.3.1 Eintrag Sperrlistenübersicht

Das Schema „Eintrag Sperrlistenübersicht“ beschreibt eine einzelne Information innerhalb der Übersicht verfügbarer Sperrlisten. Dieses Schema wird verwendet, wenn ein System eine Übersicht über vorhandene Sperrlisten abruft, beispielsweise zur Vorbereitung einer Synchronisation im Kontrollsystem.

Fachlich dient ein solcher Eintrag dazu, eine Sperrliste eindeutig zu identifizieren und deren Aktualität zu bewerten. Die enthaltenen Informationen ermöglichen es dem aufrufenden System zu entscheiden, ob eine Sperrliste neu geladen oder aktualisiert werden muss.

Typischerweise enthält ein Eintrag Angaben wie eine eindeutige Kennung der Sperrliste sowie Informationen zum Erstellungs- oder Aktualisierungszeitpunkt. Diese Daten werden genutzt, um lokale Datenbestände im Kontrollsystem aktuell zu halten.

Der Eintrag selbst enthält noch keine konkreten gesperrten Tickets, sondern dient ausschließlich der Orientierung und Steuerung des Abrufs.

Tabelle 64: Datenschema Eintrag Sperrlistenübersicht

Feld	Beschreibung	Datentyp	Wertebereich	Beispiel
blacklistId	ID der Sperrliste	int		1
productOwner	Besonderheit: Tarifgeber, dessen Tickets auf dieser Sperrliste vermerkt sind. Default = 1	int	0 = alle 1 = D-Ticket 3634 = DTV 3735 = KVV	0
createdAt	Erstelldatum der Sperrliste	string	ISO 8601 Date	"2025-02-12T13:00:00.065319+00:00"
numberOfEntries	Anzahl Sperrlisteneinträge	int		2

Tabelle 65: Beispiel JSON Eintrag Sperrlistenübersicht

```
{
  "blacklistId":1,
  "productOwner":0,
  "createdAt":"2025-02-12T13:00:00.065319+00:00",
  "numberOfEntries":2
}
```

7.3.2 Sperrlistenübersicht

Das Schema „Sperrlistenübersicht“ beschreibt die Gesamtheit der verfügbaren Sperrlisten, die vom UIC-Sicherheitsportal bereitgestellt werden. Dieses Schema wird verwendet, wenn ein System eine Übersicht über alle verfügbaren Sperrlisten abruft, beispielsweise um den eigenen Datenbestand zu prüfen und gegebenenfalls zu aktualisieren.

Fachlich dient die Sperrlistenübersicht dazu, dem aufrufenden System einen vollständigen Überblick über den aktuellen Stand der Sperrlisten zu geben. Sie besteht aus mehreren Einträgen, die jeweils eine einzelne Sperrliste beschreiben. Auf Basis dieser Übersicht kann ein System entscheiden, welche Sperrlisten neu geladen oder aktualisiert werden müssen, um den aktuellen Sperrstatus von Tickets korrekt berücksichtigen zu können.

Die Sperrlistenübersicht ist damit ein zentrales Element für die Synchronisation zwischen dem UIC-Sicherheitsportal und angebundenen Kontrollsystemen, insbesondere in Szenarien mit Offline-Fähigkeit.

Tabelle 66: Datenschema Sperrlistenübersicht

Feld	Beschreibung	Datentyp	Wertebereich	Beispiel
	Liste der vorhandenen Sperrlisten	Array Typ: 7.3.1		Siehe Beispiel – JSON

Tabelle 67: Beispiel JSON Sperrlistenübersicht


```
[
  {
    "blacklistId":4,
    "productOwner":0,
    "createdAt":"2025-02-13T13:00:00.065319+00:00",
    "numberOfEntries":5
  },
  {
    "blacklistId":3,
    "productOwner":0,
    "createdAt":"2025-02-12T13:00:00.065319+00:00",
    "numberOfEntries":3
  }
]
```

7.3.3 Sperrlisteneintrag

Das Schema „Sperrlisteneintrag“ beschreibt ein einzelnes gesperrtes Ticket innerhalb einer Sperrliste. Dieses Schema wird verwendet, wenn eine Sperrliste abgerufen wird und die darin enthaltenen gesperrten Tickets an ein System übermittelt werden, beispielsweise zur Nutzung in einem Kontrollgerät.

Fachlich enthält ein Sperrlisteneintrag die Informationen, die erforderlich sind, um ein Ticket eindeutig zu identifizieren und als gesperrt zu kennzeichnen. Grundlage hierfür ist die Kombination aus Organisationscode, TicketId und Gültigkeitsende, wobei das Gültigkeitsende kein expliziter Bestandteil der Sperrliste ist. Allerdings enthält eine Sperrliste immer nur diejenigen Tickets, deren Gültigkeit zum Zeitpunkt ihrer Erstellung begonnen und noch nicht geendet hat.

Diese Angaben ermöglichen es dem aufrufenden System, gesperrte Tickets lokal zu speichern und bei einer Kontrolle zu berücksichtigen. So kann auch ohne direkte Verbindung zum UIC-Sicherheitsportal festgestellt werden, ob ein Ticket gesperrt ist.

Sperrlisteneinträge sind damit ein zentraler Bestandteil der Offline-Fähigkeit von Kontrollsystemen und stellen sicher, dass gesperrte Tickets systemweit erkannt werden.

Tabelle 68: Datenschema Sperrlisteneintrag

Feld	Beschreibung	Datentyp	Wertebereich	Beispiel
rics				
ticketId				

Tabelle 69: Beispiel JSON Sperrlisteneintrag

```
{
  "rics":"5143",
  "ticketId":"A0815BF0"
}
```

7.3.4 Sperrliste

Das Schema „Sperrliste“ beschreibt eine vollständige Sammlung von gesperrten Tickets, die vom UIC-Sicherheitsportal bereitgestellt wird. Dieses Schema wird verwendet, wenn eine konkrete Sperrliste abgerufen wird, um die darin enthaltenen gesperrten Tickets in ein angebundenes System zu übernehmen.

Fachlich besteht eine Sperrliste aus mehreren Sperrlisteneinträgen, die jeweils ein gesperrtes Ticket repräsentieren. Sie stellt damit den aktuellen Stand der bekannten Sperrungen für einen bestimmten Zeitpunkt dar. Die Sperrliste dient dazu, diese Informationen gebündelt bereitzustellen und eine Übertragung in lokale Systeme zu ermöglichen. Dadurch können Kontrollsysteme ihre eigenen Datenbestände aktualisieren und sicherstellen, dass gesperrte Tickets bei der Prüfung berücksichtigt werden.

Insbesondere in Offline-Szenarien bildet die Sperrliste die Grundlage dafür, dass Sperrungen auch ohne direkte Verbindung zum UIC-Sicherheitsportal erkannt werden können. In Online-Szenarien kann sie ergänzend genutzt werden, um lokale Prüfungen zu unterstützen.

Auch für den Download der Offline-Sperrliste wird ein API-Key benötigt. Dies verhindert, dass die Sperrliste öffentlich bekannt wird und von unseriösen Anbietern genutzt wird.

Die Offline-Sperrliste wird periodisch generiert. Eine neue Liste wird nur erzeugt, wenn sich der Inhalt der Liste ändern würde. Im Produktivsystem erfolgt diese Prüfung mit einem Intervall von 60 Minuten. Im Testsystem beträgt das Intervall 5 Minuten.

Jede Sperrliste hat eine eindeutige und fortlaufende ID. Beim Anfordern der aktuellen Sperrliste kann ein System die letzte bekannte Sperrlisten-ID angeben. Die API gibt dann nur die Sperrliste zurück, wenn es eine neuere Sperrliste gibt. Andernfalls wird der Status-Code „304 - Not Modified“ zurückgegeben.

Die Sperrliste ist in den Formaten CSV und JSON verfügbar. Für die Verwendung in Vertriebs- und Kontrollsystemen wird das JSON-Format empfohlen, da es auch die Sperrlisten-Nummer enthält. Bei der CSV-Datei ist diese Nummer ausschließlich im Dateinamen enthalten.

Tabelle 70: Datenschema Sperrliste

Feld	Beschreibung	Datentyp	Wertebereich	Beispiel
blacklistId	ID der Sperrliste	int		1
productOwner	Besonderheit: Tarifgeber, dessen Tickets auf dieser Sperrliste vermerkt sind. Default = 1	int	0 = alle 1 = D-Ticket 3634 = DTV 3735 = KVV	0
createdAt	Erstelldatum der Sperrliste	string	ISO 8601 Date	"2025-02-12T13:00:00.065319+00:00"
numberOfEntries	Anzahl Sperrlisteneinträge	int		2
tickets	Auf der Sperrliste befindliche Tickets	Array Typ: 7.3.3		Siehe Beispiel – JSON

Tabelle 71: Beispiel JSON Sperrliste


```
{
  "blacklistId":1,
  "productOwner":0,
  "createdAt":"2025-02-12T13:00:00.065319+00:00",
  "numberOfEntries":2,
  "tickets":[
    {
      "rics":"9999",
      "ticketId":"1337"
    },
    {
      "rics":"9999",
      "ticketId":"1338"
    }
  ]
}
```

7.3.5 Kontrollereignis

Das Schema „Kontrollereignis“ beschreibt ein einzelnes Ereignis im Zusammenhang mit der Prüfung eines Tickets. Dieses Schema wird im Rahmen einer Ticketkontrolle als Bestandteil der Antwort des UIC-Sicherheitsportals zurückgegeben. Es kommt immer dann zum Einsatz, wenn zusätzliche Informationen zur Einordnung einer Kontrolle bereitgestellt werden.

Fachlich stellt ein Kontrollereignis eine Ergänzung zum eigentlichen Kontrollergebnis dar. Hierüber können beispielsweise ungewöhnliche Nutzungsmuster erkannt werden. Ein typisches Beispiel ist die Mehrfachkontrolle eines Tickets innerhalb eines bestimmten Zeitraums. In diesem Fall wird ein entsprechender Hinweis zurückgegeben, der auf eine mögliche auffällige Nutzung hindeutet.

Kontrollereignisse verändern nicht das eigentliche Kontrollergebnis, sondern ergänzen dieses um zusätzliche Kontextinformationen. Sie sind daher als unterstützende Hinweise zu verstehen und nicht als eigenständige Bewertung der Ticketgültigkeit. Die im Kontrollereignis enthaltenen Informationen können im Kontrollsystem angezeigt und für weitergehende Auswertungen genutzt werden.

