National Parking Platform

Connected Supplier Developer Guide

Version 1.0

Publication pending

National Parking Platform

Contents

Introduction

This document should be read with the NPP API Specification. Whilst the API Specification serves as an interface reference, the NPP Connected Supplier Developer Guide provides guidance for developers based on concrete use cases. The use cases in this document relate to you specific roles as a Connected (Enforcement System) Supplier.
e in mind.

> *This document is a working document. It will be updated when new use cases are introduced or necessary changes are recognised (e.g. additional details and explanations based on developer feedback/questions).*

Use Case Overview

This section provides an overview of the use cases covered in this version of the NPP Developer Guide. Use it as a springboard to the use cases relevant to you.

1. Connected Supplier Use Cases

In this first version of the document, enforcement system providers are the only category of connected suppliers.

1.1 Check Parking Right (Fetch)

The enforcement system actively fetches parking right information.

1.2 Parking Right (Push Notification)

The enforcement system (subscribed to the NPP push service) receives a notification of a new or updated parking right.

1.3 Sessions (Fetch)

The enforcement system actively fetches session information.

1.4 Sessions (Push Notification)

The enforcement system (subscribed to the NPP push service) receives a notification of a new or updated session.

2. Shared Use Cases

2.1 Read List of known Organisations on NPP

An NPP user reads the list of organisations (contacts) known to the NPP.

Use Case Details

This section provides detailed examples for all current use cases.

Preliminary Remarks

The examples in this session presume the existence of

- one operator OPERATOR1 representing city council COUNCIL1
- two ficticious service providers *PROVIDER1* and *PROVIDER2*
- two off-street parking locations *CARPARK1* and *CARPARK2*
- two on-street parking locations *STREET1* and *STREET2*

For better readability, we often use telling ids instead of UUID-type ids.

The HTTP requests shown in this document are presumed to include all mandatory headers, in particular:

- Authorization: Bearer *{valid access token}*
- Accept: application/json
- Content-type: application/json
- X-Client-Version: *{your application version}*

1. Connected Supplier Use Cases

In this first version of the document, enforcement system providers are the only category of connected suppliers.

1.1 Check Parking Rights (Pull Mode)

The enforcement system actively fetches parking right information. The search results can be narrowed down using appropriate filter criteria. For enforcement purposes, this will typically be the following query parameters:

- place - the location id of the parking location currently being monitored
- credential_id - the VRM of the vehicle being checked
- end_after - the earliest expiration timestamp of the rights to be returned

The NPP will return a paginated list of all matching assigned right records. Below is a sample request and response:

```
GET /v4/parking/rights/assigned
      ?place=CARPARK1&credential_id=TST001&end_after=1750166111
```

(query parameters shown in the second line for better readability)

```
{
      "meta": {
            "referenceInstant": 1750166111,
            "offset": 0,
            "pageSize": 200,
            "total": 1
      },
      "data": [

            {
                  “id”: “d0a16d11-9804-4c9e-bba2-99a8a6d95dfb”,
                  “version”: 1,
                  “rightHolder”: {
                        “credentials”: [
                              {
                                    “type”: “licensePlate”,
                                    “identifier”: {
                                          “id”: “TST001”, // the right holders VRM
                                          “className”: “UKNumberPlate”
                                    },
                                    “issuer”: [
                                          {
                                                “language”: “en”,
                                                “string”: DVLA”
                                          }
                                    ]
                              }
                        ]
                  },
                  “assignedRightIssuer”: {
                        “id”: “PROVIDER1”,          // the id of the service provider
                        “version”: 1
                  },
```

```
                        “rightSpecification”: {
                                “id”: “CARPARK1-RIGHT1”,
                                “version”: 1,
                                "hierarchyElements": [
                                        {
                                                "id": "7591001",
                                                "version": 1,
                                                "name": [
                                                        {
                                                                "language": "en",
                                                                "string": "Lord Street"
                                                        }
                                                ]
                                        },
                                        {
                                                "id": "7582442",
                                                "version": 1,
                                                "name": [
                                                        {
                                                                "language": "en",
                                                                "string": "Victoria Street"
                                                        }
                                                ]
                                        }
                                ],
                        },
                        “issuanceTime”: “2025-05-20T10:02:00Z”, // the time the right became valid
                        “expiry”: “2025-05-20T11:02:00Z”, // the time the right expires
                        “issueMethod”: “electronic”,
                        “monetaryValue”: {
                                “taxIncluded”: true,
                                “value”: {
                                        “currencyType”: “GBP”,
                                        “currencyValue”: 2
                                }
                        },
                        “payments”: [
                                {
                                        “id”: “23d45a6d-947b-4932-ac8b-0d97ec3a328d”,
                                        “version”: 1,
                                        “dateCollected”: “2025-05-20T10:01:00Z”,
                                        “startPeriodCovered”: “2025-05-20T10:02:00Z”,
                                        “endPeriodCovered”: “2025-05-20T11:02:00Z”,
                                        “paymentLines”: [
                                                {
                                                        “id”:“403511a6-3e79-40ab-851b-da45b4ea8b34”,
                                                        “version”: 1,
                                                        “value”: {
                                                                “currencyType”: “GBP”,
                                                                “currencyValue”: 2
                                                        },
                                                        “idCode”: “PARKING-FEE-VAT20”,
                                                        “identifierId”:“PROVIDER1-PAYMENT-REF-0001”,
                                                        “paymentType”: “payment”
                                                }
                                        ]
                                }
                        ]
                }
        ]
}
```

