Fair Fuel API Documentation

for External Integration

Document Version 1.0
4 Aug 2025

Table of Contents

1. Document information

Criteria	Details
Document title:	Fair Fuel API Documentation for External Integration
Document owner:	Service Victoria
Document author:	Service Victoria
Document Version:	1.0
Issue date:	5 Aug 2025

2. Document revision history

Document Version	Date	Updated by	Description
0.1	17 Apr 2025	Ken Singh	Initial draft prepared based on internal OpenAPI specifications. A separate copy of the OpenAPI spec is available upon request.
0.1	22 Apr 2025	David Chanter	Reviewed for early release to partners and reclassified to OFFICIAL
0.2	29 May 2025	Ken Singh	This version introduces new GET APIs for retrieving fuel prices and station metadata for a fuel retailer's own or authorised stations.
0.2	30 May 2025	David Chanter	Approved for release
0.3	28 July 2025	Ken Singh	This version includes: • New header in the API request. • Clarified price field rule when fuel type is unavailable • Troubleshooting guide.
1.0	4 Aug 2025	Ken Singh	Final approved release for production use.
1.0	5 Aug 2025	David Chanter	Approved for release

3. Document Approval & Distribution

Document Version	Name/Position	Action	Date
0.1	Ken Singh (Enterprise Solution Architect)	Approved	17 Apr 2025
0.1	David Chanter (Information Security Manager)	Reviewer	22 Apr 2025
0.1	Maryam Sahrapour (Delivery Lead)	Reviewer	22 Apr 2025
0.1	Daniel Kerr (Tech Lead)	Reviewer	22 Apr 2025
0.1	Ben Stephenson (Release Manager)	Reviewer	22 Apr 2025
0.2	Ken Singh (Enterprise Solution Architect)	Approved	29 May 2025
0.2	Maryam Sahrapour (Delivery Lead)	Reviewer	30 May 2025
0.2	Daniel Kerr (Tech Lead)	Reviewer	30 May 2025
0.2	Nick Iannelli (Tech Lead)	Reviewer	30 May 2025
0.2	David Chanter (Information Security Manager)	Reviewer	30 May 2025
0.3	Ken Singh (Enterprise Solution Architect)	Approved	28 July 2025
0.3	Maryam Sahrapour (Delivery Lead)	Reviewer	30 July 2025
0.3	Daniel Kerr (Tech Lead)	Reviewer	30 July 2025
1.0	Ken Singh (Enterprise Solution Architect)	Approved	4 Aug 2025
1.0	David Chanter (Information Security Manager)	Approved	5 Aug 2025
1.0	Scott Mineo (Senior Manager, Service Management)	Approved	5 Aug 2025
1.0	Maryam Sahrapour (Delivery Lead)	Approved	5 Aug 2025
1.0	Daniel Kerr (Tech Lead)	Approved	5 Aug 2025

4. Background

This document outlines how fuel retailers can submit fuel price data to Service Victoria using the Fair Fuel API. The Fair Fuel API is a RESTful interface that allows registered fuel retailers to programmatically publish fuel prices for their service station locations in Victoria.

This document is intended to assist fuel retailers in integrating their internal systems with the Fair Fuel application on the Service Victoria Platform to ensure accurate, timely, and compliant fuel price updates. The API has been designed to support secure, automated fuel price submissions and ensure information is received in a structured and consistent format.

4.1 Purpose

This document provides information about using the Fair Fuel API developed by Service Victoria. The API allows fuel retailers to submit up-to-date fuel price information for their service stations.

The document outlines how authorised retailers can integrate their systems with the Service Victoria platform to securely publish price updates. It is structured to provide clear, non-technical explanations in the early sections, followed by more detailed technical guidance to assist IT teams in completing the integration successfully.

An API Agreement is a reference document, like a technical manual, that defines the API and explains how to use it. It includes information about the API's services, the operations it supports, and the structure of the request and response messages.

It is expected that this API documentation will include and elaborate on the following information to support integration with the Fair Fuel application on the Service Victoria Platform:

- Authentication is required for each request.
- The structure and base path of the API endpoint.
- The HTTP method is to be used with the endpoint.
- Definitions of mandatory and optional request fields (including a field-level specification).
- The meaning and usage of each API status code.
- The expected data formats for requests and responses.
- Examples of request and response payloads (or references to these examples).

This document is structured to provide this information clearly and consistently to help fuel retailers, their vendors, and Service Victoria teams understand the API and support successful implementation.

4.2 Audience

This documentation is written for:

- Fuel retailer IT teams are responsible for system integration
- Technical developers and solution architects

Users of this API are expected to have experience working with REST APIs, JSON data formats, and secure integration using API keys.

5. Scope Summary

5.1 In-Scope (Inclusions)

This system is designed for trusted third-party systems, i.e. authorised fuel retailers, to programmatically submit fuel price updates to the Fair Fuel application on the Service Victoria Platform. The solution enables retailers to publish current price and availability information for one or more fuel types at their registered stations using the Fair Fuel API.

5.2 Out-of-Scope (Exclusions)

The following items are deemed out of scope and are not included in the current delivery scope:

- Any other API services provided by Service Victoria, unrelated to fuel pricing.
- Any activity or request that is not specifically defined as in-scope within this documentation.
- Retrieval or display of public-facing fuel price information.