Die konkrete Ausgestaltung der einzelnen **Ereignistypen** sowie die zugrunde liegenden Schwellenwerte und Zeiträume werden im folgenden Kapitel im **Abschnitt zu den validityFlags** näher beschrieben.

Tabelle 72: Datenschema Kontrollereignis

Feld	Beschreibung	Datentyp	Wertebereich	Beispiel
id	Gibt die Art des Kontrollereignisses an	int		1
description	Beschreibt die Ereignisart	string		"Mehrfachkontrolle"
details	Weitere Details zum Kontrollereignis	string		"Das Ticket wurde mehr als 1 Mal kontrolliert in den letzten 5 Minuten"
data	Gibt die rohen Daten an, die zum Ereignis geführt haben. Der Objekttyp ist abhängig von der Id.	object		{ "interval": 5, "validations": 1 }

Tabelle 73: Beispiel JSON Kontrollereignis

```
{
  "id":1,
  "description":"Mehrfachkontrolle",
  "details":"Das Ticket wurde mehr als 1 Mal kontrolliert in den letzten 5 Minuten",
  "data":{
    "interval":5,
    "validations":1
  }
}
```

7.3.6 Kontrollergebnis

Das Schema „Kontrollergebnis“ beschreibt die fachliche Bewertung eines Tickets im Rahmen einer Kontrolle und stellt die zentrale Rückmeldung des UIC-Sicherheitsportals dar. Dieses Schema wird bei jeder Ticketprüfung zurückgegeben und bildet die Grundlage für die Anzeige und Bewertung im Kontrollsystem. Besondere Bedeutung kommt dabei den sogenannten validityFlags zu, die zusätzliche Hinweise zur Nutzung und Bewertung eines Tickets liefern.

Fachlich setzt sich das Kontrollergebnis aus mehreren Komponenten zusammen, die gemeinsam interpretiert werden müssen. Es lässt sich in drei Ebenen gliedern: die Gültigkeitsbewertung als Entscheidungsgrundlage, ergänzende Hinweise sowie Kontextinformationen aus der bisherigen Nutzung.

Im Mittelpunkt steht die Aussage zur Gültigkeit eines Tickets. Diese Entscheidung basiert auf verschiedenen Prüfungen, wie beispielsweise der kryptografischen Signatur, dem Gültigkeitszeitraum sowie dem aktuellen Status des Tickets.

Ergänzend dazu können zusätzliche Hinweise enthalten sein, die auf besondere Situationen oder Auffälligkeiten hinweisen. Diese Hinweise werden im Feld validityFlags bereitgestellt und enthalten strukturierte Informationen zu konkreten Kontrollereignissen.

Jeder Eintrag in validityFlags beschreibt dabei ein fachliches Ereignis und darf pro Ereignistyp maximal einmal enthalten sein.

Tabelle 74: Datenschema Kontrollergebnis

Feld	Beschreibung	Datentyp	Wertebereich	Beispiel
isValid	Gibt an, ob das abgefragte Ticket gültig ist	boolean	true/false	true
validityFlags	Gibt optional weitere Informationen über den Gültigkeitszustand an	Array Typ: 7.2.5 Kontrollereignis	ID 1 und 2	Siehe Beispiel – JSON
errorMessage	Gibt bei ungültigen Tickets einen Grund an	String		"Ticket is locked"
lastUpdate	Gibt den Zeitpunkt an, an dem die Informationen über das Ticket zuletzt aktualisiert wurden	string	ISO 8601 Date	"2025-02-12T13:00:00.065319+00:00"
lastValidation	Gibt den Zeitpunkt an, wann das Ticket zuletzt kontrolliert wurde	string	ISO 8601 Date	"2025-02-12T13:00:00.065319+00:00"

Tabelle 75: Beispiel JSON Kontrollergebnis


```
{
  "isValid":true,
  "validityFlags":[
    {
      "id":1,
      "description":"Mehrfachkontrolle",
      "details":"Das Ticket wurde mehr als 1 Mal kontrolliert in den letzten 5 Minuten",
      "data":{
        "interval":5,
        "validations":1
      }
    }
    {
      "id":2,
      "description":"Unplausible Reisegeschwindigkeit",
      "details":"Reisegeschwindigkeit von über 43300 km/h zwischen zwei Kontrollen",
      "data":{
        "speedKmh":43300,
        "distanceKm":473
        "timeMinutes":0
      }
    }
  ],
  "errorMessage":null,
  "lastUpdate":"2025-06-19T08:17:13.522299+00:00",
  "lastValidation":"2025-06-23T09:01:21.3458736+00:00"
}
```


Zusätzliche Erläuterungen

Es gilt für das Feld **„isValid“** folgende fachliche Bedeutung:

- Es stellt die fachliche Gesamtbewertung eines Tickets dar.
- Die Bewertung basiert auf der Prüfung des Ticketstatus im UIC-Sicherheitsportal und ist in Verbindung mit den Ergebnissen der lokalen Prüfung auf dem Kontrollgerät (Signaturprüfung, Gültigkeitszeitraum) im Kontrollprozess zu nutzen.

Es gilt für das Feld **„errorMessage“** folgende fachliche Bedeutung:

- Es dient der ergänzenden Beschreibung eines Ergebnisses und ist primär für die Anzeige oder Protokollierung vorgesehen.
- Es ist nicht für die fachliche Bewertung oder systemseitige Entscheidungslogik zu verwenden.

Es gilt für das Feld **„validityFlags“** folgende fachliche Bedeutung:

- Die enthaltenen Hinweise stellen keine eigenständige Bewertung dar, sondern ergänzen das Kontrollergebnis und sind im Kontext der Gesamtbewertung zu interpretieren.

Erläuterungen zu den validityFlags

Ereignistyp Flag-ID 1: Mehrfachkontrolle innerhalb eines bestimmten Zeitraums

Dieser Ereignistyp kennzeichnet den Fall, dass ein Ticket innerhalb definierter Zeitfenster wiederholt kontrolliert wurde. Dies kann ein Hinweis auf eine mögliche missbräuchliche Nutzung sein, beispielsweise durch Weitergabe eines Screenshots.

Die Bewertung erfolgt auf Basis folgender Schwellenwerte:

- mehr als 1 Kontrolle innerhalb von 5 Minuten
- mehr als 3 Kontrollen innerhalb von 60 Minuten
- mehr als 6 Kontrollen innerhalb von 24 Stunden

Zusätzlich enthält ein solcher Eintrag erläuternde Informationen, die im Kontrollsystem angezeigt werden können, beispielsweise als Hinweis zur Prüfung von Personalien. Ein Kontrollereignis enthält dabei neben der Kennung auch erläuternde Informationen, die im Kontrollsystem genutzt werden können:

- Der Wert „description“ stellt einen kurzen Hinweis dar, der im Kontrollgerät als Hauptmeldung angezeigt wird.
- Der Wert „details“ enthält eine weiterführende Erläuterung, die die Situation genauer beschreibt und dem Kontrollpersonal zusätzliche Orientierung bietet.
- Das Feld „data“ enthält strukturierte Zusatzinformationen zum Ereignis:
  - „Interval“: Zeitraum in Minuten, in dem die Mehrfachkontrolle erfasst wurde
  - „Validations“: Anzahl erfolgter Kontrollen im Intervallzeitraum

Diese Informationen können von Systemen genutzt werden, um auf Wunsch spezifische Darstellungen umzusetzen.

Ereignistyp Flag-ID 2: Geokontrolle (unplausible Reisegeschwindigkeit)

Dieser Ereignistyp kennzeichnet den Fall, dass zwischen zwei erfassten Kontrollen eines Tickets eine theoretische Reisegeschwindigkeit ermittelt wurde, die als fachlich nicht plausibel bewertet wird. Dies kann ein Hinweis auf eine mögliche missbräuchliche Nutzung sein, beispielsweise durch parallele Nutzung eines Tickets durch mehrere Personen.

Die Bewertung erfolgt auf Basis der berechneten Distanz zwischen zwei Kontrollereignissen in Verhältnis zur verstrichenen Zeit. Überschreitet dieses Verhältnis den definierten Schwellenwert von 300 km/h, wird das Ereignis entsprechend gekennzeichnet. Zusätzlich enthält ein solcher Eintrag erläuternde Informationen, die im Kontrollsystem angezeigt werden können. Ein Kontrollereignis enthält dabei neben der Kennung auch strukturierte Zusatzinformationen:

- Der Wert „description“ stellt einen kurzen Hinweis dar, der im Kontrollgerät als Hauptmeldung angezeigt wird.
- Der Wert „details“ enthält eine weiterführende Erläuterung, die die berechnete Reisegeschwindigkeit sowie die zugrunde liegende Strecke beschreibt.

- Das Feld „data“ enthält strukturierte Zusatzinformationen zum Ereignis:
  - „speedKmh“: berechnete Reisegeschwindigkeit in km/h,
  - „distanceKm“: zugrunde gelegte Entfernung in Kilometern,
  - „timeMinutes“: Zeitdifferenz zwischen den Kontrollereignissen in Minuten.

Diese Informationen können von Systemen genutzt werden, um Auffälligkeiten weiter zu bewerten oder gezielt Hinweise für das Kontrollpersonal bereitzustellen.

Beispielhafte Anzeige im Kontrollsystem

Im Kontrollsystem sollen Hinweise aus den validityFlags typischerweise zusätzlich zum eigentlichen Kontrollergebnis dargestellt werden.

Tabelle 76: Beispiel Warnmeldung Kontrolle

Warnung
Gegebenenfalls die Personalien prüfen!
Das Ticket wurde mehr als 1 Mal kontrolliert in den letzten 5 Minuten.
Die theoretische Reisegeschwindigkeit von 20025km/h zwischen zwei Kontrollen ist unrealistisch.

Der Hinweis wird ergänzend zur Gültigkeitsanzeige dargestellt und unterstützt das Kontrollpersonal bei der weiteren Bewertung der Situation. Sie sind als Warninformationen zu verstehen. Sie ergänzen das Kontrollergebnis, verändern jedoch nicht die eigentliche Gültigkeitsbewertung. Darüber hinaus können Informationen zu bereits erfolgten Kontrollen enthalten sein. Diese liefern Kontext zur Nutzung des Tickets und ermöglichen eine Bewertung im zeitlichen Zusammenhang.

