

Requests to APDS

APX Specification: technical documentation

Version 0.16.0 (draft)

Updated 2026-11-02

Introduction

This document lists, in one place, everything APX would like added to or changed in APDS. It covers the published API document (`apds-api-4.1.yaml`), the APDS 4.1 Data Dictionary, and the data elements APX proposes for the next APDS edition. Every request here is shared with the APDS committee as a request, not a demand.

None of these requests blocks APX. APX vendors APDS 4.1 byte-for-byte and checksum-guarded (`spec/vendor/apds/4.1/`), and never edits it. Where APX needs something APDS does not yet have, it works around the gap in its own overlay (`spec/openapi/overlays/apx-data-overlay.yaml`), in an `apds-ext` extension, or in an APX code list. Each workaround is narrow and points at the request it stands in for.

When APDS adopts a request, APX removes its workaround in the same release. Wherever APDS standardizes something APX currently fills in, the APDS construct wins: the next APX edition adopts it verbatim and deprecates its own (APX Part 3 §3.3(8)). APX is a proving ground for APDS, never a competitor.

Each request says what kind it is:

Kind	Meaning
Correction	The schema cannot validate what the document describes.
Omission	Something APDS describes, in the API document or the Data Dictionary, is missing from the API document.
Suggestion	A consistency or modelling idea. It may well be intentional as it stands.
Addition	New data APDS does not have today, proposed for a future edition.

We are very grateful to the Alliance for Parking Data Standards and to everyone who has worked on APDS. APX exists because APDS gave the industry a shared model to build on. We are glad to prepare pull requests or committee text for any item the group would like to take up.

Summary

Part	What it covers	Correction	Omission	Suggestion	Addition	Total
A	Corrections to the API document (errata 001–014, filed)	7	4	3	–	14
B	Data Dictionary items missing from, or typed differently in, the API document, and one request-body omission (candidates C-015–C-026, not yet filed)	–	12	–	–	12
C	New values for existing APDS enumerations	–	–	–	4	4
D	New data elements (classes, attributes, enumerations); the tier tables in Part D also repeat the 4 Part C rows	–	–	–	115	115
E	Other suggestions	–	–	5	–	5
	Total	**7**	**16**	**8**	**119**	**150**

Notes on the counts:

- Erratum 007 is counted as an Omission; it also carries a small consistency question. Erratum 014 is counted as a Suggestion; it is phrased as a question.
- Parts C and D together are the 127 proposals of the APX submission to the committee: 47 new classes, 11 additions to existing APDS classes, 57 new enumerations and 4 sets of new values for existing APDS enumerations.

Part A — Corrections to the APDS API document

Fourteen items, all filed on `parkingdata/spec` (#33–#35, #37–#47) with the relevant snippet and a minimal suggested fix. Each was checked against `parkingdata/spec` at commit `e10dcfc`, which is byte-identical to the published 4.1 document. Full reports are in the APX errata folder; the one-page version for the working group is Requested APDS changes.

#	Issue	Kind	Requested change	APX workaround today
001	#33 `Reference` requires `id` and `className` but allows at most one property	Correction	Remove `minProperties`/`maxProperties` from `Reference`	Validators relax the two constraints

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

#	Issue	Kind	Requested change	APX workaround today
002	#34 `POST /observations` `single-element` example matches neither `oneOf` branch	Correction	Add `id`/`version` to the example, or make them server-assigned	Lint exemption scoped to this one example
003	#35 Discriminator key spelled `ObvservationSet`	Correction	Rename to `ObservationSet` (third-party PR #36 is open)	None needed; tolerant tooling resolves it
004	#37 `RateTable` requires `validityStart` and `activeTimes`, which it never defines	Correction	Drop the two entries from `required`	Overlay replaces the list
005	#38 `POST /observations` declares no responses	Omission	Declare 201/400/409 like the other creates	Overlay declares them
006	#39 Wrapper `type` collides with `ObservationElement.type`	Correction	Rename the wrapper discriminator, or use a plain `oneOf`	Overlay uses a plain `oneOf`
007	#40 `POST /quotes` has no request body; `GET /quotes` returns one object	Omission (plus a question)	Declare the request body; confirm the `GET` shape	Overlay adds an optional body
008	#41 `ReferenceToQuote` is a `oneOf` of two identical branches	Correction	Make the branches distinct and non-empty	No workaround; book with the full `AssignedRight`
009	#42 `Identifiers` requires `rateTableId`, declares `rateTableID`	Correction	Align the spelling	Overlay declares `rateTableId`
010	#43 No entity declares the `extensions` container of Use Case §C.2.5	Suggestion	Declare `Extensions` once on `VersionedIdentity`	Overlay adds it to 9 entities
011	#44 `GeoJsonObject` has no `coordinates`	Omission	Declare `coordinates`, ideally with a `GeoJsonPoint`	Overlay adds `coordinates`
012	#45 Response codes on `/rates`, `/rights/assigned/{id}`, `/contacts` differ from their siblings	Suggestion	Align codes (201 on create; 400/404/409 declared)	Overlay adds the missing codes
013	#46 `HierarchyElement` maps 3 of 10 types; `VehicularAccess` unreachable	Omission	Restore the 7 commented-out mappings	Lane fields carried unvalidated
014	#47 No identifiable `RightHolder`; `AssignedRightHolder` has no `id`	Suggestion (question)	Add an identifiable `RightHolder`, or an optional `id`	`className: "RightHolder"` with a local id

APX Part 1 §1.3 (Known APDS 4.1 errata) records how APX treats each one today.

