



# project44 Multi-modal Ocean and Truckload Tracking API Guide

---

Publication date January 27, 2023



# Table of Contents

1. Introduction .....	5
1.1. Welcome! .....	5
1.2. API .....	5
1.2.1. Authentication .....	5
1.2.2. Supported Data Formats .....	5
2. Create a Shipment .....	6
2.1. Create an Ocean Shipment .....	6
2.1.1. Description .....	6
2.1.2. Structure .....	6
2.1.3. Fields and Objects .....	10
2.1.4. Use Cases .....	11
2.2. Create a Truckload Shipment .....	13
2.2.1. Description .....	13
2.2.2. Structure .....	13
2.2.3. Fields and Objects .....	23
2.3. Link Ocean and Truckload Shipments .....	24
2.3.1. Link a Collection of Shipments that are Transported Using Multiple Modes .....	24
2.3.2. Link a Single Shipment that is Transported Using Multiple Modes .....	24
3. Retrieve Shipment Event History .....	25
3.1. Description .....	25
3.2. Structure .....	25
3.3. Field and Objects .....	28
3.3.1. Shipment Stop Types by Mode .....	29
3.3.2. Route Segment Identifiers by Mode .....	30
3.3.3. Shipment State Types by Mode .....	30
3.3.4. Shipment Event Types by Mode .....	31
4. Retrieve Shipment Position History .....	33
4.1. Description .....	33
4.2. Structure .....	33
4.3. Fields and Objects .....	35
4.3.1. Shipment Stop Types by Mode .....	36
4.3.2. Route Segment Identifiers by Mode .....	37
5. Retrieve Shipment Route Information .....	38
5.1. Description .....	38
5.2. Structure .....	38
5.3. Fields and Objects .....	40
5.3.1. Shipment Stop Types by Mode .....	41
5.3.2. Route Segment Identifiers by Mode .....	42
6. Delete a Shipment .....	43
6.1. Description .....	43
6.2. Structure .....	43
7. PUSH Updates for Shipments .....	44
7.1. Description .....	44
8. Manage Webhooks .....	45
8.1. Description .....	45
8.2. Structure .....	45
8.2.1. Create or Update a PUSH Configuration .....	45
8.2.2. Retrieve a Webhook Configuration by Name .....	45
8.2.3. Retrieve All Webhook Configurations .....	46

8.3. Fields and Objects .....	47
9. Appendix .....	48
9.1. Error Response Codes .....	48
9.2. Track Specific Containers Sample Payload .....	49
9.3. Sample PUSH Update .....	51

# 1. Introduction

## 1.1. Welcome!

Welcome to the project44 platform. Your integration with project44 will allow you to retrieve timely, accurate tracking data for shipments.

This document describes how to write a client application to integrate your TMS or other system to the project44 platform through our Ocean and Truckload Tracking APIs.

## 1.2. API

The project44 API is a RESTful API designed to enable users to work with objects using familiar HTTP verbs as follows:

- read = GET
- create = POST
- delete = DELETE

All production API requests are made to:

- Americas: <https://na12.api.project44.com/api/v4>
- Europe: <https://eu12.api.project44.com/api/v4>

A testing sandbox is available to develop and test applications with requests sent to the link below:

- Americas: <https://na12.api.sandbox.p-44.com/api/v4>

The Core Services API is available to project44 customers and integration partners. All organizations must acquire valid credentials prior to using the web services.

### 1.2.1. Authentication

Authorization is necessary to establish endpoints for project44's APIs. The project44 API offers authentication for client applications over HTTPS through OAuth 2.0 Client Credentials Grant Type. To learn more about this, please visit [the Authentication section in our Developer Portal](#). Most client software provides a mechanism to give a username and password to create the required authentication header automatically.

### 1.2.2. Supported Data Formats

The project44 API uses JSON as the default format. This format is a string representation of a key/value pair where the values can be arrays, strings, integers, or even full objects. The key names are relevant and enable JSON parsers to quickly scan through the records and convert them into objects developers can interact with within the code. Due to the lightweight format, objects and collections can be serialized and deserialized quickly.

## 2. Create a Shipment

### 2.1. Create an Ocean Shipment

#### 2.1.1. Description

The **POST: Initialize a Shipment for Tracking** endpoint is used to create an Ocean shipment in the project44 network. Currently, this endpoint only supports shipment creation of Ocean shipments, not FTL shipments.

#### 2.1.2. Structure

<b>Endpoint</b>	/api/v4/shipments/tracking
<b>Verb</b>	POST
<b>Required fields</b>	<ul style="list-style-type: none"><li>• At least one of the following shipment IDs (<code>identifiers</code> array): <code>BILL_OF_LADING</code>, <code>BOOKING_NUMBER</code>, or <code>CONTAINER_ID</code></li><li>• Exactly one <code>CARRIER_SCAC</code> (in the <code>identifiers</code> array)</li></ul>

## Request schema

```

{
  "id": "string",
  "identifiers": [
    {
      "type": "BILL_OF_LADING",
      "value": "string"
    }
  ],
  "routeInfo": {
    "stops": [
      {
        "id": "string",
        "type": "UNKNOWN",
        "location": {
          "id": "string",
          "name": "string",
          "identifiers": [
            {
              "type": "PORT_UN_LOCODE",
              "value": "string"
            }
          ],
          "address": {
            "postalCode": "string",
            "addressLines": [
              "string"
            ],
            "city": "string",
            "state": "string",
            "country": "US"
          },
          "coordinates": {
            "latitude": 0,
            "longitude": 0
          },
          "timeZone": "string"
        }
      }
    ],
    "routeSegments": [
      {
        "id": "string",
        "fromStopId": "string",
        "toStopId": "string",
        "transportationMode": "OCEAN",
        "identifiers": [
          {
            "type": "BILL_OF_LADING",
            "value": "string"
          }
        ]
      }
    ]
  },
  "attributes": [
    {
      "name": "string",
      "value": "string",
      "values": [
        "string"
      ]
    }
  ],
  "relatedShipments": [
    {
      "id": "string",
      "identifiers": [
        {

```

```

        "type": "BILL_OF_LADING",
        "value": "string"
    }
],
"attributes": [
    {
        "name": "string",
        "value": "string",
        "values": [
            "string"
        ]
    }
],
"shipmentShareLink": "string",
"createdDateTime": "string",
"lastModifiedDateTime": "string"
}
],
"events": [
    {
        "dateTime": "string",
        "receivedDateTime": "string",
        "type": "UNKNOWN",
        "description": "string",
        "stopId": "string",
        "routeSegmentId": "string",
        "estimateDateTime": "string",
        "estimateLastCalculatedDateTime": "string",
        "plannedDateTime": "string",
        "plannedEndDateTime": "string"
    }
],
"shipmentShareLink": "string",
"createdDateTime": "string",
"lastModifiedDateTime": "string"
}

```

**Expected HTTP response code**

200 OK



## Response schema

```

{
  "id": "string",
  "identifiers": [
    {
      "type": "BILL_OF_LADING",
      "value": "string"
    }
  ],
  "routeInfo": {
    "stops": [
      {
        "id": "string",
        "type": "UNKNOWN",
        "location": {
          "id": "string",
          "name": "string",
          "identifiers": [
            {
              "type": "PORT_UN_LOCODE",
              "value": "string"
            }
          ],
          "address": {
            "postalCode": "string",
            "addressLines": [
              "string"
            ],
            "city": "string",
            "state": "string",
            "country": "US"
          },
          "coordinates": {
            "latitude": 0,
            "longitude": 0
          },
          "timeZone": "string"
        }
      }
    ],
    "routeSegments": [
      {
        "id": "string",
        "fromStopId": "string",
        "toStopId": "string",
        "transportationMode": "OCEAN",
        "identifiers": [
          {
            "type": "BILL_OF_LADING",
            "value": "string"
          }
        ]
      }
    ]
  },
  "attributes": [
    {
      "name": "string",
      "value": "string",
      "values": [
        "string"
      ]
    }
  ],
  "relatedShipments": [
    {
      "id": "string",
      "identifiers": [
        {

```

```

        "type": "BILL_OF_LADING",
        "value": "string"
    }
  ],
  "attributes": [
    {
      "name": "string",
      "value": "string",
      "values": [
        "string"
      ]
    }
  ],
  "shipmentShareLink": "string",
  "createdDateTime": "string",
  "lastModifiedDateTime": "string"
}
],
"events": [
  {
    "dateTime": "string",
    "receivedDateTime": "string",
    "type": "UNKNOWN",
    "description": "string",
    "stopId": "string",
    "routeSegmentId": "string",
    "estimateDateTime": "string",
    "estimateLastCalculatedDateTime": "string",
    "plannedDateTime": "string",
    "plannedEndDateTime": "string"
  }
],
"shipmentShareLink": "string",
"createdDateTime": "string",
"lastModifiedDateTime": "string"
}

```

Error HTTP Response Codes	
400	Invalid request
401	Invalid or missing credentials
403	User not authorized to perform this operation

### 2.1.3. Fields and Objects

The table below describes only the minimum relevant fields of the API Request to create a shipment for tracking. These fields are also returned with the API Response 200 OK.

**Table 1. Minimum Relevant Fields**

Field	Description
id	The Master Shipment ID of the shipment. This is returned upon shipment creation but is not a required field to initiate. This ID is stored for future references to the shipment within project44 APIs and for identification of subsequent messages pushed to the client webhook.
identifiers	This array contains the shipment's identifiers.

Field	Description
<code>identifiers.type</code>	<p><b>Field Type:</b> string</p> <p>The type of identifier. Must be at least one of:</p> <ul style="list-style-type: none"> <li><code>BILL_OF_LADING</code>: BOLs are unique identifiers given to a shipper from the ocean carrier. These identifiers often contain multiple containers to enable a shipper to provide the correct BOL to track multiple containers.</li> <li><code>BOOKING_NUMBER</code>: BNs are unique identifiers given to a shipper from the ocean carrier. BNs often contain multiple containers. Using the BN enables shippers to track multiple containers. This document will refer to BOLs/BNs interchangeably.</li> <li><code>CONTAINER_ID</code>: Container IDs are permanent IDs given to a container and are reused from shipper to shipper. To prevent the same ID usage for multiple trips and shippers, it is recommended that supply chain personnel initiate the use of BOLs or BNs.</li> </ul> <p>AND exactly one <code>CARRIER_SCAC</code>.</p>
<code>identifiers.value</code>	<p><b>Field Type:</b> string</p> <p>The value of the identifier.</p>

### 2.1.4. Use Cases

This section describes instances where a user wants to use the POST action to complete tasks other than initializing tracking for an Ocean Shipment:

- [Include Additional Shipment Identifiers](#)
- [Track Specific Containers](#)
- [Update Shipment Attributes](#)

#### **Include Additional Shipment Identifiers**

Add identifiers to a shipment that has already been created. This update to the shipment is done using a POST. This action adds information to the shipment and never removes or replaces previous shipment data. Users do not need to use a GET before updating a shipment's information.