Fazit

Die Gültigkeitsbewertung ist die alleinige Grundlage für die Entscheidung, ob ein Ticket akzeptiert wird. Diese setzt sich zusammen aus der lokalen Prüfung der Ticketsignatur und des Gültigkeitszeitraums sowie dem Ticketstatus im UIC-Sicherheitsportal. Ergänzende Hinweise und Kontextinformationen dienen ausschließlich der Unterstützung und müssen getrennt von der eigentlichen Bewertung betrachtet werden.

Im operativen Einsatz wird das Kontrollergebnis beispielsweise wie folgt genutzt:

- Bei einer regulären Kontrolle zeigt das System an, ob ein Ticket gültig ist und genutzt werden kann.
- Bei auffälligen Nutzungsmustern werden zusätzliche Hinweise ausgegeben, die eine intensivere Prüfung nahelegen können, beispielsweise die Überprüfung von Personalien.
- Bei wiederholten Kontrollen kann die Historie genutzt werden, um die aktuelle Situation besser einzuordnen.

Das Kontrollergebnis bildet damit die zentrale Entscheidungsgrundlage im Kontrollprozess und ist maßgeblich für die fachliche Bewertung eines Tickets.

7.3.7 Ticket

Das Schema „Ticket“ beschreibt die Ticketdaten, die im Rahmen einer Antwort vom UIC-Sicherheitsportal zurückgegeben werden. Dieses Schema wird verwendet, wenn im Rahmen einer Anfrage ein konkretes Ticket erzeugt wird.

Fachlich dient das Schema dazu, das angefragte Ticket eindeutig zu identifizieren und dessen wesentliche Eigenschaften prüfen zu können. Die enthaltenen Informationen entsprechen den dem System übergebenen Ticketdaten und ermöglichen es dem aufrufenden System, das Ticket zu überprüfen und im jeweiligen Kontext weiterzuverarbeiten. Für die Prüfung stehen insbesondere die Identifikationsmerkmale sowie der Gültigkeitszeitraum im Vordergrund, da diese für die fachliche Einordnung eines Tickets maßgeblich sind. Zur Weiterverarbeitung steht das Feld ticketData im Zentrum, welches den in einen Aztec-Code umzuwandelnden Hex-Code enthält.

Je nach Anwendungsfall können zusätzlich Informationen zum Produkt oder zum Fahrgast enthalten sein, die weiteren Kontext liefern.

Tabelle 77: Datenschema Ticket

Feld	Beschreibung	Datentyp	Wertebereich	Beispiel
ticketId				
issuerRics				
securityProviderRics				
keyId				
ticketData	Ticket enkodiert als Hexadezimal String	string	Hexadecimal	"4F6B6579023A7DF…"
title	Optional Ansprache oder Titel des Ticketinhabers	string	3 Zeichen	„Hr“
firstName				
secondName	Optionaler zweiter Name des Ticketinhabers	string		„Peter“
lastName				
dateOfBirth				
gender				
productId				
productOwner				
tariffDescription				
priceInCents				
validFrom				
validTo				
issuedAt				
issuerAuthorizationId				
issuerKvpId				
uniqueOrderId				
ticketReference				

Tabelle 78: Beispiel JSON Ticket


```
{
   "ticketId":"A0815BF0",
   "issuerRics":"5143",
   "securityProviderRics":"3634",
   "keyId":"DTX04",
   "ticketData":"4F6B6579023A7DF…",
   "title":"",
   "firstName":"Maxima",
   "secondName":"",
   "lastName":"Musterfrau",
   "dateOfBirth":"1990-05-30",
   "gender":1,
   "productId":9999,
   "productOwner":"3634",
   "tariffDescription":"Deutschlandticket",
   "priceInCents":6300,
   "validFrom":"2025-02-01T00:00:00+01:00",
   "validTo":"2025-03-01T03:00:00+01:00",
   "issuedAt":"2025-01-25T02:00:00+01:00",
   "issuerAuthorizationId":15312312,
   "issuerKvpId":6321,
   "uniqueOrderId":"28d07a4f-02ee-4c6f-996d-89e47670157b"
   "ticketReference":"",
}
```

7.3.8 Ticket-Barcode

Das Schema „Ticket-Barcode“ beschreibt die Barcode-Daten eines Tickets, die vom UIC-Sicherheitsportals zurückgegeben werden. Dieses Schema wird verwendet, wenn im Rahmen der Ticketausstellung ein Barcode erzeugt und an das aufrufende System übermittelt wird.

Fachlich stellt der Barcode die digitale Repräsentation des Tickets dar. Er enthält alle für die Nutzung und Kontrolle relevanten Informationen in codierter Form und wird dem Fahrgast zur Anzeige, beispielsweise in einer App oder als Ausdruck, bereitgestellt. Der Barcode bildet die Grundlage für die Kontrolle, da er im Prüfprozess ausgelesen und die enthaltenen Daten zur Validierung herangezogen werden. Dabei ist entscheidend, dass der Barcode unverändert verwendet wird. Eine Veränderung oder eigene Interpretation der enthaltenen Daten ist nicht zulässig, da dies die korrekte Prüfung des Tickets beeinträchtigen kann.

Das Schema stellt somit sicher, dass Tickets systemübergreifend einheitlich bereitgestellt und im Kontrollprozess zuverlässig interpretiert werden können. Zusätzlich kann ein sogenannter EntityTag (entityTag) zurückgegeben werden.

- Der EntityTag dient dazu, Änderungen am Barcode oder den zugrunde liegenden Daten zu erkennen. Er ermöglicht es dem aufrufenden System festzustellen, ob sich ein bereits bekannter Stand verändert hat, ohne die vollständigen Daten erneut verarbeiten zu müssen.
- Fachlich unterstützt der EntityTag damit eine effiziente Synchronisation und Aktualisierung von Daten, insbesondere in Szenarien, in denen Informationen wiederholt abgefragt oder zwischengespeichert werden.
- Wenn sich der EntityTag nicht geändert hat, kann davon ausgegangen werden, dass auch die zugehörigen Daten unverändert sind.
- Der EntityTag entspricht einem ETag-Mechanismus aus dem HTTP-Kontext und dient der effizienten Änderungsprüfung von Ressourcen.

Tabelle 79: Datenschema Ticket-Barcode

Feld	Beschreibung	Datentyp	Wertebereich	Beispiel
fileContents	Bild (PNG) als Base64 String	string	Base64	"U29mdHdhcmUg…"
contentType	Inhaltstyp	string		"image/png"
fileDownloadName	Dateiname für Download	string		"A0815BF0.png"
lastModified	Zuletzt geändert am	ISO 8601 Date		"2025-01-25T02:33:43+01:00"
entityTag	ETag (Entity Tag)	object		Siehe Beispiel – JSON
enableRangeProcessing	Gibt an, ob Range Processing erlaubt ist.	boolean	true/false	true

Tabelle 80: Beispiel JSON Ticket-Barcode


```
{
  "fileContents":"U29mdHdhcmUg…",
  "contentType":"image/png",
  "fileDownloadName":"A0815BF0.png",
  "lastModified":"2025-01-25T02:33:43+01:00",
  "entityTag":{
    "tag":{
      "buffer":"dW5pcXVlVGFn",
      "offset":0,
      "length":10,
      "value":"uniqueTag",
      "hasValue":true
    },
    "isWeak":false
  },
  "enableRangeProcessing":true
}
```

8 API-Endpunkte

Die nachfolgenden API-Endpunkte beschreiben die konkreten Zugriffspunkte auf die Funktionen des UIC-Sicherheitsportals.

Fachlich bilden sie die operativen Anwendungsfälle des Systems ab. Über die Endpunkte werden die zentralen Prozesse in den Clustern Authentifizierung, Sperrliste, Ticket und Kontrolle technisch umgesetzt.

Während die vorangegangenen Kapitel die zugrunde liegenden Datenstrukturen und Systemprinzipien beschreiben, zeigen die API-Endpunkte, wie diese Funktionen im konkreten Anwendungsfall aufgerufen werden. Für das fachliche Verständnis ist dabei entscheidend, dass ein Endpunkt nicht nur als technischer Aufruf zu verstehen ist, sondern jeweils einen bestimmten Vorgang im Ticket-Lebenszyklus auslöst oder unterstützt.

Im operativen Einsatz werden API-Endpunkte beispielsweise genutzt, um ein Ticket auszustellen, eine Sperrliste für Kontrollgeräte bereitzustellen, den Status eines Tickets (Stornierung, Sperrung und Entsperrung) zu ändern oder im Rahmen einer Kontrolle die Gültigkeit eines Tickets zu prüfen.

Die nachfolgenden Abschnitte ordnen die einzelnen Endpunkte daher nicht nur technisch, sondern auch fachlich ein.

8.1 Authentifizierung mit OAuth2 (Auth)

POST: /api/v1/auth/token

Beschreibung

Dieser Endpunkt dient der **Anforderung eines Zugriffstokens (Access-Token)** für die Nutzung der API. Die fachlichen Grundlagen der Authentifizierung sowie die zugrunde liegenden Verfahren sind in Kapitel 6 beschrieben.

Die Token-Anforderung erfolgt im Rahmen des OAuth2-Verfahrens entweder:

- über Client Credentials zur initialen Authentifizierung,
- oder über ein Refresh-Token zur Erneuerung eines bestehenden Tokens.

Fachliche Einordnung

- Der Aufruf stellt den Übergang von einer nicht authentifizierten zu einer authentifizierten Nutzung der API dar.
- Im Erfolgsfall wird ein Access-Token zurückgegeben, das bei nachfolgenden API-Aufrufen im Header mitgeführt wird und als Nachweis der Berechtigung dient.
- Im laufenden Betrieb erfolgt die Erneuerung des Tokens in der Regel über das Refresh-Token, ohne dass eine erneute Anmeldung mit den ursprünglichen Zugangsdaten erforderlich ist. Dies gilt nicht für die Nutzung über JWT Bearer Assertion.

