



# **Fair Fuel Open Data API Documentation**

**Document Version 1.0**

**7 Oct 2025**

## Table of Contents

<b>1. DOCUMENT INFORMATION .....</b>	<b>3</b>
<b>2. DOCUMENT REVISION HISTORY .....</b>	<b>4</b>
<b>3. DOCUMENT APPROVAL &amp; DISTRIBUTION .....</b>	<b>5</b>
<b>4. BACKGROUND .....</b>	<b>6</b>
4.1    PURPOSE .....	6
4.2    AUDIENCE .....	7
<b>5. SCOPE SUMMARY .....</b>	<b>8</b>
5.1    IN-SCOPE (INCLUSIONS) .....	8
5.2    OUT-OF-SCOPE (EXCLUSIONS) .....	8
<b>6. API CONDITIONS (REQUIREMENTS) .....</b>	<b>9</b>
6.1    ASSUMPTIONS .....	9
<b>7. API SERVICES .....</b>	<b>10</b>
7.1    OVERVIEW .....	10
7.2    FULL LIST OF OPERATIONS .....	10
7.3    API SECURITY .....	11
7.4    GET FUEL PRICE API .....	11
7.5    GET FUEL STATIONS API .....	17
7.6    GET FUEL BRANDS API .....	22
7.7    GET FUEL TYPES API .....	25
<b>8. TROUBLESHOOTING GUIDE .....</b>	<b>28</b>
8.1    COMMON SCENARIOS .....	28
8.2    TROUBLESHOOTING CHECKLIST FOR API USERS .....	28
8.3    ADDITIONAL TIPS .....	29

## 1. Document information

Criteria	Details
<b>Document title:</b>	Fair Fuel Open Data API Documentation
<b>Document owner:</b>	Service Victoria
<b>Document author:</b>	Service Victoria
<b>Document Version:</b>	1.0
<b>Issue date:</b>	7 Oct 2025

## 2. Document revision history

Document Version	Date	Description
0.1	5 Sep 2025	Initial draft prepared based on internal OpenAPI specifications.
1.0	7 Oct 2025	Final approved release for production use.

### 3. Document Approval & Distribution

Document Version	Action	Date
0.1	Approved	4 Sep 2025
0.1	Reviewer – feedback provided	4 Sep 2025
0.1	Reviewer- feedback provided	4 Sep 2025
0.1	Reviewer	4 Sep 2025
0.1	Reviewer	4 Sep 2025
1.0	Approved	7 Oct 2025
1.0	Approved	9 Oct 2025
1.0	Approved	9 Oct 2025

## 4. Background

This document outlines how consumers can access fuel price data from Service Victoria using the Fair Fuel Open Data API. The Open Data API is a RESTful interface that allows the public, analysts, and third-party application developers to programmatically retrieve fuel prices for fuel stations across Victoria.

The Open Data API differs from the fuel retailer submission API in that it publishes data with a 24-hour delay. This ensures that sensitive commercial information submitted by fuel retailers is securely managed, while still providing a comprehensive dataset for public transparency, research, and innovation.

This document is intended to assist developers, analysts, and organisations in integrating with the Open Data API to retrieve fuel station and fuel price information in a structured and consistent format. It provides the technical details required to request and process the dataset, including fuel station metadata, fuel types, and associated prices.

### 4.1 Purpose

This document provides information about using the Fair Fuel Open Data API developed by Service Victoria. The API allows authorised consumers to retrieve all fuel station and fuel price information available through the Service Victoria platform.

The Open Data API is designed to:

- Provide transparent access to statewide fuel pricing information.
- Support third-party innovation such as comparison apps, dashboards, and analytic services.
- Deliver a standardised, machine-readable dataset for consistent reuse.
- Reduce barriers to information access by removing the need for individual data-sharing agreements.

Access to the Fair Fuel Open Data API is gated by a registration process using the Service Victoria platform. Approved consumers receive an API Consumer ID to authenticate requests. Use is read-only, monitored for fair-use compliance, and subject to rate limits and Service Victoria's terms of use.

It is expected that this API documentation will include and elaborate on the following information to support integration with the Open Data feed:

- Base path and structure of the API endpoint.
- Required request headers (including consumer ID and transaction ID).
- Definitions of the response fields, including station details, fuel types, and price values.
- Explanation of data freshness (prices delayed ~24 hours).
- Expected formats for requests and responses.