001 — `Reference` cannot be satisfied

`Reference` requires both `id` and `className` but declares `maxProperties: 1`. No object meets both rules, so a strict validator rejects every `Reference`, including those in the APDS examples. Suggested fix: remove `minProperties` and `maxProperties`.

002 — `POST /observations` example

The `single-element` request example omits the `id` and `version` that `ObservationElement` requires through `VersionedIdentity`, and it does not match `ObservationSet` either. Suggested fix: add `id` and `version` to the example, or say that the server assigns them.

003 — Discriminator typo

The `POST /observations` discriminator mapping spells its second key `ObvservationSet`, with an extra "v", so the value clients send, `ObservationSet`, has no entry. Suggested fix: correct the spelling. A third party has already proposed this one-character fix in PR #36.

004 — `RateTable` required list

`RateTable` lists `validityStart` and `activeTimes` in `required`, but neither is a property of `RateTable` or anything it inherits. It looks like the same kind of leftover from an earlier version that a maintainer confirmed in issue #32. Suggested fix: remove the two entries.

005 — `POST /observations` responses

As far as we can tell, this is the only APDS 4.1 operation with no `responses` member. A camera or PARCS vendor cannot tell whether success is 200, 201 or 202, or what comes back. Suggested fix: declare 201/400/409, as the other native

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

creates do.

006 — `POST /observations` type collision

The request wrapper types `type` as `ObservationDataType` (`ObservationElement`, `ObservationSet`), while `ObservationElement` types the same property as `CredentialTypeEnum`. Both apply, so no single observation validates. We may be missing something here. Suggested fix: rename the wrapper discriminator, or use a plain `oneOf`.

007 — `POST /quotes` request body

`POST /quotes` has no `requestBody`, so the `QuoteRightRequest` and `QuoteSessionExtensionRequest` schemas are never referenced. Separately, and only as a question: `GET /quotes` takes list filters but returns one object. Suggested fix: declare the request body, and confirm the intended `GET` shape.

008 — `ReferenceToQuote` branches

`ReferenceToQuote` is a `oneOf` of `ReferenceQuoteExtension` and `ReferenceQuoteNew`, which have the same properties and no `required` members. Every instance matches both, so `oneOf` always fails. Suggested fix: make the branches distinct, or merge them.

009 — `rateTableId` spelling

`Identifiers` declares `rateTableID` but requires `rateTableId`. JSON names are case-sensitive, so the required key is undeclared. Suggested fix: use `rateTableId`, matching the other APDS identifiers.

010 — The `extensions` container

The Use Case Document (§C.2.5) defines an `extensions` container with keys of the form `apds-ext:<namespace>:<class>@<major>.<minor>`, but no schema in the API document declares it. This may be deliberate, since open `additionalProperties` already lets it through. Suggestion: declare it once on `VersionedIdentity`, so the key pattern can be checked.

011 — `GeoJsonObject` coordinates

`GeoJsonObject` declares `type` and `bbox` but not `coordinates`, so a `Point` with no position still validates. This may be a deliberate simplification. Suggested fix: declare `coordinates`, and preferably a `GeoJsonPoint` for the many places that carry only a point.

012 — Response codes

Three native operations use different codes from their siblings: `POST /rates` answers 200 with no 400 or 409; `PUT /rights/assigned/{id}` declares 201 and no 404; the two `/contacts` reads alone declare 500. Each may be intentional. Suggestion: align them, or add a short note saying why they differ.

013 — `HierarchyElement` mapping

`HierarchyElementTypeEnum` has ten values, but the `HierarchyElement` discriminator maps only `campus`, `place` and `space`; the other seven are commented out. A `vehicularAccess` (lane) payload is therefore never checked against `VehicularAccess`. Suggested fix: restore the seven mappings, if they were not left out on purpose.

014 — An identifiable `RightHolder` (question)

The Information Model and the Data Dictionary describe `RightHolder` as an entity in its own right, but the API document has only `AssignedRightHolder`, embedded in each `AssignedRight` with no `id`. Two rights issued to the same holder cannot say so. Suggestion: add an identifiable `RightHolder`, or an optional `id`. See also C-018 below.

Part B — Differences between the APDS Data Dictionary and the API document

While checking each APX proposal against APDS 4.1, we compared the APDS 4.1 Data Dictionary (Version Release January 14, 2026) with the published API document. A few elements defined in the Data Dictionary do not yet appear in

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

`apds-api-4.1.yaml`. APX treats the Data Dictionary as authoritative and proposes nothing it already defines.

The request in this Part is to bring the API document in line with the Data Dictionary. We do not ask for any change to the Data Dictionary itself. All of these are **Candidate — not yet filed upstream**. They are numbered C-015 onward so they can follow errata 001–014 if the group would like them filed.