<b>Precondition</b>	<p>Have the following required information:</p> <ul style="list-style-type: none"> <li>• The Master Shipment ID (id) returned during the shipment's creation.</li> <li>• The new identifiers to add to the shipment.</li> </ul>
<b>Workflow</b>	<ol style="list-style-type: none"> <li>1. Call the POST <code>/api/v4/shipments/tracking</code> endpoint.</li> <li>2. Add the new shipment identifiers.</li> <li>3. Submit the request.</li> </ol>
<b>Postcondition</b>	The request is submitted and confirmed when a 200 OK response code is received.
<b>Exception Workflow</b>	<p>400 Invalid request</p> <p>401 Invalid or missing credentials</p> <p>403 User not authorized to perform this operation</p> <p>See <a href="#">Error Responses</a> for more information on the meaning of these error codes.</p>

#### **Track Specific Containers**

Track specific containers related to a BOL or BN by adding the container IDs when creating the shipment. Even if there are more containers than listed in the payload, only the containers added during shipment creation will have shipments created for them.

<b>Precondition</b>	<p>Have the following required information:</p> <ul style="list-style-type: none"> <li>• Shipment BOL or BN</li> <li>• Carrier SCAC</li> <li>• Container IDs for the specific containers to be added to this shipment.</li> </ul>
<b>Workflow</b>	<ol style="list-style-type: none"> <li>1. Call the POST /api/v4/shipments/tracking endpoint.</li> <li>2. Add the shipment identifier, carrier SCAC, and container IDs for relevant containers. See the sample payload here: <pre> {   "identifiers": [     {       "type": "BILL_OF_LADING",       "value": "MEDUCZ810176"     },     {       "type": "CARRIER_SCAC",       "value": "MSC"     },     {       "type": "CONTAINER_ID",       "value": "MEDU7265302"     },     {       "type": "CONTAINER_ID",       "value": "TGHU9088638"     }   ] } </pre> </li> <li>3. Submit the request.</li> </ol>
<b>Postcondition</b>	The request is submitted and confirmed when a 200 OK response code is received.
<b>Exception Workflow</b>	<p>400 Invalid request</p> <p>401 Invalid or missing credentials</p> <p>403 User not authorized to perform this operation</p> <p>See <a href="#">Error Responses</a> for more information on the meaning of these error codes.</p>

### Update Shipment Attributes

Change the shipment attributes added during shipment creation by deleting the shipment and then recreating the shipment. See [Section 3. API Request to Delete Ocean Shipments](#) for more information about the requirements for deleting a shipment.

**NOTE:** project44’s system supports updating shipments through a DELETE and POST only three times. This prevents accidental creation of the same shipment too many times.

<b>Precondition</b>	<p>Have the following required information to DELETE the shipment:</p> <ul style="list-style-type: none"> <li>• Shipment ID</li> </ul> <p>Have the following required information to recreate the shipment with the correct attributes:</p> <ul style="list-style-type: none"> <li>• Shipment BOL, BN, or Container ID</li> <li>• Carrier SCAC</li> <li>• Shipment Attributes</li> </ul>
<b>Workflow</b>	<ol style="list-style-type: none"> <li>1. Call the DELETE /api/v4/shipments/{shipmentId}/tracking endpoint. This will return a 200 OK response if successful.</li> <li>2. Call the POST /api/v4/shipments/tracking endpoint.</li> <li>3. Add the shipment identifier and carrier SCAC.</li> <li>4. Add the new shipment attributes.</li> <li>5. Submit the request.</li> </ol>

<b>Postcondition</b>	The request is submitted and confirmed when a 200 OK response code is received after the POST.
<b>Exception Workflow</b>	<p>400 Invalid request</p> <p>401 Invalid or missing credentials</p> <p>403 User not authorized to perform this operation</p> <p>See <a href="#">Error Responses</a> for more information on the meaning of these error codes.</p>

## 2.2. Create a Truckload Shipment

### 2.2.1. Description

The **POST: Start Tracking a Truckload Shipment** endpoint is used to create a Truckload shipment in the project44 network.

### 2.2.2. Structure

<b>Endpoint</b>	/api/v4/tl/shipments
<b>Verb</b>	POST
<b>Required fields</b>	<ul style="list-style-type: none"> <li>• A carrier identifier (<i>carrierIdentifier</i> array): SCAC, DOT_NUMBER, MC_NUMBER, P44_EU, or SYSTEM</li> <li>• At least one shipment ID (<i>shipmentIdentifiers</i> array): BILL_OF_LADING or ORDER</li> <li>• At least two shipment stops (<i>shipmentStops</i> array) including the stop number, appointment start and end dates and times, and location (or geocoordinates) information</li> <li>• An equipment identifier (<i>equipmentIdentifiers</i> array): MOBILE_PHONE_NUMBER, VEHICLE_ID, or LICENSE_PLATE</li> </ul>

## Request schema

```

{
  "carrierIdentifier": {
    "type": "SCAC",
    "value": "string"
  },
  "shipmentIdentifiers": [
    {
      "type": "BILL_OF_LADING",
      "value": "string"
    }
  ],
  "shipmentStops": [
    {
      "stopNumber": 0,
      "appointmentWindow": {
        "startDateTime": "2023-01-27T16:38:51",
        "endDateTime": "2023-01-28T16:38:51",
        "localTimeZoneIdentifier": "string"
      },
      "additionalAppointmentWindows": [
        {
          "type": "STRATEGIC",
          "appointmentWindow": {
            "startDateTime": "2023-01-27T16:38:51",
            "endDateTime": "2023-01-28T16:38:51",
            "localTimeZoneIdentifier": "string"
          },
          "appointmentStatus": "PENDING",
          "identifiers": [
            {
              "type": "DOCK",
              "value": "string"
            }
          ]
        }
      ]
    }
  ],
  "location": {
    "address": {
      "postalCode": "string",
      "addressLines": [
        "string"
      ],
      "city": "string",
      "state": "string",
      "country": "US"
    },
    "contact": {
      "companyName": "string",
      "contactName": "string",
      "phoneNumber": "string",
      "phoneNumberCountryCode": "string",
      "phoneNumber2": "string",
      "phoneNumber2CountryCode": "string",
      "email": "string",
      "faxNumber": "string",
      "faxNumberCountryCode": "string"
    },
    "locationIdentifiers": [
      {
        "type": "EXTERNAL",
        "value": "string"
      }
    ],
    "stopName": "string",
    "geoCoordinates": {
      "latitude": 0,
      "longitude": 0
    }
  }
}

```

```

    },
    "stopType": "PICKUP",
    "stopDescription": "string",
    "involvedParties": [
      {
        "partyIdentifiers": [
          {
            "type": "ACCOUNT",
            "value": "string"
          }
        ]
      }
    ],
    "uuid": "string"
  }
],
"capacityProviderAccountGroup": {
  "code": "string",
  "accounts": [
    {
      "code": "string"
    }
  ]
},
"equipmentIdentifiers": [
  {
    "type": "MOBILE_PHONE_NUMBER",
    "value": "string",
    "shouldDelete": false
  }
],
"sensors": [
  {
    "identifier": {
      "type": "TEMPERATURE",
      "value": "string"
    },
    "min": 0,
    "target": 0,
    "max": 0
  }
],
"shippingDetails": {
  "multipleDrivers": false,
  "truckDetails": {
    "truckDimensions": {
      "length": 0,
      "width": 0,
      "height": 0,
      "unitOfMeasure": "IN"
    },
    "weight": 0,
    "weightUnitOfMeasure": "LB",
    "truckType": "AUTO_CARRIER"
  },
  "hazmatDetails": {
    "hazardClasses": [
      "string"
    ]
  }
},
"shipmentDetails": {
  "lineItems": [
    {
      "handlingUnitQuantity": 0,
      "handlingUnitPackageType": "PLT",
      "pickupStopNumber": 0,
      "deliveryStopNumber": 0,
      "subLineItems": [

```

```

    {
      "description": "string",
      "freightClass": "50",
      "itemIdentifiers": [
        {
          "type": "PURCHASE_ORDER",
          "value": "string"
        }
      ],
      "monetaryValue": {
        "currency": "USD",
        "amount": 0
      },
      "totalWeight": {
        "weight": 0,
        "weightUnit": "LB"
      },
      "contact": {
        "companyName": "string",
        "contactName": "string",
        "phoneNumber": "string",
        "phoneNumberCountryCode": "string",
        "phoneNumber2": "string",
        "phoneNumber2CountryCode": "string",
        "email": "string",
        "faxNumber": "string",
        "faxNumberCountryCode": "string"
      },
      "involvedParties": [
        {
          "partyIdentifiers": [
            {
              "type": "ACCOUNT",
              "value": "string"
            }
          ]
        }
      ],
      "itemQuantity": 0,
      "itemPackageType": "BAG",
      "commodityCode": "string"
    }
  ],
  "description": "string",
  "freightClass": "50",
  "itemIdentifiers": [
    {
      "type": "PURCHASE_ORDER",
      "value": "string"
    }
  ],
  "monetaryValue": {
    "currency": "USD",
    "amount": 0
  },
  "totalWeight": {
    "weight": 0,
    "weightUnit": "LB"
  },
  "contact": {
    "companyName": "string",
    "contactName": "string",
    "phoneNumber": "string",
    "phoneNumberCountryCode": "string",
    "phoneNumber2": "string",
    "phoneNumber2CountryCode": "string",
    "email": "string",
    "faxNumber": "string",
    "faxNumberCountryCode": "string"
  }
}