Voraussetzungen

- Für die Nutzung mit Client Credentials ist es erforderlich, dass der Login für den entsprechenden Zugang im UIC-Sicherheitsportal zuvor freigeschaltet wurde.
- Diese Freischaltung erfolgt einmalig für die initiale Authentifizierung und wird nach erfolgreicher Anmeldung automatisch wieder deaktiviert.
- Für die Nutzung mit JWT Bearer Assertion müssen im UIC-Sicherheitsportal entsprechende Zugangsdaten eingerichtet sowie ein Public Key zur Signaturprüfung hinterlegt sein.

Authentifizierung

via Request-Parameter

Notwendige Berechtigungen

Keine, da ausschließlich für OAuth2-Flow

Request-Parameter

Type: application/x-www-form-urlencoded

Tabelle 81: Request Parameter token

Name	Art	Beschreibung	Datentyp	Pflichtfeld	Beispiel
client_id	Body	ID des Clients	string	Ja	"198483d7-d6f6-444e-a75e-edb7d082e24d"
client_secret	Body	Geheimnis des Clients	string	Ja	"ef7f4dbb-39e0-427f-879f-9041a0465323"
grant_type	Body	Art des Grants	string	Ja	"client_credentials” oder "refresh_token" oder "jwt-bearer-assertion"
refresh_token	Body	Refresh-Token, wenn ein neues Token angefordert wird.	string	Nein	"eyJhbGciOiJIUzI1... "
scope	Body	Umfang der Berechtigungen	string	Nein	
assertion	Body	Assertion-Token	String	Nein	"eyJhbGciOiJIUzI1... "

Tabelle 82: HTTP-Antworten token

200 (OK):

{ "access_token": "eyJhbGciOiJIUzI1...", "token_type": "bearer", "expires_in": 3600, "refresh_token": "eyJhbGciOiJIUzI1..." }

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnten bspw. falsche / invalide Parameter angegeben sein. Fehlerantworten werden im OAuth2-Standard zurückgegeben.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte die falsche Client-ID und/oder das falsche Client-Secret oder ein ungültiger Refresh-Token angegeben sein. Fehlerantworten werden im OAuth2-Standard zurückgegeben.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

8.2 Sperrlistenabruf (Blacklist)

Die Sperrlistenendpunkte werden genutzt, um eine Liste der verfügbaren Sperrlisten zu erhalten oder eine bestimmte oder die aktuelle Sperrliste abzurufen.

Üblicherweise wird immer die aktuelle Sperrliste abgerufen. Dabei kann die ID der aktuell im Gerät hinterlegten Sperrliste mitgegeben werden, so dass nur dann eine Sperrliste zurückgegeben wird, wenn eine neuere Version verfügbar ist.

Der Abruf einer bestimmten Sperrliste kann beispielsweise dann sinnvoll sein, wenn mehrere Kontrollgeräte zu unterschiedlichen Zeitpunkten anfragen, aber aus Konsistenzgründen denselben Datenstand erhalten sollen.

GET: /api/v1/blacklist

Beschreibung

Der Endpunkt liefert eine **Liste der verfügbaren Sperrlisten**, die typischerweise einen Zeitraum von bis zu zwei Wochen abdecken. Hierbei kann über den Product Owner gefiltert werden, für welchen Tarif die Sperrlisten abgerufen werden sollen.

Details zur Verarbeitung

Die Verarbeitung erfolgt synchron. Die Liste der verfügbaren Sperrlisten wird unmittelbar zurückgegeben.

Authentifizierung

API-Key / OAuth2

Notwendige Berechtigungen

Sperrliste herunterladen

Request Parameter

Tabelle 83: Request Parameter blacklist

Name	Art	Beschreibung	Datentyp	Pflichtfeld	Beispiel
productOwner	Query	Tariffilter der Sperrliste	int	Ja	1 für D-Ticket, 0 für alle Tarife

Tabelle 84: HTTP-Antworten blacklist

200 (OK):

Sperrlistenübersicht (siehe Schema 7.3.2)

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein falscher API-Key oder ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

GET: /api/v1/blacklist/{id}

Beschreibung

Der Endpunkt liefert den Inhalt einer **bestimmten Sperrliste**.

- Die Sperrliste wird über ihre eindeutige ID (Unique ID of Blacklist) identifiziert, die im Pfadparameter „id“ übergeben wird.
- Die Sperrliste wird nach den benötigten Tarifprodukten gefiltert.
- Das Ausgabeformat kann über den Query-Parameter „format“ gesteuert werden. Unterstützte Formate sind „json“ und „csv“.

Details zur Verarbeitung

Die Verarbeitung erfolgt synchron. Der Inhalt der Sperrliste wird unmittelbar im angeforderten Format zurückgegeben.

Authentifizierung

API-Key / OAuth2

Notwendige Berechtigungen

Sperrliste herunterladen

Request Parameter

Tabelle 85: Request Parameter blacklist/id

Name	Art	Beschreibung	Datentyp	Pflichtfeld	Beispiel
id	Path	Eindeutige ID der Sperrliste	int	Ja	1
productOwner	Query	Tariffilter der Sperrliste	int	Ja	1 für D-Ticket, 0 für alle Tarife
format	Query	Format der Sperrliste	string	Nein	"json" oder "csv", Default: "json"

Tabelle 86: HTTP-Antwort blacklist/id

200 (OK) :

Sperrliste (siehe Schema 7.3.4)

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnten bspw. falsche / invalide Parameter angegeben sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein falscher API-Key oder ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

GET: /api/v1/blacklist/latest

Beschreibung

Der Endpunkt liefert den **Inhalt der neuesten Sperrliste**. Das Ausgabeformat kann über den Query-Parameter „format“ gesteuert werden. Unterstützte Formate sind „json“ und „csv“.

Über den Product Owner wird gesteuert, für welche Tarife die Sperrliste geladen werden soll. Über den optionalen Parameter „lastVersion“ kann die zuletzt bekannte Sperrlisten-Version übergeben werden. Eine Sperrliste wird nur dann zurückgegeben, wenn eine neuere Version vorliegt.

Details zur Verarbeitung

Die Verarbeitung erfolgt synchron. Der Inhalt der Sperrliste wird unmittelbar im angeforderten Format zurückgegeben.

Authentifizierung

API-Key / OAuth2

Notwendige Berechtigungen

Sperrliste herunterladen

Request Parameter

Tabelle 87: Request Parameter blacklist/latest

Name	Art	Beschreibung	Datentyp	Pflichtfeld	Beispiel
format	Query	Format der Sperrliste	string	Nein	"json" oder "csv", Default: "json"
produc-tOwner	Query	Tariffilter der Sperrliste	int	Ja	1 für D-Ticket, 0 für alle Tarife
lastVersion	Query	Bereits heruntergeladene Sperrlisten-Version	int	Nein	1

Tabelle 88: HTTP Antworten blacklist/latest

200 (OK):

Sperrliste (siehe Schema 7.3.4)

304 (Not Modified):

Wenn der optionale Parameter „lastVersion“ angegeben wird, liefert der Server nur dann eine neue Liste, wenn eine neuere Version verfügbar ist. Andernfalls gibt der Server den Status-Code „304 - Not Modified“ zurück.

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnten bspw. falsche / invalide Parameter angegeben sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein falscher API-Key oder ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

8.3 Interaktionen im Ticketkontext (Ticket)

Über die Endpunkte im Bereich Ticket können Barcodes für verschiedene Tarife abgerufen und deren Status im UIC-Sicherheitsportal verändert werden. Das technische Statusmodell im UIC-Sicherheitsportal sieht drei verschiedene Statusmarker vor: unbekannt/ausgegeben, gesperrt/nicht gesperrt und storniert/nicht storniert. Fachlich werden daraus die zwei Status gültig und ungültig abgeleitet.

Folgende Szenarien sind in diesem Modell abbildbar:

Szenario	Status technisch	Status fachlich
Ausgabenachweis wird übermittelt	Ausgegeben	Gültig
Kontrollnachweis für gesperrtes Ticket wird übermittelt	Gesperrt	Ungültig
Kontrollnachweis für storniertes Ticket wird übermittelt	Storniert	Ungültig
Kontrollnachweis für unbekanntes Ticket wird übermittelt	Unbekannt	Gültig
Sperrung für unbekanntes Ticket wird übermittelt	Gesperrt	Ungültig
Sperrung für ausgegebenes Ticket wird übermittelt	Gesperrt	Ungültig
Sperrung eines bereits gesperrten Tickets wird übermittelt	Gesperrt	Ungültig
Stornierung für ausgegebenes Ticket wird durchgeführt	Storniert	Ungültig
Stornierung eines bereits stornierten Tickets wird durchgeführt	Storniert	Ungültig
Stornierung für unbekanntes Ticket wird durchgeführt	Storniert	Ungültig
Entsperrung für unbekanntes Ticket wird übermittelt	Unbekannt	Gültig
Entsperrung für ausgegebenes Ticket wird übermittelt	Ausgegeben	Gültig
Entsperrung für gesperrtes Ticket wird durchgeführt	Ausgegeben oder Unbekannt	Gültig
Entsperrung eines nicht gesperrten Tickets wird übermittelt	Unverändert	Gültig

Die dargestellten Szenarien zeigen, dass das UIC-Sicherheitsportal Statusänderungen unabhängig von der Reihenfolge der Ereignisse verarbeiten kann.

Insbesondere ist es möglich, Statusänderungen wie Sperrungen oder Stornierungen auch für Tickets zu übermitteln, die dem System zum Zeitpunkt der Anfrage noch nicht bekannt sind. Diese werden entsprechend vorgemerkt und bei späteren Ereignissen berücksichtigt.

Darüber hinaus werden wiederholte Statusänderungen in denselben Zustand (z. B. erneute Sperrung eines bereits gesperrten Tickets) ignoriert und führen zu keiner weiteren Veränderung. Fachlich ergibt

sich daraus ein robustes Modell, das auch bei asynchronen oder verzögerten Ereignissen eine konsistente Bewertung des Ticketstatus ermöglicht.

Es ergeben sich folgende Rahmenbedingungen für die Interaktion mit Tickets im UIC-Sicherheitsportal:

- Die Kontrolle unbekannter Tickets führt zur Anlage gültiger Tickets in der Datenbank.
- Auch unbekannte Tickets können im UIC-Sicherheitsportal gesperrt und entsperrt werden.
- Übermittlungen von Statuswechseln in einen Status, den ein Ticket bereits hat, werden ignoriert.