6. System assumptions

6.1 Assumptions

Ref#	Assumptions
A1	This functionality assumes that fuel retailers will submit data using their registered station identifiers, which are pre-configured in Service Victoria’s backend systems.
A2	Each request submitted via the API will contain at least one valid fuel type and price entry for processing.
A3	The authentication key will be managed and verified by Service Victoria before allowing access to the API.
A4	Fuel price updates are assumed to be provided in near real-time by retailers, and the latest submission will overwrite any previous values for that fuel type and station.

7. API Services

7.1 Overview

This API specification supports the requirement for fuel retailers to publish fuel price data for their registered fuel stations. This is achieved via a single POST endpoint, which accepts a payload containing one or more fuel price entries.

This API has been designed to allow automated, secure submissions directly from retailer systems. All requests must conform to the defined schema and will be subject to authentication and IP controls managed by Service Victoria.

7.2 Full List of Operations

Environment	Base URL
Production	https://api.fuel.service.vic.gov.au/b2b/v1

Action	End Point	Description
POST	*{{baseUrl}}*/fuel/prices/update	Accepts fuel price update submissions from authorised fuel retailers. Each request can include price updates for one or more registered stations and fuel types.
GET	*{{baseUrl}}*/fuel/prices	Retrieves the latest fuel prices all available authorised fuel stations submitted by the retailer. If a price update was just submitted via API or Web Portal, allow a minimum 10-second delay before calling this endpoint to ensure the data has been processed.
GET	*{{baseUrl}}*/fuel/stations	Returns metadata for fuel stations associated with the authorised retailer, including address and location, and station identifiers for display or integration purposes.

7.3 API Security

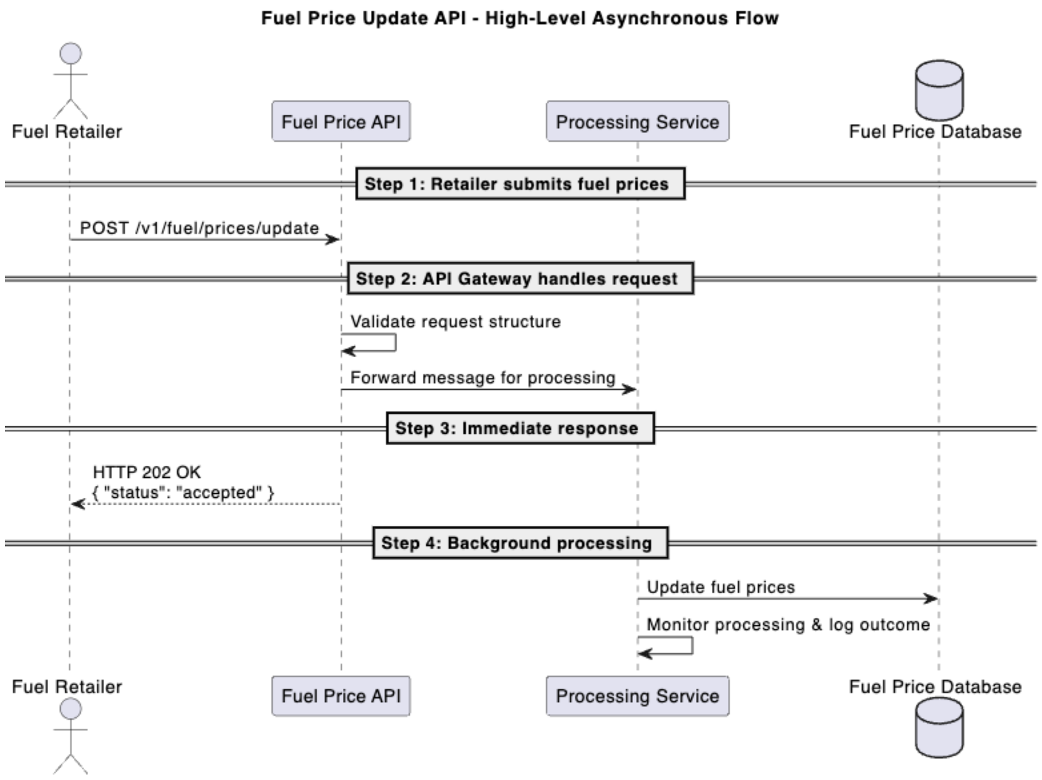
Access to this API will be protected using the following security mechanisms:

1. **API Keys:** Each registered fuel retailer will be issued a unique API key. The key must be included in all requests using the ‘x-api-key’ header.

2. **IP Whitelisting:** The API enforces IP-based access control to restrict incoming traffic and validate requests.

7.4 Fuel Price Update API

7.4.1 API Sequence Diagram


Fuel Price Update API - High-Level Asynchronous Flow
Fuel Retailer
Fuel Price API
Processing Service
Fuel Price Database
Step 1: Retailer submits fuel prices
POST /v1/fuel/prices/update
Step 2: API Gateway handles request
Validate request structure
Forward message for processing
Step 3: Immediate response
HTTP 202 OK
{ "status": "accepted" }
Step 4: Background processing
Update fuel prices
Monitor processing & log outcome
Fuel Retailer
Fuel Price API
Processing Service
Fuel Price Database

7.4.2 API Request

Action	End Point	Description
POST	*{{baseUrl}}*/fuel/prices/update	Accepts fuel price update submissions from authorised fuel retailers. Each request can

		include price updates for one or more registered stations and fuel types.