```



```

    },
    "involvedParties": [
      {
        "partyIdentifiers": [
          {
            "type": "ACCOUNT",
            "value": "string"
          }
        ]
      }
    ],
    "itemQuantity": 0,
    "itemPackageType": "BAG",
    "commodityCode": "string"
  }
],
"trackingDetails": [
  {
    "equipmentIdentifier": {
      "type": "MOBILE_PHONE_NUMBER",
      "value": "string",
      "shouldDelete": false
    },
    "equipmentIdentifierSource": "CUSTOMER",
    "trackingType": "MOBILE_PHONE"
  }
],
"apiConfiguration": {
  "fallBackToDefaultAccountGroup": false,
  "webhookEndpointConfigName": "string",
  "enableIntelligentMatching": false
},
"attributes": [
  {
    "predefined": true,
    "name": "string",
    "value": "string",
    "values": [
      "string"
    ]
  }
],
"id": 0,
"shipmentLegId": "string",
"masterShipmentId": "string",
"idleDuration": 0,
"equipmentIdentifierChanges": [
  {
    "type": "MOBILE_PHONE_NUMBER",
    "value": "string",
    "timestamp": "string"
  }
],
"pointsOfInterest": [
  {
    "location": {
      "address": {
        "postalCode": "string",
        "addressLines": [
          "string"
        ],
        "city": "string",
        "state": "string",
        "country": "US"
      },
      "contact": {
        "companyName": "string",
        "contactName": "string",

```

```
    "phoneNumber": "string",
    "phoneNumberCountryCode": "string",
    "phoneNumber2": "string",
    "phoneNumber2CountryCode": "string",
    "email": "string",
    "faxNumber": "string",
    "faxNumberCountryCode": "string"
  }
},
"locationIdentifiers": [
  {
    "type": "EXTERNAL",
    "value": "string"
  }
],
"pointOfInterestName": "string",
"geoCoordinates": {
  "latitude": 0,
  "longitude": 0
},
"pointOfInterestType": "GENERAL",
"pointOfInterestDescription": "string",
"pointOfInterestId": "string",
"shouldDelete": false
}
]
}
```

**Expected HTTP response code**

201 Created

## Response schema

```

{
  "shipment": {
    "carrierIdentifier": {
      "type": "strong",
      "value": "string"
    },
    "shipmentIdentifiers": [
      {
        "type": "string",
        "value": "string"
      }
    ],
    "shipmentStops": [
      {
        "stopNumber": 0,
        "appointmentWindow": {
          "startDateTime": "yyyy-MM-ddTHH:mm:ss",
          "endDateTime": "yyyy-MM-ddTHH:mm:ss",
          "localTimeZoneIdentifier": "string"
        },
        "additionalAppointmentWindows": [
          {
            "type": "string",
            "appointmentWindow": {
              "startDateTime": "yyyy-MM-DDTHH:mm:ss",
              "endDateTime": "yyyy-MM-DDTHH:mm:ss",
              "localTimeZoneIdentifier": "string"
            },
            "appointmentStatus": "string",
            "identifiers": [
              {
                "type": "string",
                "value": "string"
              }
            ]
          }
        ],
        "location": {
          "address": {
            "postalCode": "string",
            "addressLines": [
              "string"
            ],
            "city": "string",
            "state": "string",
            "country": "string"
          },
          "contact": {
            "companyName": "string",
            "contactName": "string",
            "phoneNumber": "string",
            "phoneNumberCountryCode": "string",
            "phoneNumber2": "string",
            "phoneNumber2CountryCode": "string",
            "email": "string",
            "faxNumber": "string",
            "faxNumberCountryCode": "string"
          }
        },
        "locationIdentifiers": [
          {
            "type": "string",
            "value": "string"
          }
        ],
        "stopName": "string",
        "geoCoordinates": {
          "latitude": 0,

```

```

        "longitude": 0
      },
      "stopType": "string",
      "stopDescription": "string",
      "involvedParties": [
        {
          "partyIdentifiers": [
            {
              "type": "string",
              "value": "string"
            }
          ]
        }
      ],
      "uuid": "string"
    }
  ],
  "capacityProviderAccountGroup": {
    "code": "string",
    "accounts": [
      {
        "code": "string"
      }
    ]
  },
  "equipmentIdentifiers": [
    {
      "type": "string",
      "value": "string",
      "shouldDelete": false
    }
  ],
  "sensors": [
    {
      "identifier": {
        "type": "string",
        "value": "string"
      },
      "min": 0,
      "target": 0,
      "max": 0
    }
  ],
  "shippingDetails": {
    "multipleDrivers": false,
    "truckDetails": {
      "truckDimensions": {
        "length": 0,
        "width": 0,
        "height": 0,
        "unitOfMeasure": "string"
      },
      "weight": 0,
      "weightUnitOfMeasure": "string",
      "truckType": "string"
    },
    "hazmatDetails": {
      "hazardClasses": [
        "string"
      ]
    }
  },
  "shipmentDetails": {
    "lineItems": [
      {
        "handlingUnitQuantity": 0,
        "handlingUnitPackageType": "string",
        "pickupStopNumber": 0,
        "deliveryStopNumber": 0,

```

```

"subLineItems": [
  {
    "description": "string",
    "freightClass": "string",
    "itemIdentifiers": [
      {
        "type": "string",
        "value": "string"
      }
    ],
    "monetaryValue": {
      "currency": "string",
      "amount": 0
    },
    "totalWeight": {
      "weight": 0,
      "weightUnit": "string"
    },
    "contact": {
      "companyName": "string",
      "contactName": "string",
      "phoneNumber": "string",
      "phoneNumberCountryCode": "string",
      "phoneNumber2": "string",
      "phoneNumber2CountryCode": "string",
      "email": "string",
      "faxNumber": "string",
      "faxNumberCountryCode": "string"
    },
    "involvedParties": [
      {
        "partyIdentifiers": [
          {
            "type": "string",
            "value": "string"
          }
        ]
      }
    ],
    "itemQuantity": 0,
    "itemPackageType": "string",
    "commodityCode": "string"
  }
],
"description": "string",
"freightClass": "string",
"itemIdentifiers": [
  {
    "type": "string",
    "value": "string"
  }
],
"monetaryValue": {
  "currency": "string",
  "amount": 0
},
"totalWeight": {
  "weight": 0,
  "weightUnit": "string"
},
"contact": {
  "companyName": "string",
  "contactName": "string",
  "phoneNumber": "string",
  "phoneNumberCountryCode": "string",
  "phoneNumber2": "string",
  "phoneNumber2CountryCode": "string",
  "email": "string",
  "faxNumber": "string",

```

```

        "faxNumberCountryCode": "string"
    },
    "involvedParties": [
        {
            "partyIdentifiers": [
                {
                    "type": "string",
                    "value": "string"
                }
            ]
        }
    ],
    "itemQuantity": 0,
    "itemPackageType": "string",
    "commodityCode": "string"
}
]
},
"trackingDetails": [
    {
        "equipmentIdentifier": {
            "type": "string",
            "value": "string",
            "shouldDelete": false
        },
        "equipmentIdentifierSource": "string",
        "trackingType": "string"
    }
],
"apiConfiguration": {
    "fallBackToDefaultAccountGroup": false,
    "webhookEndpointConfigName": "string",
    "enableIntelligentMatching": false
},
"attributes": [
    {
        "predefined": true,
        "name": "string",
        "value": "string",
        "values": [
            "string"
        ]
    }
],
"id": 0,
"shipmentLegId": "string",
"masterShipmentId": "string",
"idleDuration": 0,
"equipmentIdentifierChanges": [
    {
        "type": "string",
        "value": "string",
        "timestamp": "string"
    }
],
"pointsOfInterest": [
    {
        "location": {
            "address": {
                "postalCode": "string",
                "addressLines": [
                    "string"
                ],
                "city": "string",
                "state": "string",
                "country": "string"
            },
            "contact": {
                "companyName": "string",

```

```

        "contactName": "string",
        "phoneNumber": "string",
        "phoneNumberCountryCode": "string",
        "phoneNumber2": "string",
        "phoneNumber2CountryCode": "string",
        "email": "string",
        "faxNumber": "string",
        "faxNumberCountryCode": "string"
    },
    "locationIdentifiers": [
        {
            "type": "string",
            "value": "string"
        }
    ],
    "pointOfInterestName": "string",
    "geoCoordinates": {
        "latitude": 0,
        "longitude": 0
    },
    "pointOfInterestType": "string",
    "pointOfInterestDescription": "string",
    "pointOfInterestId": "string",
    "shouldDelete": false
}
]
},
"infoMessages": [
    {
        "severity": "ERROR",
        "message": "string",
        "diagnostic": "string",
        "source": "SYSTEM"
    }
]
}

```

Error HTTP Response Codes	
400	Invalid request
401	Invalid or missing credentials
403	User not authorized to perform this operation

### 2.2.3. Fields and Objects

The table below describes only the minimum relevant fields of the API Request to create a shipment for tracking. These fields are also returned with the API Response 201 Created.

**Table 2. Minimum Relevant Fields**

Field	Description
carrierIdentifier	The identifier of the carrier tendering the shipment. Type must be SCAC, DOT_NUMBER, MC_NUMBER, P44_EU, or SYSTEM.
shipmentIdentifiers	The shipment identifiers that are unique to this shipment and carrier identifier. These may be used in subsequent API calls along with the carrier identifier to reference a shipment. They may also be used by project44 to track the shipment. Only one shipment identifier of each type may be provided.