POST: /api/v1/ticket/lock

Beschreibung

Mit diesem Endpunkt kann die **Sperrung von Tickets** angefordert werden.

Die Sperrung eines Tickets führt dazu, dass:

- das Ticket als gesperrt markiert wird,
- das Ticket in die Sperrliste aufgenommen wird,
- die Sperrung an die KOSE übermittelt wird
- und bei nachfolgenden Kontrollen als ungültig bewertet werden kann.

Die Sperrung kann über den Endpunkt zum Entsperren wieder aufgehoben werden.

Details zur Verarbeitung

- Die Sperrung kann im Batch durchgeführt werden. Pro Anfrage können bis zu 10.000 Tickets übermittelt werden.
- Die Anfrage wird nach erfolgreicher Syntaxprüfung quittiert. Die vollständige Verarbeitung erfolgt asynchron und kann einige Sekunden bis Minuten dauern.
- Wird dieselbe Anfrage erneut übermittelt, hat dies keine zusätzliche Wirkung und führt nicht zu einem Fehler.
- Wird ein bereits gesperrtes Ticket erneut gesperrt oder ein nicht gesperrtes Ticket entsperrt, wird die Anfrage ohne weitere Wirkung verarbeitet.

Authentifizierung

API-Key / OAuth2

Notwendige Berechtigungen

Ticket sperren

Es können nur Tickets mit RICS- oder Organisations-Code gesperrt werden, für die der API-Keys autorisiert ist. Es können maximal 10.000 Tickets pro Anfrage gesendet werden.

Request Body

Type: application/json

Content: Tickets (siehe Schema 7.2.2)

Tabelle 89: HTTP Antworten lock

202 (Accepted):

Die Anfrage und deren Inhalte wurden erfolgreich verarbeitet.

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein falscher API-Key oder ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

POST: /api/v1/ticket/unlock

Beschreibung

Mit diesem Endpunkt kann das **Entsperren von Tickets** angefordert werden.

Das Entsperren eines Tickets führt dazu, dass:

- das Ticket nicht mehr als gesperrt geführt wird,
- das Ticket von der Sperrliste entfernt wird,
- die Entsperrung an die KOSE übermittelt wird
- und bei nachfolgenden Kontrollen wieder als gültig bewertet werden kann, sofern keine weiteren Einschränkungen vorliegen.

Das Entsperren wirkt ausschließlich auf den Sperrstatus und stellt keine erneute Ticketausgabe dar.

Details zur Verarbeitung

- Das Entsperren kann im Batch durchgeführt werden.
- Pro Anfrage können bis zu 10.000 Tickets übermittelt werden.
- Die Anfrage wird nach erfolgreicher Syntaxprüfung quittiert.

- Die vollständige Verarbeitung erfolgt asynchron und kann einige Sekunden bis Minuten dauern.
- Wird dieselbe Anfrage erneut übermittelt, hat dies keine zusätzliche Wirkung und führt nicht zu einem Fehler.

Authentifizierung

OAuth2

Notwendige Berechtigungen

Ticket entsperren

Es können nur Tickets mit Organisationscode (RICS-Code) entsperrt werden, für die der OAuth2-Client autorisiert ist. Es können maximal 10.000 Tickets pro Anfrage gesendet werden.

Request Body

Type: application/json

Content: Tickets (siehe Schema 7.2.2)

Tabelle 90: HTTP Antworten unlock

202 (Accepted):
Die Anfrage und deren Inhalte wurden erfolgreich verarbeitet.
400 (Bad Request):
Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.
401 (Unauthorized):
Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/ Refresh-Token angegeben sein.
403 (Forbidden):
Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.
429 (Too Many Requests):
Das Anfrage-Limit wurde überschritten.

POST: /api/v1/ticket/cancel

Beschreibung

Mit diesem Endpunkt kann die **Stornierung von Tickets** angefordert werden.

Die Stornierung eines Tickets führt dazu, dass:

- das Ticket als ungültig markiert wird,
- das Ticket in die Sperrliste aufgenommen wird,
- sowie die ursprüngliche Ticketausgabe fachlich zurückgenommen wird.

Eine Stornierung ist endgültig und kann nicht rückgängig gemacht werden.

Details zur Verarbeitung

- Die Stornierung kann im Batch durchgeführt werden.
- Pro Anfrage können bis zu 10.000 Tickets übermittelt werden.
- Die Anfrage wird nach erfolgreicher Syntaxprüfung quittiert.
- Die vollständige Verarbeitung erfolgt asynchron und kann einige Sekunden bis Minuten dauern.
- Wird eine bereits verarbeitete Anfrage erneut gesendet, bleibt das Ergebnis unverändert.

Authentifizierung

OAuth2

Notwendige Berechtigungen

Ticketausstellung stornieren

Es können nur Tickets mit Organisationscode (RICS-Code) storniert werden, für die der OAuth2-Client autorisiert ist. Es können maximal 10.000 Tickets pro Anfrage gesendet werden.

Request Body

Type: application/json

Content: Tickets (siehe Schema 7.2.2)

Tabelle 91: HTTP Antworten cancel

202 (Accepted):
Die Anfrage und deren Inhalte wurden erfolgreich verarbeitet.
400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

POST: /api/v1/ticket/issuancerecord

Beschreibung

Mit diesem Endpunkt können **Ausgabenachweise für Tickets** im UIC-Sicherheitsportal erfasst werden.

Ein Ausgabenachweis dokumentiert die Ausstellung eines Tickets im System. Er bildet die Grundlage für:

- die Nachvollziehbarkeit der Ticketausgabe,
- die fachliche Bewertung im Rahmen der Ticketkontrolle,
- sowie die Verarbeitung in nachgelagerten Systemen.

Ein Ticket kann auch ohne vorliegenden Ausgabenachweis kontrolliert werden. In diesem Fall das Ticket durch die Kontrolle in der Datenbank angelegt, aber nicht als ausgegeben markiert.

Details zur Verarbeitung

- Die Übermittlung von Ausgabenachweisen kann im Batch durchgeführt werden. Pro Anfrage können bis zu 10.000 Einträge übermittelt werden.
- Die Anfrage wird nach erfolgreicher Syntaxprüfung quittiert. Die vollständige Verarbeitung erfolgt asynchron und kann einige Sekunden bis Minuten dauern.
- Wird derselbe Ausgabenachweis erneut übermittelt, hat dies keine zusätzliche Wirkung und führt nicht zu einem Fehler.

Authentifizierung

OAuth2

Notwendige Berechtigungen

Ausgabenachweis veröffentlichen

Ausgabenachweise können nur für Tickets mit RICS- oder Organisations-Code veröffentlicht werden, für die der OAuth2-Client autorisiert ist. Es können maximal 10.000 Tickets pro Anfrage gesendet werden.

Request Body

Type: application/json

Content: Ausgabenachweise (siehe Schema 7.2.6)

Tabelle 92: HTTP Antworten issuancerecord

202 (Accepted):

Die Anfrage und deren Inhalte wurden erfolgreich verarbeitet.

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

POST: /api/v1/ticket/issuance/dticket

Beschreibung

Mit diesem Endpunkt wird ein **Deutschlandticket ausgestellt**. Dies führt zur Erzeugung eines gültigen Tickets inklusive aller relevanten Metadaten.

Dabei wird:

- ein signierter Ticketinhalt erzeugt,
- der Ticketinhalt hexadezimal kodiert zurückgegeben,
- sowie automatisch ein Ausgabenachweis im System erstellt.

Der Ticketaussteller entspricht dem Organisationscode (RICS), der im Rahmen der Authentifizierung verwendet wird. Das erzeugte Ticket bildet die Grundlage für die Nutzung durch den Fahrgast sowie für die spätere Kontrolle.

Details zur Verarbeitung

- Die Ausstellung erfolgt synchron. Das erzeugte Ticket wird unmittelbar als Antwort zurückgegeben.
- Wird dieselbe Anfrage erneut übermittelt, kann dies zur mehrfachen Ausstellung eines Tickets führen.

Authentifizierung

OAuth2

Notwendige Berechtigungen

Deutschlandticket ausstellen

Request Body

Type: application/json

Content: Deutschlandticket (siehe Schema 7.2.7)

Tabelle 93: HTTP Antworten issuance/dticket

200 (OK):

Ticket (siehe Schema 7.3.7)

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

POST: /api/v1/ticket/issuance/dtv

Beschreibung

Mit diesem Endpunkt wird ein Barcode für ein Produkt aus dem Deutschlandtarif ausgestellt. Die Ausstellung führt zur Erzeugung eines gültigen Ticket-Barcodes inklusive aller relevanten Metadaten.

Dabei wird:

- ein signierter Ticketinhalt erzeugt,
- der Ticketinhalt hexadezimal kodiert zurückgegeben,
- sowie automatisch ein Ausgabenachweis im System erstellt.

Der Ticketaussteller entspricht dem Organisationscode (RICS), der im Rahmen der Authentifizierung verwendet wird. Der erzeugte Barcode stellt die digitale Repräsentation des Tickets dar und dient als Grundlage für Anzeige und Kontrolle.

Details zur Verarbeitung

- Die Ausstellung erfolgt synchron. Der erzeugte Ticketinhalt wird unmittelbar als Antwort zurückgegeben.
- Wird dieselbe Anfrage erneut übermittelt, kann dies zur mehrfachen Ausstellung eines Tickets führen.

Authentifizierung

OAuth2

Notwendige Berechtigungen

Ticket ausstellen

Request Body

Type: application/json

Content: D-Tarif-Ticket (siehe Schema 7.2.8)

Tabelle 94: HTTP Antworten issuance/dtv

200 (OK):
Ticket (siehe Schema 7.3.7)
400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

POST: /api/v1/ticket/issuance/kvv

Beschreibung

Mit diesem Endpunkt wird ein Barcode für ein Produkt aus dem KVV-Tarif ausgestellt. Die Ausstellung führt zur Erzeugung eines gültigen Ticket-Barcodes inklusive aller relevanten Metadaten.

Dabei wird:

- ein signierter Ticketinhalt erzeugt,
- der Ticketinhalt hexadezimal kodiert zurückgegeben,
- sowie automatisch ein Ausgabenachweis im System erstellt.