#	Data Dictionary	API document today	Requested change	APX workaround today
C-015	`RefillPointStatusEnum` (A.3.53, Table A.179) and the attribute `RefillPoint.status` (0..1)	Neither appears; `RefillPoint` has no `status`	Add the enumeration and the attribute	Code list `apx-device-states` follows its style and shares four of its values
C-016	`EquipmentTypeEnum` (A.3.23, Table A.149) includes `lprCamera`, `securityCamera`, `occupancySensor`, `occupancyObjectSensingCamera`, `door`, `doorWithAccessCredential`, `doorWithSecurityReview`, `emergencyExitDoor`, `cargoShippingDoor`, `dropOffLocation`	None of these ten values	Add the ten values	Device references point at `SupplementalEquipment` without an equipment type
C-017	`PaymentTypeEnum` (A.3.46, Table A.172): `base`, `refund`, `surcharge`, `validation`, `Vat`	`payment`, `validation`	Add `base`, `refund`, `surcharge` and `Vat`	`PaymentRecord` alongside `Payment`; code list `apx-payment-line-codes`
C-018	`RightHolder` is a class, stereotype `D2VersionedIdentifiable`, with `holderCredential` and `assignedRight` associations	Only `AssignedRightHolder`, with no `id`	Add `RightHolder` as a versioned identifiable class	`className: "RightHolder"` with a local id (see 014)
C-019	`ServiceFacilityTypeEnum` (A.3.59, Table A.185) includes `currencyExchange` and `transportInterchange`	Neither value	Add the two values	None needed
C-020	`RateLineTypeEnum` (A.3.50, Table A.176) includes `perUnit`	`flatRate`, `flatRateTier`, `incrementingRate`	Add `perUnit`	None needed
C-021	`ElementDescriptorEnum` (A.3.20, Table A.146) includes `garage`	`floorOrLevel`, `row`, `street`	Add `garage`	None needed
C-022	`CredentialTypeEnum` (A.3.17, Table A.143) includes `other`	No `other`	Add `other`	None needed
C-023	`UnitOfMeasureEnum` (A.3.70, Table A.194) includes `kilogrammes`, `pounds` and `gpmi`	No `kilogrammes` or `pounds`; has `gpm` where the Data Dictionary has `gpmi`	Add `kilogrammes` and `pounds`; align `gpm`/`gpmi`	None needed
C-024	Accessibility enumeration (A.3.2, `AccessibilityEnum`) has `accessForPersonsWithDisabilities` and `disabledMarked`	`AccessibilityTypeEnum` has `disabledAccessible` and `disabledMarkedThere` instead	Align the two values	None needed
C-025	`RightSpecification.transferableConditions` reads as `PkCommon.MultilingualString` (tentative; see below)	`boolean`	Confirm the type, and align the API document if a statement was intended	None needed
C-026	`ObservationElement` has an `observationElementRecordUpdate` association (Table A.46), so an Observation can be updated	`PUT /observations/{id}` declares no request body and no 400	Declare an `ObservationElement` request body and a 400, as on the other `PUT` operations	The data overlay declares both

C-015 — Refill point status

The Data Dictionary's list of changes from 4.0 to 4.1 includes "Bug correction - Attribute "status" added to the class "RefillPoint"", typed `RefillPointStatusEnum` (13 values, from `available` to `unknown`). The API document's `RefillPoint` schema has no `status`, and the enumeration is absent. APX's proposed energy additions (Part D) reuse this enumeration as it stands, so having it in the API document would help.

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

C-016 — Equipment type values

The Data Dictionary already gives camera, sensor and door values for `EquipmentTypeEnum`. The API document lists the older set only. With these values in place, the four lane-equipment values APX suggests in Part C would complete a typical access lane.

C-017 — Payment type values

The API document's `payment` and the Data Dictionary's `base` carry the same definition ("Indicates financial payment made"), so they may be two names for one value. The committee will know which name it prefers. The API document's own `PaymentLine` description already mentions "base fee, taxes, surcharges".

C-018 — RightHolder

This is the Data Dictionary side of erratum 014. The Data Dictionary defines `RightHolder` as versioned identifiable; the API document embeds the holder without an identifier. Bringing the API document in line would answer 014.

C-025 — Type of `transferableConditions` (tentative)

In the Data Dictionary's `RightSpecification` attribute table, "Transferable conditions" is defined as a "statement of the conditions under which a RightSpecification may be transferred", which reads as `PkCommon.MultilingualString`, while `transferable` is the `PkCommon.Boolean`. The API document types both `transferable` and `transferableConditions` as boolean. The columns of that table are misaligned in the PDF, so we would be grateful if the group could confirm the intended types before anything is filed.

C-026 — Request body for `PUT /observations/{id}`

Unlike the other candidates, this one comes from the API document alone. `PUT /observations/{id}` is described as "update the given Observation", but it declares no request body, so a client cannot tell from the document what to send. It also declares no 400, which every other `PUT` does. It is the same kind of omission as erratum 007 (`POST /quotes`). APX declares an optional `ObservationElement` body and the 400 in its data overlay, leaving the vendored document untouched.