- Examples of request and response payloads (or references to these examples).

This document is structured to present this information clearly and consistently to help developers, analysts, and third-party vendors understand the Open Data API and support successful use in applications, research, and reporting.

## 4.2 Audience

This documentation is written for:

- **Application developers** who wish to integrate Victorian fuel price data into their products, services, or research tools.
- **Data analysts and researchers** seeking access to statewide fuel price information for reporting, analysis, or transparency purposes.
- **Technical solution architects** are responsible for designing systems that consume or process Open Data feeds.

## 5. Scope Summary

### 5.1 In-Scope (Inclusions)

This system is designed to provide public access to Victorian fuel price information through the Fair Fuel Open Data API. The API enables consumers, developers, and analysts to programmatically retrieve details of all fuel stations and their associated fuel prices across Victoria.

The dataset includes:

- Fuel Station details (name, address, brand, and location).
- Available fuel types and their prices.
- Indicators of price availability.
- Timestamps for the most recent updates.

The Open Data API publishes this information with a 24-hour delay to ensure secure handling of retailer-submitted data while supporting transparency, research, and innovation.

### 5.2 Out-of-Scope (Exclusions)

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

- Real-time or live fuel price updates
- Any other API services provided by Service Victoria, unrelated to fuel pricing.
- Write or update operations (the Open Data API is strictly read-only).
- Authentication and security features required for fuel retailer submissions.
- Activities or requests not specifically defined as in-scope within this documentation.

## 6. API Conditions (Requirements)

### 6.1 Assumptions

Ref#	Assumptions
A1	This functionality assumes that fuel retailers will continue to submit price and availability data via the Fair Fuel retailer submission API or Price Reporting Web Portal. The Open Data API consumes this information and republishes it with a 24-hour delay.
A2	Each Open Data API response will include valid fuel station identifiers, fuel types, and the most recent fuel price information available following the ~24-hour publication delay from fuel retailer submissions.
A3	Consumers are expected to provide a valid <b>x-consumer-id</b> and unique <b>x-transactionid</b> with each request to ensure traceability and service monitoring.
A4	Fuel price data published through the Open Data API is not real-time. Prices will be at least 24 hours old, reflecting the most recent retailer submissions processed by Service Victoria.

## 7. API Services

### 7.1 Overview

This API specification supports the requirement for public consumers to access fuel price data for all registered fuel stations across Victoria. Unlike the fuel retailer submission API, the Open Data API is read-only and publishes price information with a 24-hour delay.

This API has been designed to allow consistent, structured retrieval of statewide fuel station and fuel price data. All requests must conform to the defined schema and must include the required request headers (**x-consumer-id** and **x-transactionid**) for traceability.

### 7.2 Full List of Operations

Environment	Base URL
Production	<a href="https://api.fuel.service.vic.gov.au/open-data/v1">https://api.fuel.service.vic.gov.au/open-data/v1</a>

Action	End Point	Description
GET	<code>{{baseUrl}}/fuel/prices</code>	Retrieves all available fuel station price data across Victoria, published with a ~24-hour delay. Each record contains station metadata, fuel types, prices (cents per litre), availability, and the last updated timestamp.
GET	<code>{{baseUrl}}/fuel/reference-data/stations</code>	Returns reference metadata for all registered fuel stations, including station ID, name and address.
GET	<code>{{baseUrl}}/fuel/reference-data/types</code>	Returns the list of <b>supported fuel type codes</b> used in the fuel price.

## 7.3 API Security

Access to the Fair Fuel Open Data API is controlled using the following security mechanisms:

### 1. Consumer Identifier (x-consumer-id):

- Each approved consumer of the Open Data API will be issued a unique **Consumer ID** by Service Victoria.
- This must be included in all requests via the ‘x-consumer-id’ header.
- The Consumer ID is used to identify and monitor API usage, apply rate limits, and ensure compliance with fair-use policies.

### 2. Transport Security:

All requests are only accepted over HTTPS. Connections over HTTP are not supported.