Der Ticketaussteller entspricht dem Organisationscode (RICS), der im Rahmen der Authentifizierung verwendet wird. Der erzeugte Barcode stellt die digitale Repräsentation des Tickets dar und dient als Grundlage für Anzeige und Kontrolle.

Details zur Verarbeitung

- Die Ausstellung erfolgt synchron. Der erzeugte Ticketinhalt wird unmittelbar als Antwort zurückgegeben.
- Wird dieselbe Anfrage erneut übermittelt, kann dies zur mehrfachen Ausstellung eines Tickets führen.

Authentifizierung

OAuth2

Notwendige Berechtigungen

Ticket ausstellen

Request Body

Type: application/json

Content: KVV-Ticket (siehe Schema 7.2.12)

Tabelle 95: HTTP Antworten issuance/kvv

200 (OK):

Ticket (siehe Schema 7.3.7)

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

POST: /api/v1/ticket/issuance/NOX

Beschreibung

Mit diesem Endpunkt wird ein Barcode für ein Ticket von NOX mobility ausgestellt. Die Ausstellung führt zur Erzeugung eines gültigen Ticket-Barcodes inklusive aller relevanten Metadaten.

Dabei wird:

- ein signierter Ticketinhalt erzeugt,
- der Ticketinhalt hexadezimal kodiert zurückgegeben,
- sowie automatisch ein Ausgabenachweis im System erstellt.

Der Ticketaussteller entspricht dem Organisationscode (RICS), der im Rahmen der Authentifizierung verwendet wird. Der erzeugte Barcode stellt die digitale Repräsentation des Tickets dar und dient als Grundlage für Anzeige und Kontrolle.

Details zur Verarbeitung

- Die Ausstellung erfolgt synchron. Der erzeugte Ticketinhalt wird unmittelbar als Antwort zurückgegeben.

- Wird dieselbe Anfrage erneut übermittelt, kann dies zur mehrfachen Ausstellung eines Tickets führen.

Authentifizierung

OAuth2

Notwendige Berechtigungen

Ticket ausstellen

Request Body

Type: application/json

Content: NOX-Ticket (siehe Schema 7.2.16)

Tabelle 96: HTTP Antworten issuance/nox

200 (OK):

Ticket (siehe Schema 7.3.7)

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

POST: /api/v1/ticket/issuance/nds

Beschreibung

Mit diesem Endpunkt wird ein Barcode für ein Produkt aus dem Niedersachsen-Tarif ausgestellt. Die Ausstellung führt zur Erzeugung eines gültigen Ticket-Barcodes inklusive aller relevanten Metadaten.

Dabei wird:

- ein signierter Ticketinhalt erzeugt,
- der Ticketinhalt hexadezimal kodiert zurückgegeben,
- sowie automatisch ein Ausgabenachweis im System erstellt.

Der Ticketaussteller entspricht dem Organisationscode (RICS), der im Rahmen der Authentifizierung verwendet wird. Der erzeugte Barcode stellt die digitale Repräsentation des Tickets dar und dient als Grundlage für Anzeige und Kontrolle.

Details zur Verarbeitung

- Die Ausstellung erfolgt synchron. Der erzeugte Ticketinhalt wird unmittelbar als Antwort zurückgegeben.
- Wird dieselbe Anfrage erneut übermittelt, kann dies zur mehrfachen Ausstellung eines Tickets führen.

Authentifizierung

OAuth2

Notwendige Berechtigungen

Ticket ausstellen

Request Body

Type: application/json

Content: NDS-Ticket (siehe Schema 7.2.20)

Tabelle 97: HTTP Antworten issuance/nds

200 (OK):

Ticket (siehe Schema 7.3.7)

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

POST: /api/v1/ticket/issuance/sht

Beschreibung

Mit diesem Endpunkt wird ein Barcode für ein Produkt aus dem SH-Tarif ausgestellt. Die Ausstellung führt zur Erzeugung eines gültigen Ticket-Barcodes inklusive aller relevanten Metadaten.

Dabei wird:

- ein signierter Ticketinhalt erzeugt,
- der Ticketinhalt hexadezimal kodiert zurückgegeben,
- sowie automatisch ein Ausgabenachweis im System erstellt.

Der Ticketaussteller entspricht dem Organisationscode (RICS), der im Rahmen der Authentifizierung verwendet wird. Der erzeugte Barcode stellt die digitale Repräsentation des Tickets dar und dient als Grundlage für Anzeige und Kontrolle.

Details zur Verarbeitung

- Die Ausstellung erfolgt synchron. Der erzeugte Ticketinhalt wird unmittelbar als Antwort zurückgegeben.
- Wird dieselbe Anfrage erneut übermittelt, kann dies zur mehrfachen Ausstellung eines Tickets führen.

Authentifizierung

OAuth2

Notwendige Berechtigungen

Ticket ausstellen

Request Body

Type: application/json

Content: SHT-Ticket (siehe Schema 7.2.24)

Tabelle 98: HTTP Antworten issuance/sht

200 (OK): Ticket (siehe Schema 7.3.7) **400 (Bad Request):** Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein. **401 (Unauthorized):** Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/ Refresh-Token angegeben sein. **403 (Forbidden):** Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen. **429 (Too Many Requests):** Das Anfrage-Limit wurde überschritten.

GET: /api/v1/ticket/issuance/renderbarcode

Beschreibung

Dieser Endpunkt dient der Umwandlung eines Ticketinhalts, der als Hexadezimalstring übergeben wird, in eine grafische Darstellung des **Barcodes im PNG**-Format.

Der Endpunkt stellt **keine Ticketausstellung** oder Ticketprüfung dar. Es handelt sich um eine technische Hilfsfunktion zur Darstellung eines bereits vorhandenen Tickets.

Use Case

- Der Endpunkt wird insbesondere zu **Test-, Analyse- und Supportzwecken** verwendet. Er ermöglicht die visuelle Darstellung eines Ticketinhalts und die Überprüfung der korrekten Barcode-Erzeugung.
- Der erzeugte Barcode entspricht der Darstellung, wie sie auch im operativen Einsatz (z. B. in mobilen Anwendungen oder als Ausdruck) verwendet wird.
- Der Endpunkt ist **nicht für die produktive Ticketerstellung** vorgesehen.

Details zur Verarbeitung

Die Verarbeitung erfolgt synchron. Das erzeugte Barcode-Bild wird unmittelbar als PNG zurückgegeben.

Authentifizierung

keine

Notwendige Berechtigungen

keine

Request Parameter

Tabelle 99: Request Parameter renderbarcode

Name	Art	Beschreibung	Datentyp	Pflichtfeld	Beispiel
ticketHex	Query	Ticket als Hexadezimal String	string	Ja	"4F6B6579023A7DF…"
width	Query	Breite des PNG	int	Nein	320, Default: 256
height	Query	Höhe des PNG	int	Nein	320, Default: 256

Tabelle 100: HTTP Antworten renderbarcode

200 (OK):

Ticket-Barcode (siehe Schema 7.3.8)

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnten bspw. falsche / invalide Parameter angegeben sein.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

8.4 Kontrolle (Validation)

Die Endpunkte im Bereich Kontrolle dienen dem **Austausch von Daten im Rahmen der Ticketprüfung**.

Dabei wird zwischen **zwei fachlichen Anwendungsfällen** unterschieden:

- der unmittelbaren Prüfung eines Tickets im Kontrollmoment (Online-Kontrolle)
- der nachgelagerten Übermittlung von bereits durchgeführten Kontrollen (Kontrollnachweise)

Beide Anwendungsfälle greifen auf dieselben fachlichen Grundlagen zurück, unterscheiden sich jedoch in ihrem Zeitpunkt der Verarbeitung sowie in der Art der Systemrückmeldung.

Die Kontrolle basiert stets auf den im UIC-Barcode enthaltenen Ticketdaten sowie auf den im UIC-Sicherheitsportal verfügbaren Informationen, insbesondere Sperrlisten und Kontrollhistorien.

Das Ergebnis einer Kontrolle dient im operativen Kontext als Grundlage für die Bewertung der Ticketgültigkeit und unterstützt das Kontrollpersonal bei der Entscheidung im Fahrgastkontakt.

POST: /api/v1/validation/validate

Beschreibung

Mit Aufruf dieses Endpunkts wird ein Ticket im UIC-Sicherheitsportal geprüft.

Die Prüfung erfolgt auf Basis der übermittelten Ticketdaten sowie der im System gespeicherten Informationen. Das Ergebnis wird unmittelbar als Kontrollergebnis zurückgegeben.

Dieser Endpunkt wird im Moment der Ticketprüfung verwendet, typischerweise durch Kontrollgeräte oder Kontrollanwendungen im Fahrgastkontakt.

Fachlich dient er der unmittelbaren Bewertung, ob ein Ticket gültig ist und genutzt werden darf. Das zurückgelieferte Kontrollergebnis bildet die Entscheidungsgrundlage im Kontrollprozess.

Nur im Rahmen dieser Online-Kontrolle werden über die reine Gültigkeitsbewertung hinaus zusätzliche Kontextinformationen bereitgestellt. Dazu zählen insbesondere:

- der Zeitpunkt der letzten Kontrolle
- Hinweise auf vorherige Kontrollereignisse
- sowie abgeleitete Informationen, beispielsweise zur räumlichen oder zeitlichen Einordnung der Nutzung

Diese Informationen unterstützen das Kontrollpersonal bei der Bewertung der konkreten Situation, haben jedoch keinen Einfluss auf die eigentliche Gültigkeitsbewertung.

Details zur Verarbeitung

Im Rahmen der Verarbeitung werden die übermittelten Ticketdaten mit den im System verfügbaren Informationen abgeglichen.

Dies umfasst insbesondere:

- Prüfung des Ticketstatus (z. B. aktiv, gesperrt, storniert)

- Abgleich mit der Sperrliste
- Bewertung des Gültigkeitszeitraums
- Auswertung der bisherigen Kontrollhistorie

Das Ergebnis wird als strukturiertes Kontrollergebnis zurückgegeben. Dieses enthält:

- die Gültigkeitsbewertung (isValid)
- optionale Hinweise (validityFlags)
- ergänzende Kontextinformationen

Optional kann die Kontrolle als Kontrollereignis im System gespeichert werden. Die Steuerung, ob eine Kontrolle als Ereignis gespeichert wird, erfolgt über das Feld writeValidationRecord.