The *end_after* query parameter can also be used to for instance query recently expired assigned rights. Timestamps in query parameters are always specified as epoch seconds.

Important note: per the specification, the referenced right specification(s) only provide a reference to the applicable hierarchy elements (parking locations), i.e. id and version. As a convenience feature, it is possible to activate the additional provision of the location name, helping the caller to avoid another call to the inventory API. Please indicate this to the NPP team in case you want this option activated.

1.2 Parking Right (Push Notification)

The enforcement system (subscribed to the NPP push service) receives a notification of a new or updated parking right. The contents is the same as in the previous example, but the information will be pro-actively pushed to the system's webhook endpoint. The custom http request header named "X-Event-Type" will indicate the nature of the data event.

```
POST {provider-specified webhook endpoint}
X-Event-Type: AssignedRightCreated
```


```
{
	"id": "d0a16d11-9804-4c9e-bba2-99a8a6d95dfb",
	"version": 1,
	"rightHolder": {
		"credentials": [
			{
				"type": "licensePlate",
				"identifier": {
					"id": "TST001",
					"className": "UKNumberPlate"
				},
				"issuer": [
					{
						"language": "en",
						"string": DVLA"
					}
				]
			}
		]
	},
	"assignedRightIssuer": {
		"id": "PROVIDER1",
		"version": 1
	},
	"rightSpecification": {
		"id": "CARPARK1-RIGHT1",
		"version": 1,
		"hierarchyElements": [
			{
				"id": "7591001",
				"version": 1,
				"name": [
					{
						"language": "en",
						"string": "Lord Street"
					}
				]
			},
			{
				"id": "7582442",
				"version": 1,
				"name": [
					{
						"language": "en",
						"string": "Victoria Street"
					}
				]
			}
		],
	},
	"issuanceTime": "2025-05-20T10:02:00Z",
	"expiry": "2025-05-20T11:02:00Z",
	"issueMethod": "electronic",
```

```
        “monetaryValue”: {
                “taxIncluded”: true,
                “value”: {
                        “currencyType”: “GBP”,
                        “currencyValue”: 2
                }
        },
        “payments”: [
                {
                        “id”: “23d45a6d-947b-4932-ac8b-0d97ec3a328d”,
                        “version”: 1,
                        “dateCollected”: “2025-05-20T10:01:00Z”,
                        “startPeriodCovered”: “2025-05-20T10:02:00Z”,
                        “endPeriodCovered”: “2025-05-20T11:02:00Z”,
                        “paymentLines”: [
                                {
                                        “id”: “403511a6-3e79-40ab-851b-da45b4ea8b34”,
                                        “version”: 1,
                                        “value”: {
                                                “currencyType”: “GBP”,
                                                “currencyValue”: 2
                                        },
                                        “idCode”: “PARKING-FEE-VAT20”,
                                        “identifierId”: “PROVIDER1-PAYMENT-REF-0001”,
                                        “paymentType”: “payment”
                                }
                        ]
                }
        ]
}
```

1.3 Sessions (Pull Mode)

The enforcement system actively fetches session information. The search results can be narrowed down using appropriate filter criteria. For enforcement purposes, this will typically be the following query parameters:

- place - the location id of the parking location currently being monitored
- credential_id - the VRM of the vehicle being checked
- end_after - the earliest expiration timestamp of the rights to be returned

The NPP will return a paginated list of all matching session records. Below is a sample request and response:

```
GET /v4/parking/sessions?place=CARPARK1&credential_id=VRM123&end_after=1750166111
```

```
{
	"meta": {
		"referenceInstant": 1750166111,
		"offset": 0,
		"pageSize": 200,
		"total": 1
	},
	"data": [
		{
			"id": "SESSION1",
			"version": 1,
			// …details of session 1
		}

	]
}
```

Depending on the filters provided, the result may contain multiple sessions. See the next section for an example of session details. The *end_after* query parameter can also be used to for instance query recently expired sessions. Timestamps in query parameters are always specified as epoch seconds.