Headers:

Field	Type	Example	Comments
Content-Type	string	application/json	Required. Specifies format of request body.
User-Agent	string	MyApp/1.0	Identifies the client application making the request
x-api-key	String	8f9qWOLO3Ja14sXiIPBhn39YnRCZQtXjNFUDm9k7	API key issued to each retailer.
x-transactionid	string (UUID)	550e8400-e29b-41d4-a716-446655440000	Used for tracing/logging. Retailers are expected to generate a unique, random UUID v4 for each request. While the system does not enforce uniqueness or validate against reuse, values should not be reused across requests.

Payload:

Field	Type	Path	Example	Required	Comments
stations	array of objects	stations	See below	Yes	Each request must include at least 1 and no more than 100 station entries.
identifier	string	stations[].identifier	a019r00000iRqPOAA0	Yes	Unique ID assigned by Service Victoria, available via the Business Profile Portal.
fuelPrices	array of objects	stations[].fuelPrices	See below	Yes	One or more fuel type entries.
fuelType	string	stations[].fuelPrices[].fuelType	E10	Yes	Refer to section: 7.4.5 for a full List of permitted values.

isAvailable	boolean	stations[].fuelPrices[].isAvailable	true / false	Yes	Indicates if the fuel is available.
price	number	stations[].fuelPrices[].price	165.9	Yes, if *isAvailable* is true Must not include if *isAvailable* is false	Price is in CENTS per litre, with a single digit after the decimal point (e.g., 165.3 = $1.65 3/10). Must be greater than 0 and less than or equal to 9999.9. Required if isAvailable is true.

Note: Marking a fuel type unavailable

If a **fuel type** is marked as **unavailable** (isAvailable = false), the **price** field **must not** be included in the payload. Including a price in such cases will result in a *400 Bad Request* response.

See Section 7.4.5 – Sample Examples (Example 3) for a correct implementation.

7.4.3 API Response

HTTP Response – 202

Field	Type	Example	Comments
status	string	"accepted"	API does not process the data immediately, success indicates queuing only.
warnings	array (optional)	[]	This field is reserved for future enhancements. No warnings are currently returned.

7.4.4 Success Codes / Error Codes / Messages

If the call is unsuccessful, an error code and message are returned. Below are common responses for the Fair Fuel Price Update API:

HTTP Code	Body	Cause / Explanation
Success Codes		

202	Accepted	The request was accepted successfully. API does not process the data immediately, success indicates queuing only.
Client Error Codes		
400	Invalid request	One or more required fields are missing, malformed, or invalid. Recommendation: Do not retry. Check the response body for additional details on the specific validation error.
403	Forbidden	API key is missing or invalid, or IP address is not whitelisted. Recommendation: Do not retry. Confirm credentials and IP allowlist configuration.
413	Payload too large	Recommendation: Split into smaller requests.
429	Rate limiting / Too many requests	The request exceeded the API rate limit. **Recommendation:** Avoid burst requests. The API is configured to allow a maximum of **10 requests per second**. If this limit is exceeded, retry after the full 60-second window.
Server Error Codes		
500	Internal server error	An unexpected server error occurred. Recommendation: Retry 2 times with 5–10 second intervals. If the issue persists or occurs repeatedly, especially during time-sensitive updates, please contact Service Victoria for support.
503, 504	Timeout	No response from the Fair Fuel API within timeout window. Recommendation: Retry 2 times with delay.

7.4.5 Sample Examples

Sample request: [Maximum payload size can be 250KB]

Example 1 – Single Station with Different Fuel Types (Sample of Available Fuel Types)

```
{
  "stations": [
    {
      "identifier": "a019r00000iRqPOAA0",
      "fuelPrices": [
        {
          "fuelType": "E10",
          "isAvailable": true,
          "price": 180.0
        },
        {
            "fuelType": "U91",
            "isAvailable": true,
            "price": 188.8
          },
          {
            "fuelType": "P95",
            "isAvailable": true,
            "price": 195.0
          },
          {
            "fuelType": "P98",
            "isAvailable": true,
            "price": 209.0
          },
          {
            "fuelType": "DSL",
            "isAvailable": true,
            "price": 198
          },
          {
            "fuelType": "PDSL",
            "isAvailable": true,
            "price": 226
          },
          {
            "fuelType": "LPG",
            "isAvailable": true,
            "price": 103.5
          }
      ]
    }
  ]
}
```

Example 2 – Multiple Stations with Different Fuel Types (Sample of Available Fuel Types)

```
{
  "stations": [
    {
      "identifier": "a019r00000iRqPOAA0",
      "fuelPrices": [
        {
          "fuelType": "DSL",
          "isAvailable": true,
          "price": 194.0
        },
        {
          "fuelType": "PDSL",
          "isAvailable": true,
          "price": 186.3
        },
        {
          "fuelType": "E10",
          "isAvailable": true,
          "price": 174.0
        }
      ]
    },
    {
      "identifier": "a091r01110iArQOXS0",
      "fuelPrices": [
        {
          "fuelType": "LNG",
          "isAvailable": true,
          "price": 386.0
        },
        {
          "fuelType": "U91",
          "isAvailable": true,
          "price": 180.0
        }
      ]
    },
    {
      "identifier": "a029r00000dRaPOQW1",
      "fuelPrices": [
        {
          "fuelType": "P95",
          "isAvailable": true,
          "price": 192
        },
        {
          "fuelType": "LPG",
          "isAvailable": true,
          "price": 102.3
        },
        {
          "fuelType": "CNG",
          "isAvailable": true,
          "price": 331.0
        }
      ]
    }
  ]
}
```