C-019 to C-024 — Smaller value differences

These were found by a script that compares every enumeration table in Data Dictionary section A.3 with the `enum` arrays of the API document, then checked by hand against the Data Dictionary text. In C-024, the Data Dictionary's own definition of `barrierFreeAccessible` still mentions `disabledAccessible`, so the intended names may simply need confirming.

Not requested

- **Values in the API document but not the Data Dictionary.** A few run the other way: for example `HierarchyElementTypeEnum` (five more values in the API document) and `UnitOfTimeEnum` (`year`, `month`, `week`). We do not ask for any of these to be removed.
- **Enumerations under other names.** Several Data Dictionary enumerations appear in the API document under a different name (for example `EnergySourceEnum` as `FuelTypeEnum`, `RightTypeEnum` as `RightSpecificationTypeEnum`). We do not ask for renames.
- **Classes the API document does not carry.** Some Data Dictionary enumerations belong to classes absent from the API document (for example `SurchargeTypeEnum` and `RefundTypeEnum`, used by `Surcharge`). We have not compared these one by one and make no request about them.

Part C — New values for existing APDS enumerations

Four existing APDS enumerations would gain new values. Each value is a new name only; none repeats or redefines an existing value. These are the `change: extend` enumerations in `spec/proposals/apds-additions/`, and are part of the 127 proposals.

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

Enumeration	Domain	New values	Tier	Why	APX today
`EquipmentTypeEnum`	Place	`barrier`, `gate`, `ticketDispenser`, `intercom`	1	Off-street sites could type their lane equipment with APDS names. With the Data Dictionary values in C-016, these four complete a typical access lane.	Device references (`DeviceStatus.device`, `Command.target`, `Alert.source.device`) point at `SupplementalEquipment` without an equipment type (Part 6 §6.4)
`ImageTypeEnum`	Observation	`plateExtended`, `context`	2	Recognition engines keep a wider crop around the plate and a scene wider than the overview; consumers could ask for the still they need.	Images listed untyped on `Observation.images[]`, typed as strings on `LprRead.images[]` (Part 13 §13.3c)
`RightTypeEnum` (API: `RightSpecificationTypeEnum`)	Right	`valet`	2	Lets any APDS consumer, such as an enforcement vendor, tell a vehicle in valet custody from a transient parker.	The valet pass is typed `oneTimeUseParking`; clients learn it is valet from `ValetTicket.assignedRight` (Part 22)
`PaymentTypeEnum`	Right	`convenienceFee`, `serviceFee`, `discount`, `energy`, `idle`	2	Payment lines could tell fees, discounts and charging costs apart without a bilateral code list.	Code list `apx-payment-line-codes` for `PaymentLine.idCode` (Part 13 §13.6a)

The value definitions:

- **`barrier`:** a lifting arm or similar barrier that opens and closes a vehicular access lane.
- **`gate`:** a sliding, swinging or rolling gate or door that opens and closes a vehicular access point.
- **`ticketDispenser`:** a machine at an entry lane that issues a ticket or other credential to start a Session.
- **`intercom`:** a call point at a lane or machine that connects a customer with an attendant by voice or video.
- **`plateExtended`:** a crop around the plate wider than the plate itself, showing the surrounding bodywork.
- **`context`:** a wide view of the scene around the vehicle, wider than the overview.
- **`valet`:** intended for a vehicle placed in valet custody, valid from drop-off to handback.
- **`convenienceFee`:** a fee for the convenience of the channel used to pay. **`serviceFee`:** a fee for a service other than the right itself. **`discount`:** a reduction in the amount charged, other than a validation. **`energy`:** a charge for energy delivered to the vehicle. **`idle`:** a charge for occupying a charging bay after charging ended.

The `PaymentTypeEnum` values would sit beside the Data Dictionary values `base`, `refund`, `surcharge`, `validation` and `Vat` (see C-017).

Part D — New data elements

The proposals are grouped in three tiers so the working groups can take them in steps. **Tier 1, propose now,** is data every access control, enforcement or LPR system already holds and needs to share: access events, equipment status, credential status, how a session started and ended, plate read attributes, violations and notices, and validation redemptions. **Tier 2, proven in APX,** covers shapes that are stable in APX and widely useful, such as plate lists, validation programmes, payment outcomes, reservations and permits, valet custody and alerts. **Tier 3, for discussion,** covers areas that are still settling, such as electric vehicle charging sessions, or that may sit outside parking data, such as customer service interactions, or the comms points through which a parker calls for help. Of the 127 proposals, 34 are Tier 1, 65 Tier 2 and 28 Tier 3.

Additions to existing APDS classes are written as Level B extensions: they add optional content only, so a valid extended instance is always a valid APDS 4.1 instance. Two new domains, **Enforcement** and **Operations**, introduce new root

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

concepts and are marked **Level C**. The full committee-format text, mirroring the APDS 4.1 Overview, Information Model, Data Dictionary and Use Case Document, is in the APX submission to APDS. The tables below are generated from `spec/proposals/apds-additions/*.yaml`, the same source the submission is built from.

127 data elements are proposed. The full text of each, in the APDS Data Dictionary format, is in `docs/apds-submission/03-data-dictionary.md`.