1.4 Sessions (Push Notification)

The enforcement system (subscribed to the NPP push service) receives a notification of a new or updated session. Data will be pro-actively pushed to the system's webhook endpoint. The custom http request header named "X-Event-Type" will indicate the nature of the data event.

```
POST {provider-specified webhook endpoint}
X-Event-Type: SessionCreated
```

```
{
	"id": "PROVIDER-GENERATED-SESSION-ID-1",
	"version": 1,
	"actualStart": "2025-05-20T10:02:00Z",  // the start of the parking session
	"actualEnd": "2025-05-20T11:02:00Z",    // the end of the parking session
	"initiator": {
		"id": "PROVIDER1",
		"version": 1
	},
	"identifiedCredentials": [
		{
			"type": "licensePlate",
			"identifier": {
				"id": "TST001",     // the parking vehicle's VRM
				"className": "UKNumberPlate"
			},
			"issuer": [
				{
					"language": "en",
					"string": DVLA"
				}
			]
		}
	],
	"hierarchyElement": {
		"id": "CARPARK1",          // location id of where the vehicle is parked
		"version": 1,
		"name": [
			{
				"language": "en",
				"string": "Car Park No 1"
			}
		]
	},
	"segments": [
		{
			"id": "PROVIDER-GENERATED-SESSION-ID-1-SEGMENT-1",
			"version": 1,
			"actualStart": "2025-05-20T10:02:00Z",
			"actualEnd": "2025-05-20T11:02:00Z",
			"assignedRight": {
				"id": "NEW-PARKING-RIGHT-1",
				"version": 1
			},
			"validationType": [
				"licensePlate"
			]
		}
	],
	"identifiedVehicle": {           // details of the parked vehicle (if available)
		"make": "AUDI",
		"color": "BLUE",
		"country": "GB"
	}
}
```

Important note: per the specification, the session record only provides a reference to the hierarchy element (parking location where the vehicle is parked), i.e. id and version. As a convenience feature, it is possible to activate the additional provision of the location name, helping the caller to avoid another call to the inventory API. Please indicate this to the NPP team in case you want this option activated.

2. Shared Use Cases

2.1 Read List of Organisations on NPP

An NPP user reads the list of organisations (contacts) known to the NPP. The endpoint offers filtering by contact type, e.g. "serviceProvider", "operator".

Use cases could be

- An enforcement system retrieving the list of registered service providers via

  `GET /v4/parking/contacts?type=serviceProvider`

- A service provider wants to get a list of all active operators on the platform. For this, they would call

  `GET /v4/parking/contacts?type=operator`

Sample response:

```
{
    "meta": {
        "referenceInstant": 1753883082,
        "offset": 0,
        "pageSize": 200,
        "total": 6
    },
    "data": [
        {
            "id": "NPPDEMO",
            "version": 1,
            "organisationName": [
                {
                    "language": "en",
                    "string": "Demo Provider"
                }
            ],
            "type": "serviceProvider"
        },
// all other records
    ]
}
```

3. APDS Concepts in the NPP Context

3.1 Push Notification Service

Depending on their push notification service subscription (managed in coordination with the NPP team), push API users will receive notifications for different types of events. To do this, they must provide the NPP with corresponding call-back endpoints to which new/updated information will be sent.

This webhook mechanism is

- mandatory for *Service Providers* to receive inventory information in a timely manner (in addition, they shall sporadically query the NPP to make sure nothing was missed)
- optional for *Enforcement Systems* to receive assigned right and session information (alternatively, they can actively query the NPP)

3.1.1 Event Types

Distinguishing the type of event received can be done in two ways:

- Setting up separate endpoints by event type
- Checking the custom http request header "X-Event-Type"

Both is possible, but the latter is the recommended approach. Below is the list of possible event types to be received:

- AssignedRightCreated *(body will contain records of type AssignedRight)*
- AssignedRightUpdated *(body will contain records of type AssignedRight)*
- AssignedRightDeleted *(body will contain records of type AssignedRight)*
- SessionCreated *(body will contain records of type Session)*
- SessionUpdated *(body will contain records of type Session)*
- SessionDeleted *(body will contain records of type Session)*
- ContactCreated *(body will contain records of type ContactPoint)*
- ContactUpdated *(body will contain records of type ContactPoint)*
- ContactDeleted *(body will contain records of type ContactPoint)*

The range of subscribed event types will depend on your NPP role (service provider, operator, connected supplier).

3.1.2 Webhook Endpoint(s)

The NPP Push Notification Service requires you to expose one ore more endpoints for callback purposes. Whenever a data event that you are subscribed to occurs, you will receive a corresponding notification to your endpoint(s). The Push Service is available for both, Inventory API events as well as Assigned Right / Session events (mainly used by enforcement systems).