Example 3 – Multiple Fuel Stations (One Fuel Type is Unavailable)

```
{
  "stations": [
    {
      "identifier": "a019r00000iRqPOAA0",
      "fuelPrices": [
        {
          "fuelType": "CNG",
          "isAvailable": true,
          "price": 310.0
        },
        {
          "fuelType": "B20",
          "isAvailable": false
        },
        {
          "fuelType": "LPG",
          "isAvailable": true,
          "price": 101.3
        }
      ]
    },
    {
      "identifier": "a091r01110iArQOXS0",
      "fuelPrices": [
        {
          "fuelType": "E85",
          "isAvailable": true,
          "price": 173.2
        },
        {
          "fuelType": "P98",
          "isAvailable": false
        }
      ]
    },
    {
      "identifier": "a029r00000dRaPOQW1",
      "fuelPrices": [
        {
          "fuelType": "E10",
          "isAvailable": false
        },
        {
          "fuelType": "LNG",
          "isAvailable": true,
          "price": 416.0
        },
        {
          "fuelType": "U91",
          "isAvailable": true,
          "price": 178.0
        }
      ]
    }
  ]
}
```

Example 4 – Successful Response Payload (HTTP 202 Accepted)

API does not process the data immediately, success indicates queuing only.

```
{
  "status": "accepted",
  "warnings": []
}
```

7.4.6 Field-Level Schema (OpenAPI Format)

fuelPrices (array)

```
components:
  schemas:
    FuelPrice:
      properties:
        fuelType:
          description: Type of fuel sold at this station. Must use one of
the predefined codes.
          enum:
          - U91
          - P95
          - P98
          - DSL
          - PDSL
          - E10
          - E85
          - B20
          - LPG
          - LNG
          - CNG
          type: string
        isAvailable:
          description: Indicates if this fuel type is currently available.
          type: boolean
        price:
          description: Price per litre in AUD cents. Required if fuel type
is available. Value must include exactly one digit after the decimal point
(e.g., 165.0 = $1.65)
          maximum: 9999.99
          minimum: 0.1
          type: number
      required:
      - fuelType
      - isAvailable
      type: object
```

stations

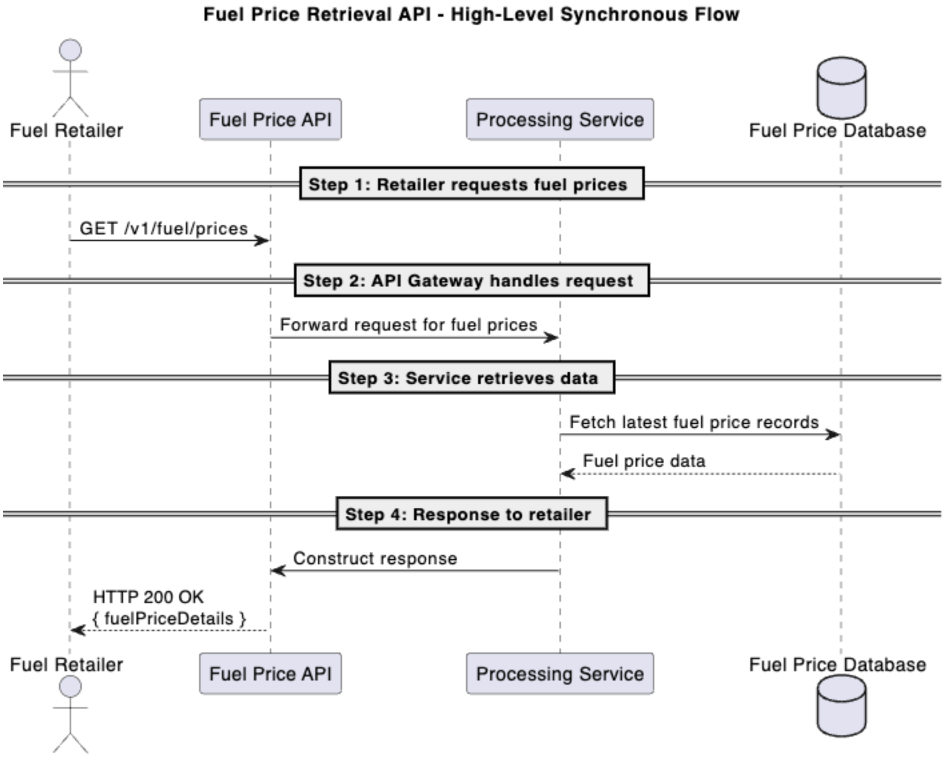
```
stations:
  type: array
  description: A list of fuel stations to update.
  minItems: 1
  maxItems: 100
  items:
    type: object
    required:
      - identifier
      - fuelPrices
    properties:
      identifier:
        type: string
        description: Unique ID assigned to the fuel station.
        example: a019r00000iRqPOAA0
      fuelPrices:
          $ref: '#/components/schemas/fuelPrices'
```

7.4.5 List of Fuel Types

SNo	Fuel Type Name	Fuel Type Code
1	Unleaded 91	U91
2	Premium Unleaded 95	P95
3	Premium Unleaded 98	P98
4	Diesel	DSL
5	Premium Diesel	PDSL
6	Ethanol 10	E10
7	Ethanol 85	E85
8	Biodiesel 20	B20
9	Liquefied Petroleum Gas	LPG
10	Liquefied Natural Gas	LNG
11	Compressed Natural Gas	CNG