Tier 1: propose now (34)

Element	Change	APDS domain	Package	Comes from in APX
`Violation`	New class	Enforcement	Enforcement	APX Part 19 §19.1 Violation
`ViolationDetection`	New class	Enforcement	Enforcement	APX Part 19 §19.4 ViolationDetection
`Notice`	New class	Enforcement	Enforcement	APX Part 19 §19.1–19.2 Violation.notice
`ViolationTypeEnum`	New enumeration	Enforcement	Enforcement	APX Part 11 code list (violation types)
`ViolationStatusEnum`	New enumeration	Enforcement	Enforcement	APX Part 19 §19.1 Violation.violationStatus
`DetectionModeEnum`	New enumeration	Enforcement	Enforcement	APX Part 19 §19.4 ViolationDetection.mode
`NoticeKindEnum`	New enumeration	Enforcement	Enforcement	APX Part 19 §19.1–19.2 Violation.notice.noticeKind
`NoticeDeliveryMethodEnum`	New enumeration	Enforcement	Enforcement	APX Part 19 §19.1–19.2 Violation.notice.deliveryMethod
`ObservationElement`	Additions to an APDS class	Observation	Observation	APX Part 13 §13.3a LprReadDetail
`AttributeRead`	New class	Observation	Observation	APX Part 13 §13.3a AttributeRead
`AlternateRead`	New class	Observation	Observation	APX Part 13 §13.3a LprReadDetail.alternateReads
`ReadAttributeEnum`	New enumeration	Observation	Observation	APX Part 13 §13.3a LprReadDetail
`VehicleMovementEnum`	New enumeration	Observation	Observation	APX Part 13 §13.3a LprReadDetail.movement
`InteractionChannelEnum`	New enumeration	PkCommon	CommonClasses	APX Parts 13, 17, 20 and 22 channel values
`EquipmentStatus`	New class	Place	SupplementalFacility	APX Part 6 §6.4 DeviceStatus
`EquipmentStatusEnum`	New enumeration	Place	SupplementalFacility	APX Part 11 code list (device states)
`EquipmentTypeEnum`	New values for an APDS enumeration	Place	SupplementalFacility	APX Part 6 §6.4 device references
`CredentialAssigned`	Additions to an APDS class	Right	Right	APX Part 21 §21.1 CredentialRecord
`CredentialMedia`	New class	Right	Right	APX Part 21 §21.1 CredentialRecord.media
`CredentialSuspension`	New class	Right	Right	APX Part 21 §21.1 CredentialRecord.suspension
`PresentedCredential`	New class	Right	Right	APX Part 21 §21.4 PresentedCredential
`CredentialStatusEnum`	New enumeration	Right	Right	APX Part 21 §21.1 CredentialRecord.credentialStatus
`CredentialMediaFormEnum`	New enumeration	Right	Right	APX Part 21 §21.1 CredentialRecord.media.form
`DepositStatusEnum`	New enumeration	Right	Right	APX Part 21 §21.1 CredentialRecord.media.depositStatus
`Session`	Additions to an APDS class	Session	Session	APX Part 5 §5.8 SessionDetail
`SessionRead`	New class	Session	Session	APX Part 5 §5.8 SessionReadBinding
`SessionClosure`	New class	Session	Session	APX Part 5 §5.8 SessionClosure
`AccessEvent`	New class	Session	Session	APX Part 21 §21.4 AccessEvent
`ValidationRedemption`	New class	Session	Session	APX Part 20 ValidationRedemption
`SessionReadRoleEnum`	New enumeration	Session	Session	APX Part 5 §5.8 SessionReadBinding.role
`SessionEndReasonEnum`	New enumeration	Session	Session	APX Part 11 code list (session end reasons)
`AccessOutcomeEnum`	New enumeration	Session	Session	APX Part 21 §21.4 AccessEvent.outcome

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

Element	Change	APDS domain	Package	Comes from in APX
`AccessDenialReasonEnum`	New enumeration	Session	Session	APX Part 11 code list (access denial reasons)
`ValidationRedemptionStatusEnum`	New enumeration	Session	Session	APX Part 20 ValidationRedemption.redemptionStatus

Tier 2: proven in APX, propose next (65)