Field	Description
shipmentStops	<p><b>Field Type:</b> array</p> <p>This array includes all shipment stops for the shipment and must include the following information</p> <ul style="list-style-type: none"> <li>• <code>stopNumber</code>: The user-defined stop number, where '1' is for the origin and the destination has the largest number, with any stops in between ordered by appointment time.</li> <li>• <code>appointmentWindow</code>: The shipment's expected window of arrival at the defined stop. The start and end times (<code>appointmentWindow.startDateTime</code> and <code>appointmentWindow.endDateTime</code>) are required. The time zone identifier (e.g., 'America/Chicago') for this date time window is an optional field. If not provided the system will default to the timezone local to the stop. Please note, we will always report back the timezone for the appointment window as local to the stop.</li> <li>• One of the following: <ul style="list-style-type: none"> <li>• <code>location</code>: The user-defined location of this stop, including a contact and an optional address. Use if no geocoordinates are provided.</li> <li>• <code>geocoordinates</code>: The latitude and longitude of the stop, determined by geocoding the user-defined address. This field may not be provided when creating a shipment, but will always be returned. Use if no address is provided.</li> </ul> </li> </ul> <p>This list must contain the shipment's origin and destination and any stops of interest in between (i.e., other pickup or delivery locations).</p>
equipmentIdentifiers	<p><b>Field Type:</b> array</p> <p>The equipment identifiers identifying truckload equipment that can be tracked, such as cell phones and electronic logging devices (ELDs). These need not be unique to this shipment. Must be at least one of <code>MOBILE_PHONE_NUMBER</code>, <code>VEHICLE_ID</code>, <code>LICENSE_PLATE</code>, <code>SENSI-TECH_DEVICE_ID</code>, or <code>TIVE_DEVICE_ID</code>.</p>

## 2.3. Link Ocean and Truckload Shipments

### 2.3.1. Link a Collection of Shipments that are Transported Using Multiple Modes

This section describes how to link a shipment that is transported via Ocean vessel then deconsolidated and transported via Truckload.

To link a collection of shipments with multiple related components that all move independently across multiple modes,

1. [Create the Ocean shipment \[6\]](#). Store the MSID from the POST response.
2. [Create the Truckload shipments \[13\]](#). Store the MSIDs from the POST responses for each shipment.
3. Update the Ocean shipment using [the endpoint for creating the Ocean shipment \[6\]](#). Add each Truckload shipments' MSID and identifier in the `relatedShipments` array.

### 2.3.2. Link a Single Shipment that is Transported Using Multiple Modes

This section describes how to link a single shipment that is transported via Ocean vessel then via Truckload.

To link a single shipment that moves across multiple modes,

1. [Create the Ocean shipment \[6\]](#). Store the MSID from the POST response.
2. [Create the Truckload shipment \[13\]](#). Store the MSID from the POST response.
3. Update the Ocean shipment using the endpoint for creating the Ocean shipment. Add the Truckload shipment's MSID and identifier to the `routeSegments` array.



## 3. Retrieve Shipment Event History

### 3.1. Description

The **GET: Fetch the event history for a shipment** endpoint should be used to return information about shipment states and events (the milestones that occurred or will occur at a point in time for the shipment). The shipment's State Type is determined by specific Event Types. The accepted State Type values for Ocean shipments are listed below with the Event Types that trigger a State Type start or stop.

Event History updates occur when the shipment's data changes. For more information on PUSH updates, see [Section 6. PUSH Updates for Ocean Shipments](#).

### 3.2. Structure

The Event History endpoint retrieves the shipment's state (condition the shipment was in for a period of time) and event (milestone that occurred or will occur at a point in time for the shipment) history.

<b>Endpoint</b>	/api/v4/shipments/{shipmentId}/eventhistory
<b>Verb</b>	GET
<b>Required fields</b>	• Shipment ID
<b>Expected HTTP response code</b>	200 OK

## Response schema

```

{
  "sharingInformation": {
    "entitlements": [
      {
        "createdById": 0,
        "createdDate": "string",
        "entitledId": 0,
        "entitledRegion": "string",
        "entityId": "string",
        "entityTenantId": 0,
        "entityType": "SHIPMENT",
        "excludedData": [
          "SHIPMENT_STOPS"
        ],
        "lastModifiedById": 0,
        "lastModifiedDate": "string",
        "loadIds": [
          "string"
        ],
        "referenceId": "string",
        "ruleId": "string",
        "shipmentLegIds": [
          "string"
        ],
        "stopIds": [
          "string"
        ]
      }
    ]
  },
  "shipment": {
    "id": "string",
    "identifiers": [
      {
        "type": "string",
        "value": "string"
      }
    ],
    "routeInfo": {
      "stops": [
        {
          "id": "string",
          "type": "string",
          "location": {
            "id": "string",
            "name": "string",
            "identifiers": [
              {
                "type": "string",
                "value": "string"
              }
            ]
          },
          "address": {
            "postalCode": "string",
            "addressLines": [
              "string"
            ],
            "city": "string",
            "state": "string",
            "country": "string"
          },
          "coordinates": {
            "latitude": 0,
            "longitude": 0
          },
          "timeZone": "string"
        }
      ]
    }
  }
}

```

```

    ],
    "routeSegments": [
      {
        "id": "string",
        "fromStopId": "string",
        "toStopId": "string",
        "transportationMode": "string",
        "identifiers": [
          {
            "type": "string",
            "value": "string"
          }
        ]
      }
    ]
  },
  "attributes": [
    {
      "name": "string",
      "value": "string",
      "values": [
        "string"
      ]
    }
  ],
  "relatedShipments": [
    {
      "id": "string",
      "identifiers": [
        {
          "type": "string",
          "value": "string"
        }
      ],
      "attributes": [
        {
          "name": "string",
          "value": "string",
          "values": [
            "string"
          ]
        }
      ],
      "shipmentShareLink": "string",
      "createdDateTime": "string",
      "lastModifiedDateTime": "string"
    }
  ],
  "shipmentShareLink": "string",
  "createdDateTime": "string",
  "lastModifiedDateTime": "string"
},
"states": [
  {
    "type": "string",
    "startDateTime": "string",
    "endDateTime": "string",
    "stopId": "string",
    "routeSegmentId": "string"
  }
],
"events": [
  {
    "dateTime": "string",
    "receivedDateTime": "string",
    "type": "string",
    "description": "string",
    "stopId": "string",
    "routeSegmentId": "string",

```

	<pre>                 "estimateDateTime": "string",                 "estimateLastCalculatedDateTime": "string",                 "plannedDateTime": "string",                 "plannedEndDateTime": "string"             }         ]     } </pre>
<b>Error HTTP Response Codes</b>	<p>400 Invalid request</p> <p>401 Invalid or missing credentials</p> <p>403 User not authorized to perform this operation</p>

### 3.3. Field and Objects

The table below describes fields returned in the 200 OK response from the **GET: Fetch the event history for a shipment** endpoint.

**Table 3. Returned Fields for Event Updates**

Field	Description
relatedShipments	<p>This is only available for Ocean shipments.</p> <p>A BOL/BN can consist of multiple containers. An array of all containers tracked under the overall shipment (as created by the user) is returned in the <code>relatedShipments</code> object. Each object uses an unique ID to retrieve or identify (for PUSH) updates related to the designated container.</p>
routeInfo	<p>An overview of the shipment route includes all the necessary IDs for mapping each event to the correct stop, vessel, and segment of the journey.</p>
stops (under routeInfo)	<p>A stop is a set point of the journey that can be an inland location or a port.</p> <p>Several data fields are used to describe an array of stops as defined in the API:</p> <ul style="list-style-type: none"> <li><code>id</code>: project44 ID of the stop the event is related to (referenced to as <code>stopId</code> in events payload)</li> <li><code>type</code>: the type of the stop (see <a href="#">below [29]</a>)</li> <li><code>location</code>: object that consists of:                         <ul style="list-style-type: none"> <li><code>name</code>: name of the location (e.g. Port of Rotterdam) – received by shipping line (no input from customer required)</li> <li><code>identifiers</code>: type:value pair – ports populate with UN LOCODE.</li> </ul> </li> <li><code>coordinates</code>: geo-coordinates of the location.</li> </ul> <p>Port locode is supplied for all types of stops to include inland locations. These locations are designated with a UN LOCODE (e.g. rail terminals). Some inland locations without a UN LOCODE can return a blank <code>identifiers</code> array that has a name.</p> <p>For descriptions of specific stop types, see <a href="#">Shipment Stop Types by Mode [29]</a>.</p>
routeSegments (under routeInfo)	<p>Describes the journey between two stops. In the API, the <code>routeSegments</code> array contains the following information:</p> <ul style="list-style-type: none"> <li><code>id</code>: project44 ID of the route segment the event is related to (referenced to as <code>routeSegmentId</code> in events and positions payload).</li> <li><code>fromStopId</code>: project44 ID of where the route segment originates.</li> <li><code>toStopId</code>: project44 ID of where the route segment terminates.</li> <li><code>transportationMode</code>: mode of transport used for the route segment.</li> <li><code>identifiers</code>: array that consists of type:value pairs. See <a href="#">Route Segment Identifiers by Mode [30]</a> for valid IDs by mode.</li> </ul>

Field	Description
events	<p>An array of all events related to the shipment. Each event contains several data points:</p> <ul style="list-style-type: none"> <li>• <code>dateTime</code>: the time at which the event took place (actual).</li> <li>• <code>estimateDateTime</code>: project44's latest estimated time for the event.</li> <li>• <code>plannedDateTime</code>: the carrier's latest scheduled time for the event. <b>NOTE:</b> Timestamps related to an event do not currently include contracted time or initial planned time by carrier.</li> <li>• <code>type</code>: the type of event that occurs (full list available <a href="#">below [31]</a>).</li> <li>• <code>description</code>: a description of the event that occurs (not available for all events).</li> <li>• <code>stopId</code>: the project44 ID of the stop the event is related to.</li> <li>• <code>routeSegmentId</code>: the project44 ID of the route segment the stop is related to. See <a href="#">Route Segment Identifiers by Mode [30]</a> for more information on the specific identifier types.</li> <li>• <code>receivedDateTime</code>: last time info about event was modified.</li> <li>• <code>estimateLastCalculatedDateTime</code>: last time the estimate date time was modified.</li> </ul> <p>See <a href="#">Shipment Event Types by Mode [31]</a> for a full list of supported event types and descriptions.</p>
states	<p>An array of all states the shipment was or currently is in.</p> <ul style="list-style-type: none"> <li>• <code>type</code>: the specific type of state the shipment was in. <b>Valid values:</b> "UNKNOWN", "AT_STOP", "IN_TRANSIT", "COMPLETED", "SCHEDULED"</li> <li>• <code>startDateTime</code>: the time this state began.</li> <li>• <code>endDateTime</code>: the time that this state ended. May not be populated if the shipment is still in this state.</li> <li>• <code>stopId</code>: the ID of the related stop.</li> <li>• <code>routeSegmentId</code>: the route segment ID this state occurred along.</li> </ul> <p>See <a href="#">Shipment State Types by Mode [30]</a> for a full list of the supported state types and their descriptions.</p>
shipment-ShareLink	The URL to a live tracking page for the shipment that does not require user login. This URL can be used for internal purposes or for sharing with external third parties.