7.5 Get Fuel Price API

7.5.1 API Sequence Diagram


Fuel Price Retrieval API - High-Level Synchronous Flow
Fuel Retailer
Fuel Price API
Processing Service
Fuel Price Database
Step 1: Retailer requests fuel prices
GET /v1/fuel/prices
Step 2: API Gateway handles request
Forward request for fuel prices
Step 3: Service retrieves data
Fetch latest fuel price records
Fuel price data
Step 4: Response to retailer
Construct response
HTTP 200 OK
{ fuelPriceDetails }
Fuel Retailer
Fuel Price API
Processing Service
Fuel Price Database

7.5.2 API request

Action	End Point	Description
GET	https://api.fuel.service.vic.gov.au/b2b/v1/fuel/prices	Retrieves the latest fuel prices for one or more authorised fuel stations submitted by the retailer. If a price update was just submitted via API or Web Portal, allow a minimum 10-second delay before calling this endpoint to ensure the data has been processed.

Headers:

Field	Type	Example	Comments
Content-Type	string	application/json	Required. Specifies format of request body.
User-Agent	string	MyApp/1.0	Identifies the client application making the request
x-api-key	String	8f9qWOLO3Ja14sXiIPBhn39YnRCZQtXjNFUDm9k7	API key issued to each retailer.
x-transactionid	string (UUID)	960e8410-e29b-42d4-d716-576655440000	Used for tracing/logging. Retailers are expected to generate a unique, random UUID v4 for each request. While the system does not enforce uniqueness or validate against reuse, values should not be reused across requests.

7.5.3 API Response

HTTP Response – 200 OK

Payload:

Field	Type	Path	Example	Comments
fuelPriceDetails	array of objects	fuelPriceDetails	See below	List of fuel stations and corresponding fuel prices.
fuelStation.id	string	fuelPriceDetails[].fuelStation.id	a019r00000iRqP0AA0	Unique identifier for the fuel station.

fuelPrices	array of objects	fuelPriceDetails[].fuelPrices	See below	List of fuel types and their price details.
fuelType	string	fuelPriceDetails[].fuelPrices[].fuelType	P95	Code for the fuel type.
price	number	fuelPriceDetails[].fuelPrices[].price	213.9	Price in cents per litre.
isAvailable	boolean	fuelPriceDetails[].fuelPrices[].isAvailable	true	Whether the fuel type is currently available.
updatedAt	string (ISO 8601 datetime)	fuelPriceDetails[].fuelPrices[].updatedAt	2025-05-22T01:30:00Z	Date Timestamp of the most recent price update.
isVisibleOnPublicApi	boolean	fuelPriceDetails[].fuelPrices[].isVisibleOnPublicApi	true	Whether the price is visible in the public API.
timestamp	string (ISO 8601 datetime)	timestamp	2025-05-22T01:30:00Z	Date Timestamp indicating when the response was generated.

7.5.4 Success Codes / Error Codes / Messages

If the call is unsuccessful, an error code and message are returned. Below are common responses for the Fair Fuel Price Update API:

HTTP Code	Body	Cause / Explanation
Success Codes		
200	OK	The request was processed successfully.
Client Error Codes		
400	Invalid request	Invalid request. Recommendation: Do not retry. Missing required headers or invalid path.
403	Forbidden	API key is missing or invalid, or IP address is not whitelisted. Recommendation: Do not retry. Confirm credentials and IP allowlist configuration.

404	Not Found	No Fuel Station Prices Found
429	Rate limiting / Too many requests	The request exceeded the API rate limit. **Recommendation:** Avoid burst requests. The API is configured to allow a maximum of 10 requests per 60 seconds. If this limit is exceeded, retry after the full 60-second window. Do not assume a 2-second retry interval is sufficient, actual wait time depends on WAF rate limit evaluation windows (1, 2, 5, or 10 minutes).
Server Error Codes		
500	Internal server error	An unexpected server error occurred. Recommendation: Retry 2 times with 5–10 second intervals. If the issue persists or occurs repeatedly, especially during time-sensitive updates, please contact Service Victoria for support.
504	Timeout	No response from the Fair Fuel API within timeout window. Recommendation: Retry 2 times with delay.

7.5.5 Sample Examples

Example 1 – Single Station with Different Fuel Types

Successful Response Payload (HTTP 200 OK)

```
{
    "fuelPriceDetails": [
        {
            "fuelStation": {
                "id": "a019r00000iRqPOAA0"
            },
            "fuelPrices": [
                {
                    "fuelType": "P95",
                    "price": 167.7,
                    "isAvailable": true,
                    "updatedAt": "2025-05-18T01:15:00Z",
                    "isVisibleOnPublicApi": true
                },
                {
                    "fuelType": "P98",
                    "price": 174.7,
                    "isAvailable": true,
                    "updatedAt": "2025-05-18T01:15:00Z",
                    "isVisibleOnPublicApi": true
                },
                {
                    "fuelType": "DSL",
                    "price": 159.5,
                    "isAvailable": true,
                    "updatedAt": "2025-05-18T01:15:00Z",
                    "isVisibleOnPublicApi": true
                },
                {
                    "fuelType": "LPG",
                    "price": 95.9,
                    "isAvailable": true,
                    "updatedAt": "2025-05-18T01:15:00Z",
                    "isVisibleOnPublicApi": true
                }
            ]
        }
    ],
    "timestamp": "2025-05-18T01:20:00Z",
}
```