Element	Change	APDS domain	Package	Comes from in APX
`Appeal`	New class	Enforcement	Enforcement	APX Part 19 §19.6 Violation.appeal
`EnforcementPolicy`	New class	Enforcement	Enforcement	APX Part 19 §19.10 EnforcementPolicy
`NoticeDeliveryRule`	New class	Enforcement	Enforcement	APX Part 19 §19.10 EnforcementPolicy.deliveryRules
`EscalationStep`	New class	Enforcement	Enforcement	APX Part 19 §19.10 EnforcementPolicy.escalation
`AppealReasonEnum`	New enumeration	Enforcement	Enforcement	APX Part 19 §19.6 Violation.appeal.reason
`AppealResolutionEnum`	New enumeration	Enforcement	Enforcement	APX Part 19 §19.6 Violation.appeal.resolution
`ReadReview`	New class	Observation	Observation	APX Part 13 §13.3a LprReadDetail.review
`Image`	Additions to an APDS class	Observation	Observation	APX Part 13 §13.3a LprReadDetail.plateBox
`ImageRegion`	New class	Observation	Observation	APX Part 13 §13.3a LprReadDetail.plateBox
`ConditionReport`	New class	Observation	Observation	APX Part 22 ConditionReport
`DamageItem`	New class	Observation	Observation	APX Part 22 ConditionReport.damage
`ImageTypeEnum`	New values for an APDS enumeration	Observation	Observation	APX Part 13 §13.3c LprRead.images[].imageType
`PlateFaceEnum`	New enumeration	Observation	Observation	APX Part 13 §13.3a LprReadDetail.plateFace
`ReadReviewStateEnum`	New enumeration	Observation	Observation	APX Part 13 §13.3a LprReadDetail.review
`ReadReviewReasonEnum`	New enumeration	Observation	Observation	APX Part 11 code list (read review reasons)
`DamageSeverityEnum`	New enumeration	Observation	Observation	APX Part 22 ConditionReport.damage
`TollTransaction`	New class	Observation	Tolling	APX Part 15 TollTransaction
`TollDispute`	New class	Observation	Tolling	APX Part 15 §15.1 TollTransaction.dispute
`TollTransactionStatusEnum`	New enumeration	Observation	Tolling	APX Part 15 TollTransaction.transactionStatus
`TollDisputeReasonEnum`	New enumeration	Observation	Tolling	APX Part 11 code list (toll dispute reasons)
`TollDisputeResolutionEnum`	New enumeration	Observation	Tolling	APX Part 11 code list (toll dispute resolutions)
`Alert`	New class	Operations	Operations	APX Part 7 §7.1 Alert
`AlertTypeEnum`	New enumeration	Operations	Operations	APX Part 11 code list (alert types)
`AlertSeverityEnum`	New enumeration	Operations	Operations	APX Part 7 Alert.severity
`AlertStatusEnum`	New enumeration	Operations	Operations	APX Part 7 §7.2 Alert.status
`RecordEndReasonEnum`	New enumeration	PkCommon	CommonClasses	APX Parts 19, 20 and 24 status values
`VehicularAccess`	Additions to an APDS class	Place	VehicularAccess	APX Part 13 §13.3a LaneCameras
`LaneCamera`	New class	Place	VehicularAccess	APX Part 13 §13.3a LaneCameras.cameras
`CameraFacingEnum`	New enumeration	Place	VehicularAccess	APX Part 13 §13.3a LaneCameras
`Signage`	New class	Place	Signage	APX Part 19 §19.11 Signage
`SignTypeEnum`	New enumeration	Place	Signage	APX Part 19 §19.11 Signage.signType
`SupplementalEquipment`	Additions to an APDS class	Place	EquipmentDetail	APX Part 26 §26.1 EquipmentDetail (apds-ext:apx:equipment-detail@1.0)
`Camera`	New class	Place	EquipmentDetail	APX Part 26 §26.3 CameraDetail (apds-ext:apx:camera@1.0)
`EquipmentRoleEnum`	New enumeration	Place	EquipmentDetail	APX Part 11 code list apx-equipment-roles (Part 26 §26.1)

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

Element	Change	APDS domain	Package	Comes from in APX
`AreaTypeEnum`	New enumeration	Place	EquipmentDetail	APX Part 11 code list apx-area-types (Part 26 §26.2)
`CameraMountingEnum`	New enumeration	Place	EquipmentDetail	APX Part 26 §26.3 CameraDetail.mounting
`CameraViewEnum`	New enumeration	Place	EquipmentDetail	APX Part 26 §26.3 CameraDetail.view
`RateTable`	Additions to an APDS class	Rates	Rates	APX Part 6 §6.6 RatePolicy
`ValidationBenefit`	New class	Rates	Rates	APX Part 20 §20.1 ValidationBenefit
`AssignedRight`	Additions to an APDS class	Right	Right	APX Part 14 §14.1 ReservationExtension
`OverageBilling`	New class	Right	Right	APX Part 14 §14.2d PermitExtension.overageBilling
`RightHolder`	Additions to an APDS class	Right	Right	APX Part 13 §13.1 Account
`Payment`	Additions to an APDS class	Right	Right	APX Part 13 §13.1a PaymentRecord
`ReservationStateEnum`	New enumeration	Right	Right	APX Part 14 §14.1 ReservationExtension.reservationState
`OverageBillingModeEnum`	New enumeration	Right	Right	APX Part 14 §14.2d PermitExtension.overageBilling.mode
`AccountStatusEnum`	New enumeration	Right	Right	APX Part 13 §13.1 Account.accountStatus
`PaymentStatusEnum`	New enumeration	Right	Right	APX Part 13 §13.1a PaymentRecord.paymentStatus
`CaptureStatusEnum`	New enumeration	Right	Right	APX Part 13 §13.1a PaymentRecord.captureStatus
`RightTypeEnum`	New values for an APDS enumeration	Right	Right	APX Part 22 ValetTicket.assignedRight
`PaymentTypeEnum`	New values for an APDS enumeration	Right	Right	APX Part 13 §13.6a payment line codes
`ValidationProgram`	New class	Right	Validation	APX Part 20 §20.1 ValidationProgram
`ValidationRules`	New class	Right	Validation	APX Part 20 §20.3 ValidationProgram.rules
`ValidationIssuance`	New class	Right	Validation	APX Part 20 §20.2 ValidationIssuance
`ValidationInstrument`	New class	Right	Validation	APX Part 20 §20.2 ValidationInstrument
`ValidationIssuanceMethodEnum`	New enumeration	Right	Validation	APX Part 20 §20.2 ValidationIssuance.method
`ValidationIssuanceStatusEnum`	New enumeration	Right	Validation	APX Part 20 §20.2 ValidationIssuance.issuanceStatus
`ValidationInstrumentStatusEnum`	New enumeration	Right	Validation	APX Part 20 §20.2 ValidationInstrument.instrumentStatus
`PlateList`	New class	Right	PlateList	APX Part 24 §24.1 PlateList
`PlateListPlaceFlags`	New class	Right	PlateList	APX Part 24 §24.1 PlateListPlaceFlags
`PlateListEntry`	New class	Right	PlateList	APX Part 24 §24.2 PlateListEntry
`PlateListCategoryEnum`	New enumeration	Right	PlateList	APX Part 11 code list (plate list categories)
`ValetTicket`	New class	Session	Session	APX Part 22 ValetTicket
`SessionLifecycleEnum`	New enumeration	Session	Session	APX Part 5 §5.8 SessionDetail.lifecycle
`ValetStatusEnum`	New enumeration	Session	Session	APX Part 22 ValetTicket.valetStatus
`HandbackVerificationEnum`	New enumeration	Session	Session	APX Part 22 ValetTicket.handback