### 3.3.1. Shipment Stop Types by Mode

Mode	Stop Type	Details
Ocean	PORT_OF_LOADING	Where container are loaded onto a vessel. <b>NOTE:</b> The <code>ORIGIN</code> , <code>PICKUP</code> , <code>RETURN</code> , or <code>TRANSFER</code> stop types are often the same location as the <code>PORT_OF_LOADING</code> or <code>PORT_OF_DISCHARGE</code> . In those situations, the stop type will show either <code>PORT_OF_LOADING</code> or <code>PORT_OF_DISCHARGE</code> .
	ORIGIN	Where the empty container is filled by the shipper with the goods to be transported.
	TRANSSHIPMENT_PORT	Where the containers were discharged from an initial vessel to be loaded on another vessel. There could be multiple transshipment ports along the transit or there could be no transshipment ports - depending on the shipment.
	PORT_OF_DISHCARGE	Where the containers were discharged from the vessel. <b>NOTE:</b> The <code>ORIGIN</code> , <code>PICKUP</code> , <code>RETURN</code> , or <code>TRANSFER</code> stop types are often the same location as the <code>PORT_OF_LOADING</code> or <code>PORT_OF_DISCHARGE</code> . In those situations, the stop type will show either <code>PORT_OF_LOADING</code> or <code>PORT_OF_DISCHARGE</code> .
	DESTINATION	Where the container is delivered to the consignee.
	UNKNOWN	An undefined stop type.
	PICKUP	Where the empty container is picked up.
	RETURN	Where the empty container is returned.
	TRANSFER	Where the land transfer stop is.

Mode	Stop Type	Details
Truckload	ORIGIN	Where the truck is filled by the shipper with the goods to be transported. <b>NOTE:</b> Stops are merged if the stop ID or the address of the Truckload ORIGIN matches the stop ID or address of the the Ocean PORT_OF_DISCHARGE.
	PICKUP	Where the empty vehicle is loaded with the shipment.
	DELIVERY	Where the shipment is delivered.
	TRANSFER	Where the transfer stop is.
	DESTINATION	Where the shipment is finally delivered to.

### 3.3.2. Route Segment Identifiers by Mode

This table describes the route segment identifiers by mode.

Mode	Route Segment ID	Description
Ocean	VESSEL_NAME	The name of the vessel assigned to the route segment.
	VESSEL_IMO	The IMO number of the vessel assigned to the route segment.
	VOYAGE_NUMBER	The voyage number/ID of the voyage of the vessel assigned to the route segment.
Truckload	LICENSE_PLATE	The license plate of the vehicle assigned to the route segment.
	VEHICLE_ID	The vehicle ID of the vehicle assigned to the route segment.
	SENSITECH_DEVICE_ID	The ID of the Sensitech device used in the vehicle assigned to the route segment.
	TIVE_DEVICE_ID	The ID of the Tive device used in the vehicle assigned to the route segment.

### 3.3.3. Shipment State Types by Mode

Below are the accepted State Type values for Ocean and Truckload shipments and the Event Types that trigger a State Type start or stop.

Mode	State Type	Start Event	Stop Event
Ocean	SCHEDULED	TRACKING_STARTED	Any next event.
		DEPARTURE_FROM_STOP	ARRIVAL_AT_STOP
	IN_TRANSIT	GATE_OUT_EMPTY	GATE_IN_EMPTY
		GATE_OUT_FULL	GATE_IN_FULL
		ARRIVAL_AT_STOP	GATE_OUT_FULL
	AT_STOP	GATE_IN_FULL	DEPARTURE_FROM_STOP
		ARRIVAL_AT_STOP	GATE_OUT_EMPTY
COMPLETED	GATE_IN_EMPTY	N/A	
UNKNOWN	N/A	N/A	
Truckload	SCHEDULED	TRACKING_START	Any next event.
		DEPARTURE_FROM_STOP	ARRIVAL_AT_STOP
	IN_TRANSIT	LOAD	UNLOAD
		ARRIVAL_AT_STOP	DEPARTURE_FROM_STOP
		LOAD	UNLOAD
	AT_STOP	ARRIVAL_AT_STOP	DEPARTURE_FROM_STOP
		DELIVERY	N/A
COMPLETED	TRACKING_END_BY_USER		
	TRACKING_END_DUE_TO_TIMEOUT		

### 3.3.4. Shipment Event Types by Mode

Below are the available Event Types for Ocean and Truckload Shipments and the descriptions for each Event Type.

Mode	Event Type	Description	
Ocean	GATE_OUT_EMPTY	The container has left the origin port of loading or origin place of receipt to be filled by the manufacturer or shipper. This event can be used to indicate tracking for a shipment has ended.	
	GATE_IN_FULL	The container has been brought back to the origin port of loading or place of receipt to be loaded onto the ship.	
	LOAD	The shipment has been loaded onto the vessel. The container has been loaded onto the ship and is now officially in transit to the next port.	
	DEPARTURE_FROM_STOP	The vessel has departed from the port.	
	ARRIVAL_AT_STOP	The vessel has arrived at the port.	
	DISCHARGE	The shipment has been discharged from the vessel.	
	GATE_OUT_FULL	The container has left the port full at the destination to be unloaded.	
	DELIVERY	The cargo has been delivered to the shipper. This event can be used to indicate tracking for a shipment has ended.	
	GATE_IN_EMPTY	The container has returned empty to the port.	
	TRACKING_START	The shipment has been created and tracking has been initialized.	
	TRACKING_INITIATE	The shipment has been created, but tracking has not yet been initialized.	
	Truckload	UNLOAD_FROM_VEHICLE	The shipment is being unloaded from the vehicle at a stop.
		UNKNOWN_EQUIPMENT_IDENTIFIER	User submitted the wrong equipment identifier.
TRACKING_START		Shipment tracking has started.	
TRACKING_END_DUE_TO_TIMEOUT		Shipment tracking has ended because the appointment window for delivery has ended.	
ARRIVAL_AT_STOP		The shipment has arrived at a stop.	
TRACKING_FAILED		Tracking has failed.	
MISSING_EQUIPMENT_IDENTIFIER		An equipment identifier was not added during shipment creation.	
IDLE		The vehicle transporting the shipment is not moving.	
INFO		Shipment exception. See the <code>event.description</code> for more information on this exception.	
LOAD		The shipment is being loaded onto a vehicle.	
DEPARTURE_FROM_STOP		The vehicle transporting the shipment has left the stop.	
DRIVER_DENY_TRACKING		The driver has not allowed tracking.	
DEPARTURE_FROM_STOP		The vehicle transporting the shipment has left the stop.	
DELIVERY		The shipment is being delivered to its final destination.	
TRACKING_END_BY_USER		Tracking of the shipment has been ended by the shipper.	

### Determine the Estimated Time of Arrival and Actual Time of Arrival from Events

Estimated Time of Arrival (ETA) and Actual Time of Arrival (ATA) can be returned by the **GET: Fetch the event history for a shipment** endpoint. All Event Types can sometimes return an ETA or ATA. Please connect with your project44 Customer Success Manager to identify which event types return an ETA or ATA most often for your shipments.

Mode	Event Type	Will return ETA?	Will return ATA?
Ocean	GATE_OUT_EMPTY	Sometimes	Sometimes
	GATE_IN_FULL	Sometimes	Sometimes

Mode	Event Type	Will return ETA?	Will return ATA?
	LOAD	Sometimes	Sometimes
	DEPARTURE_FROM_STOP	Sometimes	Sometimes
	ARRIVAL_AT_STOP	Sometimes	Sometimes
	DISCHARGE	Sometimes	Sometimes
	GATE_OUT_FULL	Sometimes	Sometimes
	DELIVERY	Sometimes	Sometimes
	GATE_IN_EMPTY	Sometimes	Sometimes
	TRACKING_START	Sometimes	Sometimes
	TRACKING_INITIATE	Sometimes	Sometimes
	Truckload	UNLOAD_FROM_VEHICLE	Sometimes
UNKNOWN_EQUIPMENT_IDENTIFIER		Sometimes	Sometimes
TRACKING_START		Sometimes	Sometimes
TRACKING_END_DUE_TO_TIMEOUT		Sometimes	Sometimes
ARRIVAL_AT_STOP		Sometimes	Sometimes
TRACKING_FAILED		Sometimes	Sometimes
MISSING_EQUIPMENT_IDENTIFIER		Sometimes	Sometimes
IDLE		Sometimes	Sometimes
INFO		Sometimes	Sometimes
LOAD		Sometimes	Sometimes
DEPARTURE_FROM_STOP		Sometimes	Sometimes
DRIVER_DENY_TRACKING		Sometimes	Sometimes
DEPARTURE_FROM_STOP		Sometimes	Sometimes
DELIVERY		Sometimes	Sometimes
TRACKING_END_BY_USER		Sometimes	Sometimes



## 4. Retrieve Shipment Position History

### 4.1. Description

The **GET: Fetch the position history for a shipment endpoint** should be used to return information about a shipment's location coordinates at a specific point in time. Position updates are returned every 1-8 hours while a vessel is in transit.

**NOTE:** The frequency of returns also applies to PUSH. Additionally, the geolocation update is not a specific event code but a new update in the positions array.