Example 2 – Three Fuel Stations with Different Fuel Types

Successful Response Payload (HTTP 200 OK)

```
{
  "fuelPriceDetails": [
    {
      "fuelStation": {
        "id": "z019x00000qRdP3AA1"
      },
      "fuelPrices": [
        {
          "fuelType": "P95",
          "price": 212.5,
          "isAvailable": true,
          "updatedAt": "2025-05-28T07:30:00Z",
          "isVisibleOnPublicApi": true
        },
        {
          "fuelType": "DSL",
          "price": 189.0,
          "isAvailable": true,
          "updatedAt": "2025-05-28T07:30:00Z",
          "isVisibleOnPublicApi": true
        }
      ]
    },
    {
      "fuelStation": {
        "id": "n342f00000tRxP2AA0"
      },
      "fuelPrices": [
        {
          "fuelType": "U91",
          "price": 199.7,
          "isAvailable": true,
          "updatedAt": "2025-05-28T07:32:00Z",
          "isVisibleOnPublicApi": false
        },
        {
          "fuelType": "E10",
          "price": 195.3,
          "isAvailable": true,
          "updatedAt": "2025-05-28T07:32:00Z",
          "isVisibleOnPublicApi": true
        },
        {
          "fuelType": "P98",
          "price": 223.4,
          "isAvailable": true,
          "updatedAt": "2025-05-28T07:32:00Z",
          "isVisibleOnPublicApi": true
        }
      ]
    },
    {
      "fuelStation": {
        "id": "c025r00000iRqP3AA0"
      },
      "fuelPrices": [
        {
          "fuelType": "LPG",
          "price": 177.8,
          "isAvailable": true,
          "updatedAt": "2025-05-28T07:34:00Z",
          "isVisibleOnPublicApi": true
        }
      ]
    }
  ],
  "timestamp": "2025-05-28T07:35:00Z"
}
```

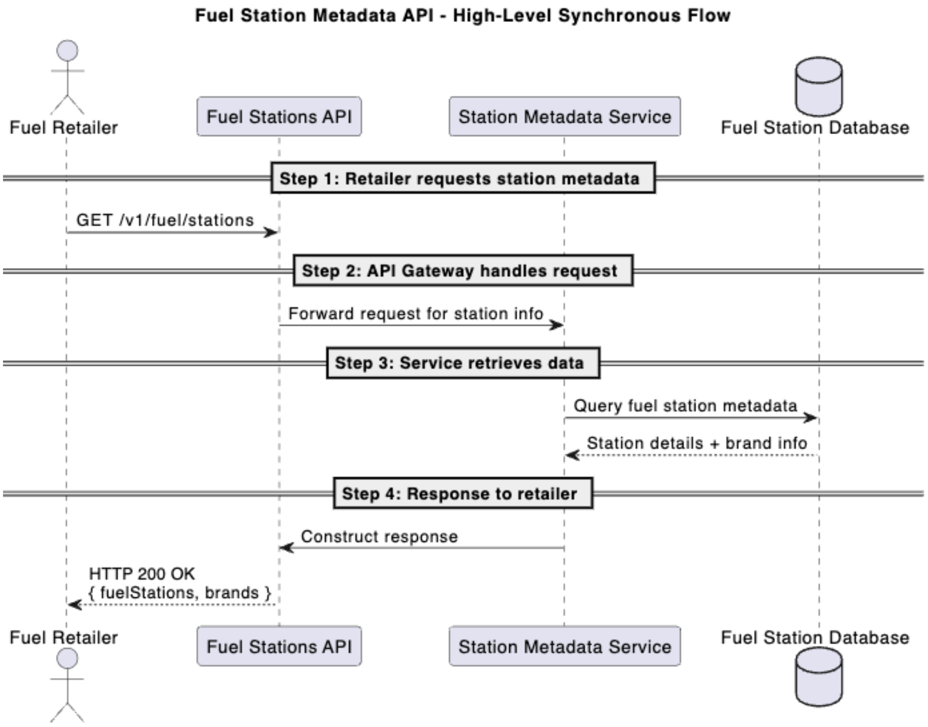
7.5.6 Field-Level Schema (OpenAPI Format)

fuelPriceDetails (array)

```
components:
  schemas:
    FuelPriceDetails:
      properties:
        fuelStation:
          type: object
          properties:
            id:
              type: string
              description: The unique ID of the fuel station that this
record pertains to.
        fuelPrices:
          type: array
          items:
            type: object
            properties:
              fuelType:
                type: string
                enum:
                  - U91
                  - P95
                  - P98
                  - DSL
                  - PDSL
                  - E10
                  - E85
                  - B20
                  - LPG
                  - LNG
                  - CNG
                description: The fuel type associated with this item.
              price:
                type: number
                minimum: 0.1
                maximum: 9999.9
                multipleOf: 0.1
                nullable: true
                description: Price per litre in AUD cents. Must include
exactly one digit after the decimal point (e.g., 165.0 = $1.65).
              isAvailable:
                type: boolean
                description: Indicates if the fuel is currently available.
              updatedAt:
                type: string
                format: date-time
                description: The date and time this fuel price record was
last updated (ISO 8601).
              isVisibleOnPublicApi:
                type: boolean
                description: True if the value is available on the public
API. False means price is suppressed from public view.
      required:
        - fuelStation
        - fuelPrices
```

7.6 Get Fuel Stations API

7.6.1 API Sequence Diagram


Fuel Station Metadata API - High-Level Synchronous Flow
Fuel Retailer
Fuel Stations API
Station Metadata Service
Fuel Station Database
Step 1: Retailer requests station metadata
GET /v1/fuel/stations
Step 2: API Gateway handles request
Forward request for station info
Step 3: Service retrieves data
Query fuel station metadata
Station details + brand info
Step 4: Response to retailer
Construct response
HTTP 200 OK
{ fuelStations, brands }
Fuel Retailer
Fuel Stations API
Station Metadata Service
Fuel Station Database

7.6.2 API request

Action	End Point	Description
GET	*{{baseUrl}}*/fuel/stations	Returns metadata for fuel stations associated with the authorised retailer, including address, and station identifiers for display or integration purposes.