- true (Default): Speicherung als Kontrollereignis
- false: keine Speicherung, rein informatorische Abfrage

Die korrekte Verwendung dieses Feldes ist insbesondere für die Nachvollziehbarkeit von Kontrollen und die Auswertung von Nutzungsmustern relevant.

Authentifizierung

API-Key / OAuth2

Notwendige Berechtigungen

Ticket kontrollieren

Request Body

Type: application/json

Content: Kontrolle (siehe Schema 7.2.3)

Tabelle 101: HTTP Antworten validate

200 (OK): Kontrollergebnis (siehe Schema 7.3.6) **400 (Bad Request):** Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

POST: /api/v1/validation/records

Beschreibung

Mit Aufruf dieses Endpunkts werden ein oder mehrere **Kontrollnachweise im UIC-Sicherheitsportal erfasst**. Die übermittelten Daten entsprechen bereits durchgeführten Kontrollen und werden im System gespeichert.

Dieser Endpunkt dient der nachgelagerten Übermittlung von Kontrollen, die nicht im Moment der Prüfung direkt an das System übertragen wurden.

Dies ist insbesondere relevant für:

- Offline-Kontrollen
- Systeme mit verzögerter Synchronisation
- nachträgliche Verarbeitung von Kontrollereignissen

Fachlich werden die übermittelten Kontrollnachweise wie reguläre Kontrollen behandelt und fließen in die Kontrollhistorie eines Tickets ein.

Details zur Verarbeitung

Die im Request enthaltenen Kontrollnachweise werden einzeln verarbeitet und als Kontrollereignisse im System gespeichert.

Für jeden Datensatz erfolgt:

- Zuordnung zum Ticket anhand der eindeutigen Ticketidentifikation
- Validierung der übermittelten Daten
- Speicherung als Kontrollereignis

Die Verarbeitung erfolgt unabhängig vom aktuellen Ticketstatus.

Die gespeicherten Daten stehen anschließend für:

- die Anzeige im Portal,

- die Auswertung von Kontrollhistorien und
- die Erkennung auffälliger Nutzungsmuster

zur Verfügung.

Im Gegensatz zum Endpunkt POST /api/v1/validation/validate erfolgt keine unmittelbare Rückgabe eines Kontrollergebnisses, sondern ausschließlich die Übermittlung und Speicherung von Kontrollereignissen.

Der Endpunkt dient somit der Dokumentation und nicht der Entscheidung im Kontrollmoment.

Authentifizierung

API-Key / OAuth2

Notwendige Berechtigungen

Kontrollnachweis veröffentlichen

Request Body

Type: application/json

Content: Kontrollnachweise (siehe Schema 7.2.4)

Tabelle 102: HTTP Antworten records

202 (Accepted):

Die Anfrage und deren Inhalte wurden erfolgreich verarbeitet.

400 (Bad Request):

Die Anfrage konnte nicht verarbeitet werden. Es könnte bspw. ein falscher / invalider Inhalt (Body) vorhanden sein.

401 (Unauthorized):

Die Authentifizierung ist fehlgeschlagen / nicht möglich. In diesem Fall könnte ein ungültiger Auth-/ Refresh-Token angegeben sein.

403 (Forbidden):

Der anfragende Client ist erfolgreich authentifiziert, verfügt jedoch nicht über die notwendigen Berechtigungen, um den Endpunkt aufrufen zu dürfen.

429 (Too Many Requests):

Das Anfrage-Limit wurde überschritten.

9 Fehlerbehandlung

Im Rahmen der Nutzung der API können unterschiedliche Arten von Rückmeldungen auftreten, die über den Verlauf und das Ergebnis einer Anfrage informieren.

Grundsätzlich ist dabei zwischen technischen Fehlern und fachlichen Ereignissen zu unterscheiden.

Technische Fehler treten auf, wenn eine Anfrage nicht verarbeitet werden kann, beispielsweise aufgrund fehlender Authentifizierung, fehlerhafter Parameter oder systemseitiger Einschränkungen. In diesen Fällen wird kein fachliches Ergebnis erzeugt.

Fachliche Ereignisse hingegen entstehen im Rahmen einer erfolgreichen Verarbeitung und beziehen sich auf die inhaltliche Bewertung des Vorgangs. Dies ist insbesondere bei der Ticketkontrolle relevant, bei der ein Ticket geprüft und bewertet wird, unabhängig davon, ob es gültig oder ungültig ist.

Für die Integration ist es daher entscheidend, diese beiden Ebenen klar zu unterscheiden und entsprechend zu verarbeiten. Technische Fehler können sowohl über HTTP-Statuscodes als auch über das im Antwortkörper enthaltene Fehlerobjekt zurückgemeldet werden. Während HTTP-Statuscodes den technischen Zustand der Anfrage auf Protokollebene beschreiben (z. B. Authentifizierung oder Zugriff), liefert das Fehlerobjekt im Response zusätzliche strukturierte Informationen zur Ursache innerhalb der API.

Für die Integration ist es daher erforderlich, beide Ebenen gemeinsam auszuwerten. Die nachfolgenden Fehlercodes beziehen sich ausschließlich auf technische Rückmeldungen.

Fehlercodes und Maßnahmen

Im Fehlerfall wird ein standardisierter Fehlercode an den aufrufenden Client zurückgemeldet. Die folgende Tabelle enthält neben der Beschreibung auch typische Maßnahmen zur Behebung.

Tabelle 103: Fehlercodes und Maßnahmen

Fehler-nummer	Fehlername	Beschreibung	Maßnahme
-1	Unknown	Unbekannter Fehler ist aufgetreten	Anfrage erneut ausführen; bei wiederholtem Auftreten Log prüfen und Support kontaktieren.
0	Ok	Alles in Ordnung	Keine Maßnahme erforderlich.
1	Unauthenticated	Fehlerhafte Authentifizierung	Neues Access-Token anfordern und Anfrage erneut ausführen.
2	Unauthorizied	Keine Berechtigung für die Aktion	Berechtigungen des API-Zugangs prüfen.
3	InvalidRicsUsed	Es wurde ein RICS- oder Organisations-Code verwendet, für den keine Berechtigung vorhanden ist.	Verwendeten Organisationscode prüfen und korrigieren.
4	SchemaValidation-Failed	Ein fehlerhaftes Schema wurde verwendet	Request-Daten prüfen; Details im Feld „validationErrors“ beachten.
5	ObjectNotFound	Ein Objekt, auf das verwiesen wurde, existiert nicht	Identifikationsdaten prüfen.
6	TicketIssuanceValid-tiyOutOfRange	Es wurde versucht ein Ticket mit einem Gültigkeitsmonat auszugeben, welche nicht in der erlaubten Range ist.	Gültigkeitszeitraum anpassen.
7	TicketIssuanceNoSig-ningKeyConfigured	Es wurde kein Signierschlüssel zur Ausgabe des Tickets konfiguriert.	Signierschlüssel im System konfigurieren.
8	BlacklistNotFound	Die angegeben Blacklist existiert nicht.	Bezeichnung der Blacklist prüfen.
9	RateLimitExceeded	Es wurden zu viele Anfragen gesendet.	Anfragen zeitlich verteilen und verzögert erneut senden.

Struktur der Fehlermeldung

Im Fehlerfall wird ein Ergebnisobjekt zurückgeliefert, das Informationen zur Fehlerursache enthält.

Tabelle 104: Struktur Fehlermeldung

Feld	Beschreibung	Daten-typ	Pflicht-feld	Wertebe-reich	Beispiel
errorCode	Fehlernummer, siehe Tabelle 103: Fehlercodes und Maßnahmen	int	-		1
errorCode-Description	Fehlername	string	-		"Unauthenticated"
errorMessage	Details zum aufgetretenen Fehler	string	-		"Invalid JWT Token"

Tabelle 105: Beispiel JSON Fehlermeldung

```
{
  "errorCode": 1,
  "errorCodeDescription": "Unauthenticated",
  "errorMessage": "Invalid JWT Token"
}
```

Erweiterte Fehlermeldung bei Schemafehlern

Im Fall eines Schemafehlers (Fehlernummer 4: „*SchemaValidationFailed*“) wird zusätzlich ein weiteres Feld im Ergebnis mitgeliefert. Dies gibt an welches Feld im Schema die Validierung nicht erfolgreich überstanden hat und ermöglicht es, fehlerhafte Eingaben gezielt zu identifizieren und zu korrigieren.

Tabelle 106: Struktur erweiterte Fehlermeldung bei Schemafehlern

Feld	Beschreibung	Datentyp	Wertebe-reich	Beispiel
validationEr-rors	Auflistung der fehler-haften Felder im Schema	object		"FirstName": "The FirstName field is required."

Tabelle 107: Beispiel JSON erweiterte Fehlermeldung bei Schemafehlern

```
{
  "validationErrors": {
    "FirstName": "The FirstName field is required."
  },
  "errorCode": 4,
  "errorCodeDescription": "SchemaValidationFailed",
  "errorMessage": "Error while parsing the request"
}
```

10 Ergänzungen und Anmerkungen

10.1 Betrieb & Support

Alle wichtigen Informationen zum Support finden Sie auch in **Anlage 4.2**.

Umgebungen

Das UIC-Sicherheitsportal wird auf einer Produktivumgebung betrieben, die unter folgender URL erreichbar ist: https://sicherheitsportal.deutschlandtarifverbund.de/

Zugang zur Produktivumgebung erhalten nur Organisationen und ihre Mitarbeitenden und Dienstleister, für die eine Nutzungsvereinbarung vorliegt, oder die über andere Verträge den geltenden Nutzungsbedingungen zugestimmt haben.

Neben der Produktivumgebung wird eine Stagingumgebung vorgehalten, die unter folgender URL erreichbar ist: https://test.sicherheitsportal.deutschlandtarifverbund.de/

Zugang zur Stagingumgebung ist auch ohne Nutzungsvereinbarung möglich, da dort keine Echtdaten vorgehalten werden.

Verfügbarkeit

Das UIC-Sicherheitsportal wird als zentraler Dienst für die Ausstellung und Kontrolle von Tickets betrieben. Ziel ist eine uneingeschränkte Verfügbarkeit zur Unterstützung der operativen Prozesse.