Tier 3: for discussion (28)

Element	Change	APDS domain	Package	Comes from in APX
`ChargingPointStatus`	New class	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 §23.1 ChargingPointStatus
`ConnectorStatus`	New class	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 §23.1 ChargingPointStatus.connectors
`ChargingBayPresence`	New class	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 §23.3 ChargingBayPresence

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

Element	Change	APDS domain	Package	Comes from in APX
`ChargingSession`	New class	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 §23.2 ChargingSession
`ChargingCost`	New class	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 §23.5 ChargingSession.cost
`ChargingIdleTerms`	New class	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 §23.2, §23.5 ChargingSession.idlePolicy
`BayPresenceEnum`	New enumeration	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 §23.3 ChargingBayPresence.presence
`BayPresenceSourceEnum`	New enumeration	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 §23.3 ChargingBayPresence.source
`ChargingSessionStatusEnum`	New enumeration	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 §23.2 ChargingSession.chargingStatus
`ChargingStopReasonEnum`	New enumeration	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 ChargingSession.stopReason
`ChargingSettlementModeEnum`	New enumeration	EnergyInfrastructure	EnergyInfrastructure	APX Part 23 §23.5 ChargingSession.settlement.mode
`PenaltyLimit`	New class	Enforcement	Enforcement	APX Part 19 §19.10 EnforcementPolicy.penaltyCap
`DemandType`	Additions to an APDS class	Occupancy	Demand	APX Part 5 §5.5a OccupancyHistoryPage
`SupportInteraction`	New class	Operations	Operations	APX Part 17 §17.6 SupportInteraction
`SupportAgentTypeEnum`	New enumeration	Operations	Operations	APX Part 17 SupportInteraction.agentType
`SupportIssueTypeEnum`	New enumeration	Operations	Operations	APX Part 11 code list (issue types)
`RecordType`	Additions to an APDS class	PkCommon	CommonClasses	APX Part 18 §18.6 RecordInfo.origin
`CommsPoint`	New class	Place	CommsPoint	APX Part 26 §26.4 CommsPoint
`CommsPointTypeEnum`	New enumeration	Place	CommsPoint	APX Part 11 code list apx-comms-point-types (Part 26 §26.4)
`RateSold`	New class	Right	Right	APX Part 13 §13.6a PaymentDetail.rateSold
`RateProductEnum`	New enumeration	Right	Right	APX Part 13 §13.6a PaymentDetail.rateSold.product
`RateClassEnum`	New enumeration	Right	Right	APX Part 13 §13.6a PaymentDetail.rateSold.class
`ValidationBilling`	New class	Right	Validation	APX Part 20 §20.1 ValidationProgram.billing
`ValidationBillingModelEnum`	New enumeration	Right	Validation	APX Part 20 §20.1 ValidationProgram.billing.model
`ValidationBillingCycleEnum`	New enumeration	Right	Validation	APX Part 20 §20.1 ValidationProgram.billing.billingCycle
`MatchEvidence`	New class	Session	Session	APX Part 5 §5.8 MatchEvidence
`MatchTypeEnum`	New enumeration	Session	Session	APX Part 5 §5.8 MatchEvidence.matchType
`MatchDecisionEnum`	New enumeration	Session	Session	APX Part 5 §5.8 MatchEvidence.decidedBy

Part E — Other suggestions

These are not errata and not additions to the data model. Each is a **Suggestion**, offered for the committee to consider. Where APDS covers any of them, APX moves to the APDS form and retires its own.