Headers:

Field	Type	Example	Comments
Content-Type	string	application/json	Required. Specifies format of request body.
User-Agent	string	MyApp/1.0	Identifies the client application making the request
x-api-key	String	8f9qWOLO3Ja14sXiIPBhn39YnRCZQtXjNFUDm9k7	API key issued to each retailer.

x-transactionid	string (UUID)	960e8410-e29b-42d4-d716-576655440000	Used for tracing/logging. Retailers are expected to generate a unique, random UUID v4 for each request. While the system does not enforce uniqueness or validate against reuse, values should not be reused across requests.

7.6.3 API Response

HTTP Response – 200 OK

Payload:

Field	Type	Path	Example	Comments
brands	array of objects	brands	See below	List of fuel brand entries.
id	string	brands[].id	testb001	Unique identifier for the fuel brand.
name	string	brands[].name	TestBrand	Name of the fuel brand.
logoUrl	string	brands[].logoUrl	http://example.com/test-marker.png	URL for the brand's logo or map marker. The Logo image at the given path is protected by the API key and IP Allowlist. When you make a request to retrieve this image, you must supply the x-api-key header.
fuelStations	array of objects	fuelStations	See below	List of fuel station entries.
id	string	fuelStations[].id	n342f00000tRxP2AA0	Unique identifier for the fuel station.
name	string	fuelStations[].name	Main Street Fuel	Display name of the fuel station.

brandId	string	fuelStations[].brandId	testb001	ID of the associated fuel station brand.
location.address	string	fuelStations[].location.address	123 Main St	Street address of the fuel station.
location.suburb	string	fuelStations[].location.suburb	Melbourne	Suburb where the fuel station is located.
location.postcode	string	fuelStations[].location.postcode	3000	Postcode of the fuel station location.
location.state	string	fuelStations[].location.state	VIC	State of the fuel station location.
location.latitude	number	fuelStations[].location.latitude	-37.8136	Latitude coordinate of the fuel station.
location.longitude	number	fuelStations[].location.longitude	144.9631	Longitude coordinate of the fuel station.
isVisibleOnPublicApi	boolean	fuelStations[].isVisibleOnPublicApi	true	Indicates if fuel station is visible on public API.
timestamp	string (ISO 8601 datetime)	timestamp	2025-05-18T01:20:00Z	Timestamp when the data was generated. As the data at this endpoint can be cached, this timestamp may show as a value in the past.

7.6.4 Success Codes / Error Codes / Messages

If the call is unsuccessful, an error code and message are returned. Below are common responses for the Fair Fuel Price Update API:

HTTP Code	**Body**	**Cause / Explanation**
Success Codes		

200	OK	The request was processed successfully.
Client Error Codes		
400	Invalid request	Invalid request. Recommendation: Do not retry. Missing required headers or invalid path.
403	Forbidden	API key is missing or invalid, or IP address is not whitelisted. Recommendation: Do not retry. Confirm credentials and IP allowlist configuration.
429	Rate limiting / Too many requests	The request exceeded the API rate limit. **Recommendation:** Avoid burst requests. The API is configured to allow a maximum of 10 requests per 60 seconds. If this limit is exceeded, retry after the full 60-second window. Do not assume a 2-second retry interval is sufficient, actual wait time depends on WAF rate limit evaluation windows (1, 2, 5, or 10 minutes).
Server Error Codes		
500	Internal server error	An unexpected server error occurred. Recommendation: Retry 2 times with 5–10 second intervals. If the issue persists or occurs repeatedly, especially during time-sensitive updates, please contact Service Victoria for support.
503, 504	Timeout	No response from the Fair Fuel API within timeout window. Recommendation: Retry 2 times with delay.

7.6.5 Sample Examples

Example 1 – Single Fuel Station Metadata
Successful Response Payload (HTTP 200 OK)

```
{
    "brands": [
        {
            "id": "testb001",
            "name": "TestBrand",
            "mapMarkerImageUrl": "http://example.com/test-marker.png?hash=x"
        }
    ],
    "fuelStations": [
        {
            "fuelStation": {
                "id": "n342f00000tRxP2AA0",
                "name": "Main Street Fuel",
                "brandId": "testb001",
                "location": {
                    "address": "123 Main St",
                    "suburb": "Melbourne",
                    "postcode": "3000",
                    "state": "VIC",
                    "latitude": -37.8136,
                    "longitude": 144.9631
                },
                "isVisibleOnPublicApi": true
            }
        }
    ],
    "timestamp": "2025-05-18T01:20:00Z",
}
```