### 4.2. Structure

<b>Endpoint</b>	/api/v4/shipments/{shipmentId}/positionhistory
<b>Verb</b>	GET
<b>Required Fields</b>	• Shipment ID
<b>Query parameters*</b>	• <code>includeRouteInfo</code> , Boolean (default = false) – if enabled, returns the <code>routeInfo</code> for the shipment (as explained in the Push section of this document).
<b>Expected HTTP response code</b>	200 OK

## Response schema

```

{
  "shipment": {
    "id": "string",
    "identifiers": [
      {
        "type": "BILL_OF_LADING",
        "value": "string"
      }
    ],
    "routeInfo": {
      "stops": [
        {
          "id": "string",
          "type": "UNKNOWN",
          "location": {
            "id": "string",
            "name": "string",
            "identifiers": [
              {
                "type": "AIRPORT_IATA",
                "value": "string"
              }
            ],
            "address": {
              "postalCode": "string",
              "addressLines": [
                "string"
              ],
              "city": "string",
              "state": "string",
              "country": "US"
            },
            "coordinates": {
              "latitude": 0,
              "longitude": 0
            },
            "timeZone": "string"
          }
        }
      ],
      "routeSegments": [
        {
          "id": "string",
          "fromStopId": "string",
          "toStopId": "string",
          "transportationMode": "OCEAN",
          "identifiers": [
            {
              "type": "BILL_OF_LADING",
              "value": "string"
            }
          ]
        }
      ]
    },
    "attributes": [
      {
        "name": "string",
        "value": "string",
        "values": [
          "string"
        ]
      }
    ],
    "relatedShipments": [
      {
        "id": "string",
        "identifiers": [

```

```

        {
          "type": "BILL_OF_LADING",
          "value": "string"
        }
      ],
      "attributes": [
        {
          "name": "string",
          "value": "string",
          "values": [
            "string"
          ]
        }
      ],
      "shipmentShareLink": "string",
      "createdDateTime": "string",
      "lastModifiedDateTime": "string"
    }
  ],
  "shipmentShareLink": "string",
  "createdDateTime": "string",
  "lastModifiedDateTime": "string"
},
"positions": [
  {
    "dateTime": "string",
    "receivedDateTime": "string",
    "routeSegmentId": "string",
    "latitude": 0,
    "longitude": 0
  }
]
]
}

```

Error HTTP Response Codes	
	400 Invalid request
	401 Invalid or missing credentials
	403 User not authorized to perform this operation

\*Query parameters are parameters that are appended to the URL.

### 4.3. Fields and Objects

The table below describes fields returned in the response 200 OK from the **GET: Fetch the position history for a shipment** endpoint.

**Table 4. Returned Fields for Position Updates**

Field	Description
related-Shipments	A BOL/BN can consist of multiple containers. An array of all containers tracked under the overall shipment (as initialized by the user) returns in the relatedShipments object. Each object uses an unique ID to retrieve or identify (for PUSH) updates related to the designated container.
routeInfo	An overview of the shipment route includes all the necessary IDs for mapping each event to the correct stop, vessel, and segment of the journey.

Field	Description
stops	<p>A stop is a set point of the journey that can be an inland location or a port. A full list of stop types is available <a href="#">here [36]</a>.</p> <p>A number of data fields describe an array of stops as defined in the API:</p> <ul style="list-style-type: none"> <li>• <code>id</code>: project44 ID of the stop associated with the shipment.</li> <li>• <code>type</code>: the type of the stop (see <a href="#">here [36]</a>).</li> <li>• <code>location</code>, object that consists of: <ul style="list-style-type: none"> <li>• <code>name</code>: name of the location (e.g., Port of Rotterdam) received by the shipping line (no input from the shipper is required).</li> <li>• <code>identifiers</code>: type:value pair - ports will always populate with UN_LOCODE.</li> </ul> </li> <li>• <code>coordinates</code>: geocoordinates of the location.</li> </ul> <p>Port locode is supplied for all types of stops to include inland locations. These locations are designated with a UN LOCODE (e.g. rail terminals). Some inland locations without a UN LOCODE can return a blank identifiers array that has a name.</p> <p><b>NOTE:</b> All stop information is retrieved by project44 and requires no input from the user.</p>
routeSegments	<p>Describes the journey between two stops. For Ocean freight, this is always associated with a vessel. It is important to understand this concept to map each event to the correct location and/or vessel. In the API, it is defined as an array of routeSegments that contain the following information:</p> <ul style="list-style-type: none"> <li>• <code>id</code>: project44 ID of the route segment the position is related to (referenced to as routeSegmentId in positions payload).</li> <li>• <code>fromStopId</code>: project44 ID of where the route segment originates.</li> <li>• <code>toStopId</code>: project44 ID of where the route segment terminates.</li> <li>• <code>transportationMode</code>: mode of transport used for the route segment.</li> <li>• <code>identifiers</code>, array that consists of type:value pairs, typically. See <a href="#">Route Segment Identifiers by Mode [37]</a> for a list of mode-specific route segment IDs with descriptions.</li> </ul>
positions	<p>An array of all geo-positions related to the shipment. Each event consists of a number of data points (not all data points are included for all events):</p> <ul style="list-style-type: none"> <li>• <code>dateTime</code>: the time the recorded position.</li> <li>• <code>receivedDateTime</code>: time the position was received by project44.</li> <li>• <code>routeSegmentId</code>: ID of the route segment the stop is related to.</li> </ul>
shipment-Share-Link	<p>The URL to a live tracking page for the shipment that does not require user login. This URL can be used for internal purposes or for sharing with external third parties.</p>

### 4.3.1. Shipment Stop Types by Mode

Mode	Stop Type	Details
Ocean	PORT_OF_LOADING	<p>Where container are loaded onto a vessel.</p> <p><b>NOTE:</b> The <code>ORIGIN</code>, <code>PICKUP</code>, <code>RETURN</code>, or <code>TRANSFER</code> stop types are often the same location as the <code>PORT_OF_LOADING</code> or <code>PORT_OF_DISCHARGE</code>. In those situations, the stop type will show either <code>PORT_OF_LOADING</code> or <code>PORT_OF_DISCHARGE</code>.</p>
	ORIGIN	<p>Where the empty container is filled by the shipper with the goods to be transported.</p>
	TRANSSHIPMENT_PORT	<p>Where the containers were discharged from an initial vessel to be loaded on another vessel. There could be multiple transshipment ports along the transit or there could be no transshipment ports - depending on the shipment.</p>
	PORT_OF_DISHCHARGE	<p>Where the containers were discharged from the vessel.</p> <p><b>NOTE:</b> The <code>ORIGIN</code>, <code>PICKUP</code>, <code>RETURN</code>, or <code>TRANSFER</code> stop types are often the same location as the <code>PORT_OF_LOADING</code> or <code>PORT_OF_DISCHARGE</code>. In those situations, the stop type will show either <code>PORT_OF_LOADING</code> or <code>PORT_OF_DISCHARGE</code>.</p>

Mode	Stop Type	Details
	DESTINATION	Where the container is delivered to the consignee.
	UNKNOWN	An undefined stop type.
	PICKUP	Where the empty container is picked up.
	RETURN	Where the empty container is returned.
	TRANSFER	Where the land transfer stop is.
Truckload	ORIGIN	Where the truck is filled by the shipper with the goods to be transported. <b>NOTE:</b> Stops are merged if the stop ID or the address of the Truckload ORIGIN matches the stop ID or address of the the Ocean PORT_OF_DISCHARGE.
	PICKUP	Where the empty vehicle is loaded with the shipment.
	DELIVERY	Where the shipment is delivered.
	TRANSFER	Where the transfer stop is.
	DESTINATION	Where the shipment is finally delivered to.

### 4.3.2. Route Segment Identifiers by Mode

This table describes the route segment identifiers by mode.

Mode	Route Segment ID	Description
Ocean	VESSEL_NAME	The name of the vessel assigned to the route segment.
	VESSEL_IMO	The IMO number of the vessel assigned to the route segment.
	VOYAGE_NUMBER	The voyage number/ID of the voyage of the vessel assigned to the route segment.
Truckload	LICENSE_PLATE	The license plate of the vehicle assigned to the route segment.
	VEHICLE_ID	The vehicle ID of the vehicle assigned to the route segment.
	SENSITECH_DEVICE_ID	The ID of the Sensitech device used in the vehicle assigned to the route segment.
	TIVE_DEVICE_ID	The ID of the Tive device used in the vehicle assigned to the route segment.

## 5. Retrieve Shipment Route Information

### 5.1. Description

The **GET: Fetch the route information for a shipment including the stops and route segments** endpoint should be used to return information about shipment stops and route segments.

### 5.2. Structure

<b>Endpoint</b>	/api/v4/shipments/{shipmentId}/tracking
<b>Verb</b>	GET
<b>Required Fields</b>	• Shipment ID
<b>Query parameters</b>	• <code>includeRouteInfo</code> , <code>&lt;Boolean&gt;</code> (default = false) - if enabled, will return the <code>routeInfo</code> for the shipment (as explained in the Push section of this document).
<b>Expected HTTP response code</b>	200 OK

## Response schema

```

{
  "id": "string",
  "identifiers": [
    {
      "type": "AIR_WAYBILL",
      "value": "string"
    }
  ],
  "routeInfo": {
    "stops": [
      {
        "id": "string",
        "type": "UNKNOWN",
        "location": {
          "id": "string",
          "name": "string",
          "identifiers": [
            {
              "type": "AIRPORT_IATA",
              "value": "string"
            }
          ],
          "address": {
            "postalCode": "string",
            "addressLines": [
              "string"
            ],
            "city": "string",
            "state": "string",
            "country": "US"
          },
          "coordinates": {
            "latitude": 0,
            "longitude": 0
          },
          "timeZone": "string"
        }
      }
    ],
    "routeSegments": [
      {
        "id": "string",
        "fromStopId": "string",
        "toStopId": "string",
        "transportationMode": "AIR",
        "identifiers": [
          {
            "type": "AIR_WAYBILL",
            "value": "string"
          }
        ]
      }
    ]
  },
  "attributes": [
    {
      "name": "string",
      "value": "string",
      "values": [
        "string"
      ]
    }
  ],
  "relatedShipments": [
    {
      "id": "string",
      "identifiers": [
        {

```

```

        "type": "AIR_WAYBILL",
        "value": "string"
    }
  ],
  "attributes": [
    {
      "name": "string",
      "value": "string",
      "values": [
        "string"
      ]
    }
  ],
  "shipmentShareLink": "string",
  "createdDateTime": "string",
  "lastModifiedDateTime": "string"
}
],
"shipmentShareLink": "string",
"createdDateTime": "string",
"lastModifiedDateTime": "string"
}
}

```

Error HTTP Response Codes	
400	Invalid request
401	Invalid or missing credentials
403	User not authorized to perform this operation

### 5.3. Fields and Objects

The table below describes fields returned in the response 200 OK from the **GET: Fetch the route information for a shipment including the stops and route segments** endpoint.