- Die Verfügbarkeit des Produktivsystems wird im Jahresmittel gemessen und es werden **mindestens 99,7 %** garantiert.
- Die Verfügbarkeit bezieht sich auf die Erreichbarkeit der API sowie die Funktion der bereitgestellten API-Endpunkte.
- Für das Stagingsystem gibt es keine garantierte Verfügbarkeit.

Besondere Hinweise

Kurzzeitige Einschränkungen im Rahmen von Wartungsarbeiten oder ungeplanten Störungen können nicht vollständig ausgeschlossen werden. Die Verfügbarkeit einzelner API-Endpunkte kann dabei temporär eingeschränkt sein. Bei geplanten Wartungsmaßnahmen erfolgt im Vorfeld eine Information.

Wartung

Zur Sicherstellung eines stabilen und sicheren Betriebs werden regelmäßig Wartungsarbeiten durchgeführt. Wartungsarbeiten können sowohl geplant als auch ungeplant erfolgen.

Geplante Wartung

Geplante Wartungsarbeiten werden in der Regel mit ausreichendem Vorlauf angekündigt. Ziel ist es, diese außerhalb der Hauptnutzungszeiten durchzuführen.

Ungeplante Wartung

Ungeplante Wartungsmaßnahmen können insbesondere zur Behebung von Störungen oder zur Schließung von Sicherheitslücken erforderlich sein und erfolgen kurzfristig.

Besondere Hinweise

Während Wartungsarbeiten kann es zu Einschränkungen oder temporärer Nichtverfügbarkeit der API oder einzelner API-Endpunkte kommen.

Störungen

Störungen sind ungeplante Beeinträchtigungen der Verfügbarkeit oder Funktionalität der API oder einzelner API-Endpunkte. Im Falle von Störungen wird eine schnellstmögliche Analyse und Behebung angestrebt.

Eine durchgängige Verfügbarkeit in Echtzeit kann nicht garantiert werden.

Integrationen sollten daher so ausgelegt sein, dass temporäre Nichtverfügbarkeiten abgefangen werden können, beispielsweise durch Wiederholversuche von Requests oder eine zeitversetzte Verarbeitung.

Fehlerfälle und Systemreaktionen sind im Kapitel 9 Fehlerbehandlung beschrieben.

Support

Für fachliche und technische Anfragen sowie zur Meldung von Störungen stehen zentrale Supportstrukturen zur Verfügung.

Service-Desk

Der Service Desk ist die zentrale Anlaufstelle für fachliche und technische Supportanfragen sowie für die Meldung von Störungen im Zusammenhang mit der Nutzung der API. Supportanfragen können jederzeit über das Ticketsystem eingereicht werden: https://amcon.atlassian.net/servicedesk/customer/portal/43/group/169/create/562

Die Bearbeitung erfolgt in der Regel werktags zwischen 09:00 Uhr und 16:00 Uhr.

Service-Hotline

Für dringende Störmeldungen steht der AMCON Service Desk telefonisch zur Verfügung.

Telefon: +49 4471 91420

Erreichbarkeit: werktags 09:00–16:00 Uhr

Ansprechpersonen bei der Deutschlandtarifverbund-GmbH

Darüber hinaus stehen Ihnen folgende Ansprechpersonen bei der Deutschlandtarifverbund-GmbH für Fragen und bei Unterstützungsbedarf zur Verfügung:

Dorothee Hoßbach – Business Application Managerin UIC-Sicherheitsportal
E-Mail: d.hossbach@deutschlandtarifverbund.de
Telefon: +49 151 11104823

Christian Skobijn – technischer Projektleiter UIC-Sicherheitsportal
E-Mail: c.skobjin@deutschlandtarifverbund.de
Telefon: +49 171 1541211

Besondere Hinweise

Für eine effiziente Bearbeitung von Anfragen sollten diese möglichst präzise beschrieben werden.

Hierzu zählen insbesondere Angaben zum betroffenen API-Endpunkt, zum Zeitpunkt des Requests sowie zu erhaltenen Responses oder Fehlermeldungen.

Hinweise für Integratoren

Die Integration der API sollte unter Berücksichtigung typischer Betriebsbedingungen erfolgen.

Temporäre Nichtverfügbarkeiten einzelner API-Endpunkte können auftreten.

Responses können in Ausnahmefällen verzögert zurückgegeben werden.

Besondere Hinweise

Es wird empfohlen:

- Wiederholmechanismen (Retry-Strategien) für fehlgeschlagene Requests zu implementieren
- Fehlerfälle systemseitig zu protokollieren (Logging)
- zeitkritische Prozesse nicht ausschließlich von einer synchronen Verarbeitung einzelner API-Requests abhängig zu machen

10.2 Rate Limiting

Die Nutzung der API des UIC-Sicherheitsportals unterliegt einer Begrenzung der Anzahl von Anfragen innerhalb eines bestimmten Zeitraums. Ziel dieser Begrenzung ist es, eine stabile und zuverlässige Verfügbarkeit des Systems für alle angeschlossenen Organisationen sicherzustellen.

Fachlich bedeutet dies, dass API-Aufrufe nicht unbegrenzt und beliebig gebündelt erfolgen dürfen, sondern gleichmäßig über die Zeit verteilt werden müssen.

Die Begrenzung erfolgt dabei pro technischem Zugang, also pro API-Key oder OAuth2-Client. Jede Organisation ist somit selbst dafür verantwortlich, die eigenen Anfragen so zu steuern, dass die vorgegebenen Grenzen eingehalten werden.

Für die Integration ergibt sich daraus die Anforderung, dass aufrufende Systeme ihr Anfrageverhalten entsprechend steuern. Insbesondere ist darauf zu achten, dass Lastspitzen vermieden werden und Anfragen nicht gesammelt in sehr kurzer Zeit gesendet werden.

Wird die zulässige Anzahl an Anfragen überschritten, werden weitere Anfragen temporär abgelehnt. Dies kann dazu führen, dass einzelne Funktionen kurzfristig nicht verfügbar sind, obwohl sie technisch korrekt aufgerufen wurden.

Typische Ursachen hierfür sind beispielsweise:

- das gleichzeitige Versenden vieler Anfragen ohne zeitliche Verteilung,
- wiederholte automatische Wiederholungsversuche ohne Steuerung,
- oder nicht abgestimmte parallele Systeme innerhalb einer Organisation.

Um dies zu vermeiden, sollten Systeme so gestaltet werden, dass Anfragen gleichmäßig verteilt und bei Ablehnungen kontrolliert erneut ausgeführt werden. In besonders lastintensiven Szenarien, wie beispiels-weise bei der massenhaften Ticketausstellung oder bei großflächigen Kontrollen, ist eine abgestimmte Nutzung der API besonders wichtig.

Die konkrete technische Umsetzung der Begrenzung erfolgt über ein sogenanntes Token-Bucket-Verfahren und ist nachfolgend beschrieben:

Alle Anfragen gegen die API unterliegen einem Rate Limit. Das Rate Limit wird pro API-Key bzw. pro

OAuth2-Credential festgelegt. Initial hinterlegt ist ein Standardwert von 300 Anfragen pro Minute, der nicht überschritten werden darf.

Technisch wird ein Token-Bucket verwendet, welcher anfangs mit einer konfigurierten Anzahl an Tokens (im Standard 300 Tokens) befüllt ist. Alle 10 Sekunden wird 1/6 des Buckets wieder aufgefüllt (im Standard 50 Tokens alle 10 Sekunden) bis zum Maximalwert (im Standard auch 300 Tokens). Jede Anfrage an die API nimmt genau einen Token aus dem Bucket heraus. Ist der Bucket leer, wird die jeweilige Anfrage mit dem Statuscode „429 – Too Many Requests“ abgelehnt.

Das Refill-Intervall von 1/6 pro 10 Sekunden bedeutet auch, dass eine Dauerbelastung der API von

(1/6 * Rate Limit) / 10 s nicht überschritten werden darf. Denn bei 300 Anfragen pro Minute dürfen unter Last entsprechend nur 50 Anfragen alle 10 Sekunden durchgeführt werden.

Das aufrufende System muss sich entsprechend darauf einstellen, dass im Lastbetrieb nicht direkt in der ersten Sekunde einer Minute das gesamte Rate Limit aufgebraucht wird (im Standard 300 Anfragen alle 60 Sekunden) und ohne Pause weiter Anfragen gesendet werden – dies würde zu einer Ablehnung mit Statuscode 429 führen. Die Anfragen sollten stattdessen über die gesamte Minute verteilt werden (im Standard 50 Anfragen alle 10 Sekunden).

Der Token-Bucket hat einen automatischen Puffer. Wenn mindestens eine Minute keine Anfrage gesendet wurde und der Bucket voll ist (im Standard 300 Tokens), kann in den nächsten 60 Sekunden die doppelte Anzahl des Rate Limits an Anfragen gesendet werden (im Standard 600 Anfragen), da sich der Bucket wiederum alle 10 Sekunden um eine feste Anzahl an Tokens regeneriert (im Standard 50 Tokens

alle 10 Sekunden). Der Ablauf wäre dann der folgende: Direkt zu Beginn 300 Anfragen und anschließend schrittweise 50 Anfragen alle 10 Sekunden.

Die tatsächlich erreichbare Anzahl möglicher Anfragen kann aufgrund von Race Conditions geringfügig vom eingestellten Rate Limit abweichen, jedoch nur nach oben (z.B. 305 Anfragen pro Minute bei einem Rate Limit von 300 Anfragen pro Minute).

Sollte das Standard Rate Limit von 300 Anfragen pro Minute nicht ausreichen, so kann dies auf Anfrage individuell erhöht werden.

10.3 Protokollierung

Alle Anfragen an das System werden von diesem protokolliert.

Um später zu Debugging-Zwecken Geräte/Systeme eindeutig identifizieren zu können, kann optional der HTTP-Header „Device-Id“ befüllt werden. Das System wertet dies nicht automatisch aus, jedoch können die Anfragen über HTTP-Accesslogs nachvollzogen werden, die für 14 Tage gespeichert werden. Dieser Header ist explizit kein Pflichtfeld.

11 Verzeichnisse

11.1 Abbildungsverzeichnis

11.2 Tabellenverzeichnis

11.3 Anlagenverzeichnis

Anlage 4.1 – Datenverwendungskonzept UIC-Sicherheitsportal

Anlage 4.2 – Quick Reference Card UIC-Sicherheitsportal