## 7.4 Get Fuel Price API

### 7.4.1 API request

Action	End Point	Description
GET	<a href="https://api.fuel.service.vic.gov.au/open-data/v1/fuel/prices">https://api.fuel.service.vic.gov.au/open-data/v1/fuel/prices</a>	Returns all Victorian fuel stations and their 24-hour delayed prices. Each item includes fuel station metadata, a list of fuel types, prices (cents per litre) and timestamps.

#### Headers:

Field	Type	Example	Comments
User-Agent	string	MyApp/1.0	<b>Required.</b> Identifies the client application making the request
x-consumer-id	String	8f9qWOLO3Ja14sXilPBhn39YnRCZQtXjNFUDm9k7	<b>Required.</b> Issued by Service Victoria for each consumer; used for identification, monitoring and rate limits.
x-transactionid	string (UUID)	960e8410-e29b-42d4-d716-576655440000	<b>Required.</b> Used for tracing/logging. Consumers 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.

### 7.4.2 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.
fuelStation.name	string	fuelPriceDetails[].fuelStation.name	Main Street Fuel	Display name of the fuel station.
fuelStation.brandId	string	fuelPriceDetails[].fuelStation.brandId	testb001	Unique identifier for the fuel brand.
fuelStation.address	string	fuelPriceDetails[].fuelStation.address	123 Main St, Melbourne 3000 VIC	Physical Address of the fuel station location.
fuelStation.contactPhone	string	fuelPriceDetails[].fuelStation.contactPhone	03 9999 0000	Contact Phone Number of the fuel station.
fuelStation.location.latitude	number	fuelPriceDetails[].fuelStation.location.latitude	-37.8136	Latitude coordinate of the fuel station.
fuelStation.location.longitude	number	fuelPriceDetails[].fuelStation.location.longitude	144.9631	Longitude coordinate of 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 most recent price change for this fuel type.
updatedAt	string (ISO 8601 datetime)	fuelPriceDetails[].updatedAt	2025-05-22T01:30:00Z	The date and time of the most recent change to this record (station or price data).

#### 7.4.3 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 Open Data Fuel 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	x-consumer-id is missing or invalid or you're using the old key which is now invalid after you have requested a new key. Recommendation: Do not retry. Confirm header 'x-consumer-id' value.
429	Rate limiting / Too many requests	The request exceeded the API rate limit. <b>Recommendation:</b> 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 Open Data API within timeout window. Recommendation: Retry 2 times with delay.

#### 7.4.4 Sample Example

##### Example – Fuel Stations with Different Fuel Types

Successful Response Payload (HTTP 200 OK)

```
{
  "fuelPriceDetails": [
    {
      "fuelStation": {
        "id": "a019r00000iRqPOAA0",
        "name": "Main Street Fuel",
        "brandId": "testb001",
        "address": "123 Main St, Melbourne VIC 3000",
        "contactPhone": "03 9999 0000",
        "location": {
          "latitude": -37.8136,
          "longitude": 144.9631
        }
      },
      "fuelPrices": [
        {
          "fuelType": "U91",
          "price": 188.8,
          "isAvailable": true,
          "updatedAt": "2025-08-24T11:35:22Z"
        },
        {
          "fuelType": "P95",
          "price": 195.0,
          "isAvailable": true,
          "updatedAt": "2025-08-24T11:35:22Z"
        },
        {
          "fuelType": "P98",
          "price": 209.0,
          "isAvailable": true,
          "updatedAt": "2025-08-24T11:35:22Z"
        },
        {
          "fuelType": "DSL",
          "price": 198.0,
          "isAvailable": true,
          "updatedAt": "2025-08-24T11:35:22Z"
        }
      ],
    }
  ]
}
```

```

    "updatedAt": "2025-08-24T11:35:22Z"
},
{
  "fuelStation": {
    "id": "a019r00000iRqPXBB1",
    "name": "Footscray Service Centre",
    "brandId": "testb002",
    "address": "1 McNab Ave, Footscray VIC 3011",
    "contactPhone": "03 8888 1111",
    "location": {
      "latitude": -37.8010,
      "longitude": 144.9006
    }
  },
  "fuelPrices": [
    {
      "fuelType": "U91",
      "price": 185.5,
      "isAvailable": true,
      "updatedAt": "2025-08-24T10:50:10Z"
    },
    {
      "fuelType": "E10",
      "price": 179.9,
      "isAvailable": true,
      "updatedAt": "2025-08-24T10:50:10Z"
    },
    {
      "fuelType": "DSL",
      "price": 192.0,
      "isAvailable": false,
      "updatedAt": "2025-08-24T10:50:10Z"
    },
    {
      "fuelType": "LPG",
      "price": 99.9,
      "isAvailable": true,
      "updatedAt": "2025-08-24T10:50:10Z"
    }
  ],
  "updatedAt": "2025-08-24T10:50:10Z"
}
]
}