**Table 5. Returned Fields for Position Updates**

Field	Description
relatedShipments	A BOL/BN can consist of multiple containers. An array of all containers tracked under the overall shipment (as initialized by the user) returns in the relatedShipments object. Each object will use a unique ID to retrieve or identify (for PUSH) updates related to the designated container.
routeInfo	An overview of the shipment route includes all the necessary IDs for mapping each event to the correct stop, vessel, and segment of the journey.
stops (under routeInfo)	<p>A stop is a set point of the journey that can be an inland location or a port. A full list of stop types is available <a href="#">here [41]</a>.</p> <p>A number of data fields describe an array of stops as defined in the API:</p> <ul style="list-style-type: none"> <li>• <code>id</code>: project44 ID of the stop associated with the shipment.</li> <li>• <code>type</code>: the type of the stop (see <a href="#">here [42]</a>).</li> <li>• <code>location</code>, object that consists of:               <ul style="list-style-type: none"> <li>• <code>name</code>: name of the location (e.g., Port of Rotterdam) received by the shipping line (no input from the shipper required).</li> <li>• <code>identifiers</code>: type:value pair - ports will always populate with UN LOCODE.</li> <li>• <code>coordinates</code>: geocoordinates of the location.</li> </ul> </li> </ul> <p>Port locode is supplied for all types of stops to include inland locations. These locations are designated with a UN LOCODE (e.g. rail terminals). Some inland locations without a UN LOCODE can return a blank identifiers array that has a name.</p> <p><b>NOTE:</b> All stop information is retrieved by project44 and requires no input from the user.</p>



Field	Description
routeSegments (under routeInfo)	<p>Describes the journey between two stops. For Ocean freight, this is always associated with a vessel. It is important to understand this concept to map each event to the correct location and/or vessel. In the API, it is defined as an array of routeSegments that contain the following information:</p> <ul style="list-style-type: none"> <li>• id: project44 ID of the route segment the position is related to (referenced to as routeSegmentId in positions payload).</li> <li>• fromStopId: project44 ID of where the route segment originates.</li> <li>• toStopId: project44 ID of where the route segment terminates.</li> <li>• transportationMode: mode of transport used for the route segment.</li> <li>• identifiers, array that consists of type:value pairs, typically:</li> </ul> <p>See <a href="#">Route Segment Identifiers [42]</a> by mode for more information on the route segment IDs for each mode.</p>
positions	<p>An array of all geo-positions related to the shipment. Each event consists of a number of data points (not all data points are included for all events):</p> <ul style="list-style-type: none"> <li>• dateTime: the time the recorded position.</li> <li>• receivedDateTime: time the position was received by project44.</li> <li>• routeSegmentId: ID of the route segment the stop is related to.</li> </ul>
shipment-ShareLink	The URL to a live tracking page for the shipment that does not require user login. This URL can be used for internal purposes or for sharing with external third parties.

### 5.3.1. Shipment Stop Types by Mode

Mode	Stop Type	Details
Ocean	PORT_OF_LOADING	Where container are loaded onto a vessel. <b>NOTE:</b> The ORIGIN, PICKUP, RETURN, or TRANSFER stop types are often the same location as the PORT_OF_LOADING or PORT_OF_DISCHARGE. In those situations, the stop type will show either PORT_OF_LOADING or PORT_OF_DISCHARGE.
	ORIGIN	Where the empty container is filled by the shipper with the goods to be transported.
	TRANSSHIPMENT_PORT	Where the containers were discharged from an initial vessel to be loaded on another vessel. There could be multiple transshipment ports along the transit or there could be no transshipment ports - depending on the shipment.
	PORT_OF_DISHCARGE	Where the containers were discharged from the vessel. <b>NOTE:</b> The ORIGIN, PICKUP, RETURN, or TRANSFER stop types are often the same location as the PORT_OF_LOADING or PORT_OF_DISCHARGE. In those situations, the stop type will show either PORT_OF_LOADING or PORT_OF_DISCHARGE.
	DESTINATION	Where the container is delivered to the consignee.
	UNKNOWN	An undefined stop type.
	PICKUP	Where the empty container is picked up.
	RETURN	Where the empty container is returned.
Truckload	TRANSFER	Where the land transfer stop is.
	ORIGIN	Where the truck is filled by the shipper with the goods to be transported. <b>NOTE:</b> Stops are merged if the stop ID or the address of the Truckload ORIGIN matches the stop ID or address of the the Ocean PORT_OF_DISCHARGE.
	PICKUP	Where the empty vehicle is loaded with the shipment.
	DELIVERY	Where the shipment is delivered.
	TRANSFER	Where the transfer stop is.
DESTINATION	Where the shipment is finally delivered to.	

### 5.3.2. Route Segment Identifiers by Mode

This table describes the route segment identifiers by mode.

Mode	Route Segment ID	Description
Ocean	VESSEL_NAME	The name of the vessel assigned to the route segment.
	VESSEL_IMO	The IMO number of the vessel assigned to the route segment.
	VOYAGE_NUMBER	The voyage number/ID of the voyage of the vessel assigned to the route segment.
Truckload	LICENSE_PLATE	The license plate of the vehicle assigned to the route segment.
	VEHICLE_ID	The vehicle ID of the vehicle assigned to the route segment.
	SENSITECH_DEVICE_ID	The ID of the Sensitech device used in the vehicle assigned to the route segment.
	TIVE_DEVICE_ID	The ID of the Tive device used in the vehicle assigned to the route segment.

## 6. Delete a Shipment

### 6.1. Description

The **DELETE: Delete shipment route from project44 to stop tracking** endpoint should be used to stop a shipment from tracking in project44 and delete record in the UI.

### 6.2. Structure

<b>Endpoint</b>	/api/v4/shipments/{shipmentId}/tracking
<b>Verb</b>	DELETE
<b>Required Fields</b>	• Shipment ID
<b>Expected HTTP response code</b>	204 No content
<b>Error HTTP Response Codes</b>	400 Invalid request 401 Invalid or missing credentials 403 User not authorized to perform this operation

## 7. PUSH Updates for Shipments

### 7.1. Description

Users can PUSH Event History and Position History data updates to a webhook as defined by the customer. This sent data contains:

- positions
- events
- route information
- all related shipments

PUSH updates are sent on all shipments to include those created by the user (e.g., using the BOL or BN) and those discovered by project44 (e.g., containers for Ocean shipments).

BOLs and BNs send very few updates through their lifecycles. These updates only relate to tracking initialization and tracking completion. In the payload for each tracking update, the relatedShipments array includes an overview of all discovered containers and their associated identifiers.

Containers send updates with each new GPS position or milestone event. The container is identified in the payload with its internal project44 shipment ID. This ID is specified in the relatedShipments array of the BOL/booking shipment. Conversely, when receiving push updates, the relatedShipments array contains the BOL related to the designated container.

Each new event message includes all events related to the shipment. For position updates, only the latest is included in each new message.

For samples PUSH Updates, see [Sample PUSH Updates \[51\]](#).

## 8. Manage Webhooks

### 8.1. Description

Webhooks can be created and maintained via API. This API is used to receive PUSH tracking updates. A webhook must be set prior to receiving updates on shipments.

All tracking updates are sent to one webhook that is specified during implementation as master data in project44's backend. Sending data to multiple webhooks at the same time is not supported.

### 8.2. Structure

#### 8.2.1. Create or Update a PUSH Configuration

Use this to set up a webhook initially or to update an existing webhook configuration.

**NOTE:** This is a one-time activity.

<b>Endpoint</b>	/api/v4/webhooks
<b>Verb</b>	PUT
<b>Request schema</b>	<pre>{   "authParams": {},   "authType": "API_KEY",   "authenticationMethods": [     {       "parameters": {},       "type": "API_KEY"     }   ],   "configName": "string",   "id": 0,   "method": "GET",   "methodParams": {},   "payloadFormat": "JSON",   "url": "string",   "version": 0 }</pre>
<b>Expected HTTP response code</b>	200 OK

#### 8.2.2. Retrieve a Webhook Configuration by Name

<b>Endpoint</b>	/api/v4/webhooks/name/{webhookName}
<b>Verb</b>	GET
<b>Path parameters</b>	• webhookName, <string> (required)
<b>Expected HTTP response code</b>	200 OK

<b>Response schema</b>	<pre> {   "authParams": {},   "authType": "API_KEY",   "authenticationMethods": [     {       "parameters": {},       "type": "API_KEY"     }   ],   "configName": "string",   "id": 0,   "method": "GET",   "methodParams": {},   "payloadFormat": "JSON",   "url": "string",   "version": 0 } </pre>
------------------------	--

### 8.2.3. Retrieve All Webhook Configurations

<b>Endpoint</b>	/api/v4/webhooks
<b>Verb</b>	GET
<b>Expected HTTP Response Code</b>	200 OK
<b>Response schema</b>	<pre> {   "webhookEndpoints": [     {       "authParams": {},       "authType": "API_KEY",       "authenticationMethods": [         {           "parameters": {},           "type": "API_KEY"         }       ],       "configName": "string",       "id": 0,       "method": "GET",       "methodParams": {},       "payloadFormat": "JSON",       "url": "string",       "version": 0     }   ] } </pre>