#	Suggestion	Why it helps	APX today
E-1	Make `VehicleAncillaryIdentification.country` optional (`0..1` instead of `1..1`)	A read whose plate country the engine could not place could still carry make, model and colour, often the reads that most need them	The attributes ride in the APX `detail` decoration and the APDS object is omitted (Part 13 §13.3a(1))
E-2	Agree the names for two equipment states: `faulted` and `closed`	APX and APDS would use the same words for the same state	Code list `apx-device-states` uses `fault` and `inoperative`
E-3	Event types for access and corrections in `EventTypeEnum`	Subscribers could hear a lane denial or a corrected read through APDS events	APX topics `apx.credentials.access.v1` and `apx.data.observation.updated.v1` (Parts 8, 21)

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

#	Suggestion	Why it helps	APX today
E-4	`application/problem+json` (RFC 9457) as an error option on native routes	One machine-readable error shape across APDS and APX	Offered when the client sends `Accept: application/problem+json` (Part 5 §5.1a, Part 12)
E-5	A member on `Payment` naming the rate sold	A receiver could tell which rate a payment bought	Decoration `apds-ext:apx:payment-detail@1.0` with `rateSold` (Part 13 §13.6a)

E-1 — `VehicleAncillaryIdentification.country`

This is the one suggestion that changes an existing multiplicity, so it is kept outside the Level B additions in Part D and offered separately. Today a read with no country must omit the whole `VehicleAncillaryIdentification`. The change relaxes a constraint, so every instance valid today stays valid. See the Observation section of the submission's Information Model and Requested APDS changes.

E-2 — Equipment state names

The proposed `EquipmentStatusEnum` (Place, Part D) reuses `RefillPointStatusEnum` names where the meaning is the same: `available`, `occupied`, `blocked`, `faulted` and `unknown`. It uses `closed` for equipment not operating for a planned reason, such as closed hours or a maintenance window. APX's current code list says `fault` and `inoperative` for these two states; APX would adopt whatever names APDS agrees. For reference, `RefillPointStatusEnum` defines `inoperative` as "not yet active or it is no longer available (deleted)". The committee may prefer either pair of names.

E-3 to E-5 — Interface and payment detail

These came from building APX on the APDS routes. E-3 and E-4 concern the API rather than the data model, so they are not in the submission, which proposes data only. E-5 is the one part of the payment-detail suggestion in Requested APDS changes not already covered by Part C (`PaymentTypeEnum`) or Part D (`Payment` outcomes).

Open questions for the committee

1. **Enumerations or code lists for open vocabularies.** APX proposes enumerations that list common values and allow `other`, extendable by an external code list, as APDS already does for vehicle types and occupancy levels. Where values vary by market (violation types, end reasons, device states), would the group prefer a `ReferencedCodeListEntry` seeded with the values instead?
2. **Customer service interactions.** `SupportInteraction` (Operations, Tier 3) may lie outside the scope APDS sets itself in its guiding principles. It is included so the group can decide. APX is equally happy to keep it in its own profile.
3. **Tier order.** Do the tiers suit the working groups' calendar, or would the group rather take a domain at a time?
4. **Filing the Part B candidates.** Would the group like C-015 to C-026 filed as issues on `parkingdata/spec`, as one combined issue, or handled within the Data Dictionary process?
5. **Payment type names.** In C-017, are `payment` (API document) and `base` (Data Dictionary) the same value, and which name should stay?
6. **Batching the errata.** Requested APDS changes suggests an order: first the items that block validation today (001, 004, 006, 013), then one-line fixes, then missing declarations, then modelling questions. Is that order helpful?

What APX does in the meantime

APX does not wait for any of these requests. It keeps working on APDS 4.1 as published:

- **The vendored file is never edited.** `apds-api-4.1.yaml` is vendored byte-identical and checksum-guarded (Part 0 §0.2, Part 1 §1.1).
- **Errata are handled in an overlay.** Each Part A workaround lives in `spec/openapi/overlays/apx-data-overlay.yaml` or in APX tooling, narrowly, with a comment naming the erratum. Part 1 §1.3 lists them.
- **Missing data rides in extensions.** Where APDS has no member, APX uses the APDS `extensions` container with namespaced, versioned keys (`apds-ext:apx:<class>@<M.m>`), for example `session-detail@1.0` and `payment-detail@1.0` (Part 10 §10.1).

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS

- **Open values live in registries.** APX code lists are published as APDS UserDefinedCodeList documents in `spec/registries/`, for example `apx-device-states`, `apx-session-end-reasons` and `apx-payment-line-codes` (Part 11).
- **New surfaces live in APX modules.** Alerts, lane control, tolling, valet and enforcement are APX conformance classes. Their data elements are proposed in Part D; the routes and workflows stay in the APX API.

When APDS adopts any request, APX vendors that release and removes the matching overlay action, extension or registry entry in the same change, so the two documents never disagree (Part 3 §3.3(8)).

Thank you again to the APDS committee for considering these requests.

© 2026 Umojo, Inc. and APX contributors (MIT): APX Requests to APDS