Currently, the subscription process is a manual one. You provide the NPP team with:

- endpoint URL
- authentication method
- authentication credentials
- list of event types to be notified of

3.1.3 Authentication

While NPP test environments support, but do not require authentication by the NPP (in its role as an HTTP client), this is mandatory for the production environment. You have a choice between

- API Key based authentication and
- OAUTH2 authentication

3.1.3.1 API Key

For the API key based authentication, you will have to provide the NPP team with either the name of a custom header to hold the actual API key / access token or an auth type to be used in the standard *Authorization* header. In addition, you'll have to pre-share the actual API key with us.

Examples for the two variants:

- `Authorization: ProviderApiKey {yourApiKey}`
- `Provider-Specific-Header: {yourApiKey}`

3.1.3.2 OAUTH2

As an alternative, you can offer OAUTH2 based authentication. In that case, you'll have to provide

- the URL of the endpoint where fresh access tokens can be obtained
- the client id and
- the client secret created by you for the NPP push client

The NPP will then create new access tokens as needed (we'll always use a *grant_type* of *client_credentials*).

3.1.4 Example Push Notification

```
POST
X-CLIENT-VERSION: NPP Notification Service 1.2
X-CORRELATION-ID: 097b36ba-ce52-4edd-9123-ac8d20b9acfb
X-EVENT-TYPE: RightSpecificationUpdated
{
	"id": "f1e7af08-eada-4b5e-8d81-1e29a11006f0",
	"version": 1,
	"type": "oneTimeUseParking",
	"description": [{ "language": "en", "string": "Standard Day"}],
	"issuer": { "id": "TESTCOUNCIL", "version": 1, "className": "Operator"},
	"hierarchyElements": [
		{ "id": "5621001", "version": 1},
		{ "id": "5621002", "version": 2}
	],
	// remaining right specification details
}
```

Please note that all notifications are **POST**ed.

3.1.5 Retry on Notification Failure

If sending a push notification to your webhook endpoint fails, the NPP will retry (currently up to three times), with increasing waiting times in between. After the maximum number of configured attempts, the NPP will give up, and the information will have to be retrieved in one of your sporadic active enquiries.

Whilst the NPP currently logs returned error payloads, they're never processed/evaluated. During the introductory phase of the live system inventory API, the NPP may pull error logs upon request.

4. Important Concepts outside the APDS Context

4.1 Authorisation (Roles and Permissions)

The NPP uses a combination of RBAC (role-based access control) and ABAC (attribute-based access control). This chapter provides an overview of the implemented controls.

4.1.1 Roles

4.1.1.1 Atomic Roles

The NPP currently defines the following atomic roles:

Role	Comments
INVENTORY_CONSUMER	read inventory information (locations, right specifications, tariffs, code lists)
INVENTORY_PROVIDER	publish inventory information (locations, right specifications, tariffs, code lists)
RIGHT_CONSUMER	read assigned right information
RIGHT_PROVIDER	create/update and send assigned right information
SESSION_CONSUMER	read session information
SESSION_PROVIDER	create/update and send session information

4.1.1.2 Role Groups

Atomic rules are then grouped as follows:

- OPERATOR
- SERVICE_PROVIDER
- ENFORCEMENT_PROVIDER

The the following table shows the atomic roles included in each one of the above role groups:

Role Group	Included Atomic Roles
OPERATOR	INVENTORY_CONSUMER, INVENTORY_PROVIDER, RIGHT_CONSUMER, RIGHT_PROVIDER, SESSION_CONSUMER, SESSION_PROVIDER
SERVICE_PROVIDER	INVENTORY_CONSUMER, RIGHT_CONSUMER, RIGHT_PROVIDER, SESSION_CONSUMER, SESSION_PROVIDER
ENFORCEMENT_PROVIDER	INVENTORY_CONSUMER, RIGHT_CONSUMER, SESSION_CONSUMER

4.1.2 Attribute-based Access Control

In addition to the role-based access control, further – attribute-based – constraints apply.

- An Operator can only create and update inventory information that relates to locations within their own NPP-assigned range of location codes.
- Users with the RIGHT_PROVIDER/SESSION_PROVIDER roles can only modify assigned right/session records that they created.
- Users with the SERVICE_PROVIDER group role can only read assigned right/session data created by themselves.
- Users with the ENFORCEMENT_PROVIDER group role can read session data across all service providers.
- Users with the ENFORCEMENT_PROVIDER group role can only read data relating to locations that they have been contracted for (and subscribed to).

Annexes

Annex 1: Document Management

Date	Version	Author(s)	Description
14/04/2026	1.0	Markus Schneider	Initial Version