## 8.3. Fields and Objects

**Table 6. Request/OK Response Fields and Descriptions**

Field	Description
authenticationMethods	<p>A list of authentication methods that consists of 'type' and 'parameters' fields. Typically, only one form of authentication is used (though it is possible to have two API keys or both an API key and OAuth2 authentication). Supported types are BASIC, API_KEY, OAUTH2.</p> <p>The fields of a 'parameters' object vary for each authentication method. All fields are strings unless noted otherwise.</p> <ul style="list-style-type: none"> <li>• API Key authentication - A new HTTP header is created for each supplied api key (max 2). <ul style="list-style-type: none"> <li>• <code>api_key_name</code>: the name of the API Key.</li> <li>• <code>api_key_value</code>: the value of the API Key.</li> </ul> </li> <li>• Basic authentication - Applied to the 'Authorization' header with the 'Basic' identifier. Values are concatenated (with a separating colon character) and encoded to base64. <ul style="list-style-type: none"> <li>• <code>username</code>: username for authentication.</li> <li>• <code>password</code>: password for authentication.</li> </ul> </li> <li>• OAuth2 authentication - The OAuth2 bearer token is generated then applied to the 'Authorization' header with the 'Bearer' identifier. <ul style="list-style-type: none"> <li>• <code>grantType</code>: Type of authentication granted to the OAuth2 provider. Currently only 'client_credentials' is supported.</li> <li>• <code>accessTokenUri</code>: endpoint used to grant the bearer token.</li> <li>• <code>clientId</code>: unique client identifier.</li> <li>• <code>clientSecret</code>: authentication key for the client; functionally similar to a password.</li> <li>• <code>clientAuthenticationScheme</code>: identifies whether the OAuth2 fields are supplied to the authorization grant in the header or the request body. Accepted values are 'header' and 'form'.</li> <li>• <code>extraParams</code>: Optional - An object containing additional fields to be added to the request to the access token provider.</li> </ul> </li> </ul>
configName	A unique name used to identify the push configuration.
id	A unique identifier of the push configuration. Assigned by the push configuration API service and used for updates only.
method	The HTTP method associated with the webhook endpoint. If not provided, will default to POST. Supported values: "GET", "HEAD", "POST", "PUT", "PATCH", "DELETE", "OPTIONS", "TRACE"
methodParams	A map of optional method parameters. If provided, these will appear as query parameters on the URL.
payload-Format	The format of the payload to send to the webhook, either JSON or XML. If not provided, will default to JSON.
url	The URL of the webhook endpoint.
version	An optional version of the push configuration.

## 9. Appendix

### 9.1. Error Response Codes

This Appendix describes the schema and fields of the standard project44 API Error Response.

The table below shows the API Error Response Schema:

**Table 7. API Error 400, 401, & 403 Response Schema**

<b>Error HTTP Response codes</b>	<p>400 Invalid request</p> <p>401 Invalid or missing credentials</p> <p>403 User not authorized to perform this operation</p>
<b>Error Response schema</b>	<pre>{   "statusCode": 0,   "httpMessage": "string",   "errorMessage": "string",   "errors": [     {       "severity": "ERROR",       "message": "string",       "diagnostic": "string",       "source": "SYSTEM"     }   ],   "supportReferenceId": "string" }</pre>

The table below describes each field of the API Error Response:

**Table 8. Error Response Schema Fields & Descriptions**

Error Response Field	Description
statusCode	<p><b>Field Type:</b> integer&lt;32&gt;</p> <p>This field contains the value of the HTTP status code.</p> <p><b>Possible Values:</b></p> <ul style="list-style-type: none"> <li>• 400</li> <li>• 401</li> <li>• 403</li> </ul>
httpMessage	<p><b>Field Type:</b> string</p> <p>This field contains a brief description of the HTTP status code.</p> <p><b>Possible Values:</b></p> <ul style="list-style-type: none"> <li>• Invalid request (statusCode = 400)</li> <li>• Invalid or missing credentials (statusCode = 401)</li> <li>• User not authorized to perform this operation (statusCode = 403)</li> </ul>
errorMessage	<p><b>Field Type:</b> string</p> <p>This field contains a more detailed description of the error.</p> <p><b>Example:</b></p> <p>"errorMessage": "Text '2022-03-23T18:38:00Z' could not be parsed, unparsed text found at index 19"</p>



Error Response Field	Description
errors	This array contains more details about the error
errors.severity	<p><b>Field Type:</b> string</p> <p>The field contains the severity of this message.</p> <p><b>Possible Values:</b></p> <ul style="list-style-type: none"> <li>• ERROR</li> <li>• WARNING</li> <li>• INFO</li> </ul>
error.message	<p><b>Field Type:</b> string</p> <p>This field contains more information about the error.</p> <p><b>Example:</b></p> <p>Appointment in the Past</p>
error.diagnostic	<p><b>Field Type:</b> string</p> <p>This field contains diagnostic information from project44 (The System) or from the Capacity Provider/Carrier.</p> <p><b>NOTE:</b> In a Capacity Provider/Carrier integration, it would come only from project44.</p>
error.source	<p><b>Field Type:</b> string</p> <p>This field contains the originator of the error message.</p> <p><b>NOTE:</b> In a Capacity Provider/Carrier integration, the originator will always be SYSTEM.</p> <p><b>Possible Values:</b></p> <ul style="list-style-type: none"> <li>• SYSTEM: project44.</li> <li>• CAPACITY PROVIDER: Capacity Provider or Carrier.</li> </ul>
supportReferenceId	<p><b>Field Type:</b> string</p> <p>This field contains a reference identifier used by project44 support to assist with certain error messages.</p>

## 9.2. Track Specific Containers Sample Payload

View sample payload for creating an Ocean shipment for tracking while tracking specific containers.

```

{
  "identifiers": [
    {
      "type": "BILL_OF_LADING",
      "value": "string"
    }
  ],
  "routeInfo": {
    "stops": [
      {
        "id": "string",
        "type": "UNKNOWN",
        "location": {
          "id": "string",
          "name": "string",
          "identifiers": [
            {
              "type": "AIRPORT_IATA",
              "value": "string"
            }
          ],
          "address": {
            "postalCode": "string",
            "addressLines": [
              "string"
            ],
            "city": "string",
            "state": "string",
            "country": "US"
          },
          "coordinates": {
            "latitude": 0,
            "longitude": 0
          },
          "timeZone": "string"
        }
      ]
    ],
    "routeSegments": [
      {
        "id": "string",
        "fromStopId": "string",
        "toStopId": "string",
        "transportationMode": "AIR",
        "identifiers": [
          {
            "type": "AIR_WAYBILL",
            "value": "string"
          }
        ]
      }
    ]
  },
  "attributes": [
    {
      "name": "string",
      "value": "string",
      "values": [
        "string"
      ]
    }
  ],
  "relatedShipments": [
    {
      "id": "string",
      "identifiers": [
        {
          "type": "AIR_WAYBILL",

```

```

        "value": "string"
      }
    ],
    "attributes": [
      {
        "name": "string",
        "value": "string",
        "values": [
          "string"
        ]
      }
    ],
    "shipmentShareLink": "string",
    "createdDateTime": "string",
    "lastModifiedDateTime": "string"
  }
],
"events": [
  {
    "dateTime": "string",
    "receivedDateTime": "string",
    "type": "UNKNOWN",
    "description": "string",
    "stopId": "string",
    "routeSegmentId": "string",
    "estimateDateTime": "string",
    "estimateLastCalculatedDateTime": "string",
    "plannedDateTime": "string",
    "plannedEndDateTime": "string"
  }
],
"shipmentShareLink": "string",
"createdDateTime": "string",
"lastModifiedDateTime": "string"
}

```

### 9.3. Sample PUSH Update

This section includes examples of sample PUSH Event History Updates.

**NOTE:** Each event is reported when it happens. It is not guaranteed that the event history is sent in every update.

See examples below (BOL identifiers in the left column and container identifiers in the right column).

```

{
  "shipment": {
    "id": "sample-p44-id-11111",
    "identifiers": [
      {
        "type": "BILL_OF_LADING",
        "value": "ABCD11223344"
      },
      {
        "type": "CARRIER_SCAC",
        "value": "MAEU"
      }
    ],
    "attributes": [],
    "shipmentShareLink": "string",
    "relatedShipments": [
      {
        "id": "sample-p44-id-22222",
        "identifiers": [
          {
            "type": "CONTAINER_ID",
            "value": "EFGH99887766"
          }
        ]
      }
    ]
  },
  "routeInfo": {
    "stops": [
      {
        "id": "string",
        "type": "PORT_OF_LOADING",
        "location": {
          "name": "string",
          "identifiers": [
            {
              "type": "PORT_UN_LOCODE",
              "value": "string"
            }
          ]
        },
        "coordinates": {
          "latitude": 0,
          "longitude": 0
        },
        "timeZone": "string"
      }
    ]
  },
  "routeSegments": [
    {
      "id": "string",
      "fromStopId": "string",
      "toStopId": "string",
      "transportationMode": "OCEAN",
      "identifiers": [
        {
          "type": "VOYAGE_NUMBER",
          "value": "string"
        },
        {
          "type": "VESSEL_NAME",
          "value": "string"
        },
        {
          "type": "VESSEL_IMO",
          "value": "string"
        }
      ]
    }
  ]
}

{
  "shipment": {
    "id": "sample-p44-id-22222",
    "identifiers": [
      {
        "type": "CONTAINER_ID",
        "value": "EFGH99887766"
      }
    ],
    "attributes": [],
    "shipmentShareLink": "string",
    "relatedShipments": [
      {
        "id": "sample-p44-id-11111",
        "identifiers": [
          {
            "type": "BILL_OF_LADING",
            "value": "ABCD11223344"
          },
          {
            "type": "CARRIER_SCAC",
            "value": "MAEU"
          }
        ]
      }
    ]
  },
  "routeInfo": {
    "stops": [
      {
        "id": "string",
        "type": "PORT_OF_LOADING",
        "location": {
          "name": "string",
          "identifiers": [
            {
              "type": "PORT_UN_LOCODE",
              "value": "string"
            }
          ]
        },
        "coordinates": {
          "latitude": 0,
          "longitude": 0
        }
      }
    ]
  },
  "routeSegments": [
    {
      "id": "string",
      "fromStopId": "string",
      "toStopId": "string",
      "transportationMode": "OCEAN",
      "identifiers": [
        {
          "type": "VOYAGE_NUMBER",
          "value": "string"
        },
        {
          "type": "VESSEL_NAME",
          "value": "string"
        },
        {
          "type": "VESSEL_IMO",
          "value": "string"
        }
      ]
    }
  ]
}

```

```
    ],
  },
},
"states": [],
"events": [
  {
    "dateTime": "string",
    "type": "TRACKING_INITIATE",
    "description": "Shipment tracking now",
  }
]
}

    },
"states": [],
"events": [
  {
    "dateTime": "string",
    "receivedDateTime": "string",
    "type": "DISCHARGED",
    "description": "string",
    "stopId": "string",
    "routeSegmentId": "string",
    "estimateDateTime": "string",
    "plannedDateTime": "string"
  }
]
}
```