Example 2 – Two Fuel Stations Metadata

Successful Response Payload (HTTP 200 OK)

```
{
  "brands": [
    {
      "id": "testb001",
      "name": "TestBrand",
      "mapMarkerImageUrl": "http://example.com/test-marker.png?hash=y"
    },
    {
      "id": "fastf002",
      "name": "FastFuel",
      "mapMarkerImageUrl": "http://example.com/fastfuel-marker.png?hash=z"
    }
  ],
  "fuelStations": [
    {
      "fuelStation": {
        "id": "n342f00000tRxP2AA0",
        "name": "Main Street Fuel",
        "brandId": "testb001",
        "location": {
          "address": "123 Main St",
          "suburb": "Melbourne",
          "postcode": "3000",
          "state": "VIC",
          "latitude": -37.8136,
          "longitude": 144.9631
        },
        "isVisibleOnPublicApi": true
      }
    },
    {
      "fuelStation": {
        "id": "x874g00000kTyZ8BB1",
        "name": "Fast Fuel",
        "brandId": "fastf002",
        "location": {
          "address": "41 West Rd",
          "suburb": "Footscray",
          "postcode": "3011",
          "state": "VIC",
          "latitude": -37.8000,
          "longitude": 144.9000
        },
        "isVisibleOnPublicApi": true
      }
    }
  ],
  "timestamp": "2025-05-28T02:00:00Z"
}
```

7.6.6 Field-Level Schema (OpenAPI Format)

fuelStations & brands (array)

```
components:
  schemas:
    StationMetadataResponse:
      type: object
      properties:
        brands:
          type: array
          items:
            type: object
            properties:
              brandId:
                type: string
                description: Unique identifier for the brand.
              name:
                type: string
                description: Display name of the brand.
              mapMarkerImageUrl:
                type: string
                description: URL to the brand's logo or map marker image.
            required:
              - brandId
              - name
              - mapMarkerImageUrl

        fuelStations:
          type: array
          items:
            type: object
            properties:
              fuelStation:
                type: object
                properties:
                  id:
                    type: string
                    description: Unique identifier for the fuel station.
                  name:
                    type: string
                    description: Display name of the fuel station.
                  brandId:
                    type: string
                    description: ID of the associated brand.
                  location:
                    type: object
                    properties:
                      address:
                        type: string
                      suburb:
                        type: string
                      postcode:
                        type: string
                      state:
                        type: string
                      latitude:
                        type: number
                      longitude:
                        type: number
                    required:
                      - address
                      - suburb
                      - postcode
                      - state
                      - latitude
                      - longitude
                  isVisibleOnPublicApi:
                    type: boolean
                    description: Indicates if this station appears on the
public API.
                required:
                  - id
                  - name
                  - brandId
                  - location
                  - isVisibleOnPublicApi

        timestamp:
          type: string
          format: date-time
          description: ISO 8601 timestamp of the response.

      required:
        - brands
        - fuelStations
        - timestamp
```

8. Troubleshooting Guide

8.1 Common Scenarios

This section provides common troubleshooting scenarios and a checklist to assist fuel retailers in diagnosing and resolving integration issues when using the Fair Fuel API or the Price Reporting Web Portal.

Scenario: Cannot Publish Prices Using the B2B API

Before publishing, ensure the following conditions are met:

- Your business and stations are registered and active.
- You have received and configured your API key.
- Your IP address has been allowed/whitelisted via the Business Profile Portal.
- You are using the correct HTTP method and endpoint.
- All required request headers are present:
  - x-api-key
  - x-transactionid
  - User-Agent
  - Content-Type: application/json
- Your JSON payload matches the required schema, including valid data types and field values.

Scenario: Cannot Publish Prices Using the Price Reporting Portal

Ensure:

- You are using a valid Service Victoria customer account to log in.
- Your business and station data are active and linked to your account.
- You are accessing the correct portal environment.
  - **Price Reporting Web Portal URL:**
    - Production: https://service.vic.gov.au/services/fuel/portal
- Your session is valid (clear browser cookies if necessary).
- You are submitting fuel price updates for one station at a time with prices in valid ranges (e.g. greater than $0 and less than $99.999).

8.2 Troubleshooting Checklist for API Users

Checkpoint	Description
☑ API key and IP address	Ensure the API key is active, and the request is from an IP address that is registered in your businesses IP address allowlist on the Business Profile Portal.
☑ HTTP method and endpoint	If you are performing a fuel price update, ensure you are using the HTTP "POST" method. If you are retrieving fuel price listings, ensure you are using the HTTP "GET" method.
☑ Headers included	Verify all required headers are included. Common required headers include: • Transaction ID (X-TransactionID) • API Key (x-api-key) • User Agent (User-Agent)
☑ Valid JSON payload	Ensure payload is well-formed and matches the contract.
☑ Station and fuel identifiers	Confirm identifiers are correct and belong to your business.
☑ Response review	Check HTTP status codes and response messages for error details.

Common error examples:

HTTP Code	Explanation
400	Bad request. Review your payload for format or schema issues.
403	Forbidden. Check your API key and ensure your public IP is in the allowlist.

404	Not Found. Confirm endpoint URL is correct.
429	Rate limit exceeded. Reduce frequency of API calls.
500-504	Temporary error. Retry with delay and contact support if persistent.

8.3 Additional Tips

- After updating your IP allowlist or API key, wait a few minutes before retrying.
- Use unique transaction IDs for each request to assist with tracing.
- Retry using different tools (e.g. Postman, CURL) or from another network to isolate the issue.
- If problems persist, provide full request/response examples and transaction IDs when contacting Service Victoria.