```

#### 7.4.5 Field-Level Schema (OpenAPI Format)

```

components:
schemas:

```

```

FuelPriceDetails:
  type: object
  description: Price snapshot for a single fuel station.
  required: [fuelStation, fuelPrices, updatedAt]
  properties:
    fuelStation:
      $ref: '#/components/schemas/FuelStation'
    fuelPrices:
      type: array
      items:
        $ref: '#/components/schemas/FuelPriceItem'
    updatedAt:
      type: string
      format: date-time
      description: The most recent change to this station or any of its
prices (ISO 8601).

FuelPriceItem:
  type: object
  required: [fuelType, price, isAvailable, updatedAt]
  properties:
    fuelType:
      type: string
      description: Fuel type code.
      enum: [U91, P95, P98, DSL, PDSL, E10, E85, B20, LPG, LNG, CNG]
    price:
      type: number
      nullable: true
      minimum: 0.0
      maximum: 9999.9
      multipleOf: 0.1
      description: Price per litre in **AUD cents**. One decimal place
(e.g., 165.0 = $1.65).
    isAvailable:
      type: boolean
      description: Whether this fuel type is currently available at the
station.
    updatedAt:
      type: string
      format: date-time
      description: Time this fuel price was last updated (ISO 8601).

FuelStation:
  type: object
  required: [id, name, brandId, address, location]
  properties:
    id:
      type: string
      description: Unique identifier for the fuel station.
      maxLength: 50
    name:
      type: string
      description: Display name of the fuel station.
      maxLength: 100
    brandId:
  
```

```

    type: string
    description: ID of the associated brand.
    maxLength: 50
  address:
    type: string
    description: Physical address of the fuel station.
    maxLength: 400
  contactPhone:
    type: string
    nullable: true
    description: Contact phone number.
    maxLength: 20
  location:
    type: object
    required: [latitude, longitude]
    properties:
      latitude:
        type: number
        nullable: true
        minimum: -90
        maximum: 90
        description: Latitude.
      longitude:
        type: number
        nullable: true
        minimum: -180
        maximum: 180
        description: Longitude.
  
```

## 7.5 Get Fuel Stations API

### 7.5.1 API request

Action	End Point	Description
GET	<a href="https://api.fuel.service.vic.gov.au/open-data/v1/fuel/reference-data/stations">https://api.fuel.service.vic.gov.au/open-data/v1/fuel/reference-data/stations</a>	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
User-Agent	string	MyApp/1.0	<b>Required.</b> Identifies the client application making the request

x-consumer-id	String	8f9qWOLO3Ja14sXiPBhn39YnRCZQtXjNFUDm9k7	<b>Required.</b> Issued by Service Victoria for each consumer; used for identification, monitoring and rate limits.
x-transactionid	string (UUID)	960e8410-e29b-42d4-d716-576655440000	<b>Required.</b> 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.

## 7.5.2 API Response

HTTP Response – 200 OK

### Payload:

Field	Type	Path	Example	Comments
fuelStations	array of objects	fuelStations	See below	List of fuel stations.
id	string	fuelStations[].id	a019r00000iRqP0AA0	Unique identifier for the fuel station.
name	string	fuelStations[].name	Main Street Fuel	Display name of the fuel station.
brandId	string	fuelStations[].brandId	testb001	Unique identifier for the fuel brand.
address	string	fuelStations[].address	123 Main St, Melbourne 3000 VIC	Physical Address of the fuel station location.
contactPhone	string	fuelStations[].contactPhone	03 9999 0000	Contact Phone Number of the fuel station.
updatedAt	string(ISO 8601 datetime)	fuelStations[].updatedAt	2025-05-22T01:30:00Z	The date and time of the most recent change to this record

location.latitude	number	fuelStations[0].location.latitude	-37.8136	Latitude coordinate of the fuel station.
location.longitude	number	fuelStations[0].location.longitude	144.9631	Longitude coordinate of the fuel station.

### 7.5.3 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 Open Data 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	x-consumer-id is missing or invalid. Recommendation: Do not retry. Confirm header 'x-consumer-id' value.
429	Rate limiting / Too many requests	The request exceeded the API rate limit. <b>Recommendation:</b> 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.

### 7.5.4 Sample Example

#### Example – Fuel Stations Metadata

Successful Response Payload (HTTP 200 OK)

```
{
  "fuelStations": [
    {
      "id": "n342f00000tRxP2AA0",
      "name": "Main Street Fuel",
      "brandId": "testb001",
      "address": "123 Main St, Melbourne VIC 3000",
      "contactPhone": "03 9999 0000",
      "updatedAt": "2025-08-24T10:50:10Z",
      "location": {
        "latitude": -37.8136,
        "longitude": 144.9631
      }
    },
    {
      "id": "x874g00000kTyZ8BB1",
      "name": "Footscray Service Centre",
      "brandId": "testb002",
      "address": "1 McNab Ave, Footscray VIC 3011",
      "contactPhone": "03 8888 1111",
      "updatedAt": "2025-08-24T10:50:10Z",
      "location": {
        "latitude": -37.8010,
        "longitude": 144.9006
      }
    }
  ]
}
```

### 7.5.5 Field-Level Schema (OpenAPI Format)

```
components:
  schemas:
    FuelStationsResponse:
      type: object
      required: [fuelStations]
      properties:
        fuelStations:
          type: array
          items:
            $ref: '#/components/schemas/FuelStation'

    FuelStation:
      type: object
      required: [id, name, brandId, address, location]
      properties:
        id:
          type: string
```

```
description: Unique identifier for the fuel station.
maxLength: 50
name:
  type: string
  description: Display name of the fuel station.
  maxLength: 100
brandId:
  type: string
  description: ID of the associated brand.
  maxLength: 50
address:
  type: string
  description: Physical address of the fuel station.
  maxLength: 400
contactPhone:
  type: string
  nullable: true
  description: Contact phone number of the fuel station.
  maxLength: 20
updatedAt:
  type: string
  format: date-time
  description: The date and time of the most recent change.
location:
  type: object
  required: [latitude, longitude]
  properties:
    latitude:
      type: number
      minimum: -90
      maximum: 90
      nullable: true
      description: Latitude coordinate.
    longitude:
      type: number
      minimum: -180
      maximum: 180
      nullable: true
      description: Longitude coordinate.
```

## 7.6 Get Fuel Brands API

### 7.6.1 API request

Action	End Point	Description
GET	<a href="https://api.fuel.service.vic.gov.au/open-data/v1/fuel/reference-data/brands">https://api.fuel.service.vic.gov.au/open-data/v1/fuel/reference-data/brands</a>	Returns the list of <b>fuel brands</b> with their identifiers, including <b>brandId</b> and <b>display name</b> . Use brandId to join against station and price records.

#### Headers:

Field	Type	Example	Comments
User-Agent	string	MyApp/1.0	<b>Required.</b> Identifies the client application making the request
x-consumer-id	String	8f9qWQLO3Ja14sX iIPBhn39YnRCZQt XjNFUDm9k7	<b>Required.</b> Issued by Service Victoria for each consumer; used for identification, monitoring and rate limits.
x-transactionid	string (UUID)	960e8410-e29b-42d4-d716-576655440000	<b>Required.</b> 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.

### 7.6.2 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.

type	string	brands[].type	major	Classification of brand. Two types: major or independent
------	--------	---------------	-------	---

### 7.6.3 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 Open Data 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	x-consumer-id is missing or invalid. Recommendation: Do not retry. Confirm header 'x-consumer-id' value.
429	Rate limiting / Too many requests	The request exceeded the API rate limit. <b>Recommendation:</b> 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.

## 7.6.4 Sample Example

### Example – Fuel Brands

Successful Response Payload (HTTP 200 OK)

```
{  
  "brands": [  
    {  
      "id": "testb001",  
      "name": "TestBrand",  
      "type": "major"  
    }  
  ]  
}
```

## 7.6.5 Field-Level Schema (OpenAPI Format)

```
components:  
schemas:  
  BrandsResponse:  
    type: object  
    required: [brands]  
    properties:  
      brands:  
        type: array  
        items:  
          $ref: "#/components/schemas/Brand"  
  
  Brand:  
    type: object  
    required: [id, name, type]  
    properties:  
      id:  
        type: string  
        maxLength: 50  
        description: Unique identifier for the fuel brand.  
      name:  
        type: string  
        maxLength: 100  
        description: Name of the fuel brand.  
      type:  
        type: string  
        maxLength: 50  
        description: Classification of brand.  
        enum: [major, independent]
```

## 7.7 Get Fuel Types API

### 7.7.1 API request

Action	End Point	Description
GET	<a href="https://api.fuel.service.vic.gov.au/open-data/v1/fuel/reference-data/types">https://api.fuel.service.vic.gov.au/open-data/v1/fuel/reference-data/types</a>	Returns the list of supported fuel type codes used in the fuel price.

**Headers:**

Field	Type	Example	Comments
User-Agent	string	MyApp/1.0	<b>Required.</b> Identifies the client application making the request
x-consumer-id	string	8f9qWOLO3Ja14sXiPBhn39YnRCZQtXjNFUDm9k7	<b>Required.</b> Issued by Service Victoria for each consumer; used for identification, monitoring and rate limits.
x-transactionid	string (UUID)	960e8410-e29b-42d4-d716-576655440000	<b>Required.</b> 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.

### 7.7.2 API Response

HTTP Response – 200 OK

**Payload:**

Field	Type	Path	Example	Comments
fuelTypes	array of objects	brands	See below	List of fuel type entries.
id	string	fuelTypes[].id	U91	Unique identifier for the fuel brand.
name	string	fuelTypes[].name	Unleaded 91	Fuel type name.

### 7.7.3 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 Open Data 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 consumer key is missing or invalid. Recommendation: Do not retry. Confirm credentials.
429	Rate limiting / Too many requests	The request exceeded the API rate limit. <b>Recommendation:</b> 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.

### 7.7.4 Sample Example

#### Example – Fuel Types

Successful Response Payload (HTTP 200 OK)

```
{
  "fuelTypes": [
    {
      "id": "U91",
      "name": "Unleaded 91"
    }
  ]
}
```

### 7.7.5 Field-Level Schema (OpenAPI Format)

```

components:
  schemas:
    FuelTypesResponse:
      type: object
      required: [fuelTypes]
      properties:
        fuelTypes:
          type: array
          items:
            $ref: "#/components/schemas/FuelType"

    FuelType:
      type: object
      required: [id, name]
      properties:
        id:
          type: string
          description: Fuel type code.
          enum: [U91, P95, P98, DSL, PDSL, E10, E85, B20, LPG, LNG, CNG]
        name:
          type: string
          description: Display name of the fuel type.
  
```

### 7.7.6 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

## 8. Troubleshooting Guide

### 8.1 Common Scenarios

This section provides common troubleshooting scenarios and a checklist to assist in diagnosing and resolving integration issues when using the Fair Fuel Open Data APIs.

#### Scenario: Cannot retrieve Fuel Prices Using the Open Data API

Before making the API call, ensure the following conditions are met:

- You have submitted the access request via the Service Victoria Website Form.
- You have received and configured your API consumer key.
- You are using the correct HTTP method and endpoint.
- All required request headers are present:
  - x-consumer-id
  - x-transactionid
  - User-Agent

### 8.2 Troubleshooting Checklist for API Users

Checkpoint	Description
<input checked="" type="checkbox"/> API Consumer ID	Ensure the API Consumer ID is correct.
<input checked="" type="checkbox"/> HTTP method and endpoint	If you are retrieving fuel price, station, brand or fuel types, ensure you are using the HTTP "GET" method.
<input checked="" type="checkbox"/> Headers included	Verify all required headers are included. Common required headers include: <ul style="list-style-type: none"><li>• Transaction ID (X-TransactionID)</li><li>• API Key (x-consumer-id)</li><li>• User Agent (User-Agent)</li></ul>
<input checked="" type="checkbox"/> Response review	Check HTTP status codes and response messages for error details.

**Common error examples:**

HTTP Code	Explanation
400	Bad request.
403	Forbidden. Check your API consumer id.
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

- 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.