

SF-LWR Extension Framework : UPS Shipping Integration

[Functional Overview](#)

[Technical Details](#)

[Sequence Diagram](#)

[Configuration details](#)

[Record - UPS Shipping \(Shipping Provider \)](#)

[Record - UPS Shipping \(HTTP Service\)](#)

[Source Code](#)

Functional Overview

This document provides details specific to integrating with UPS.

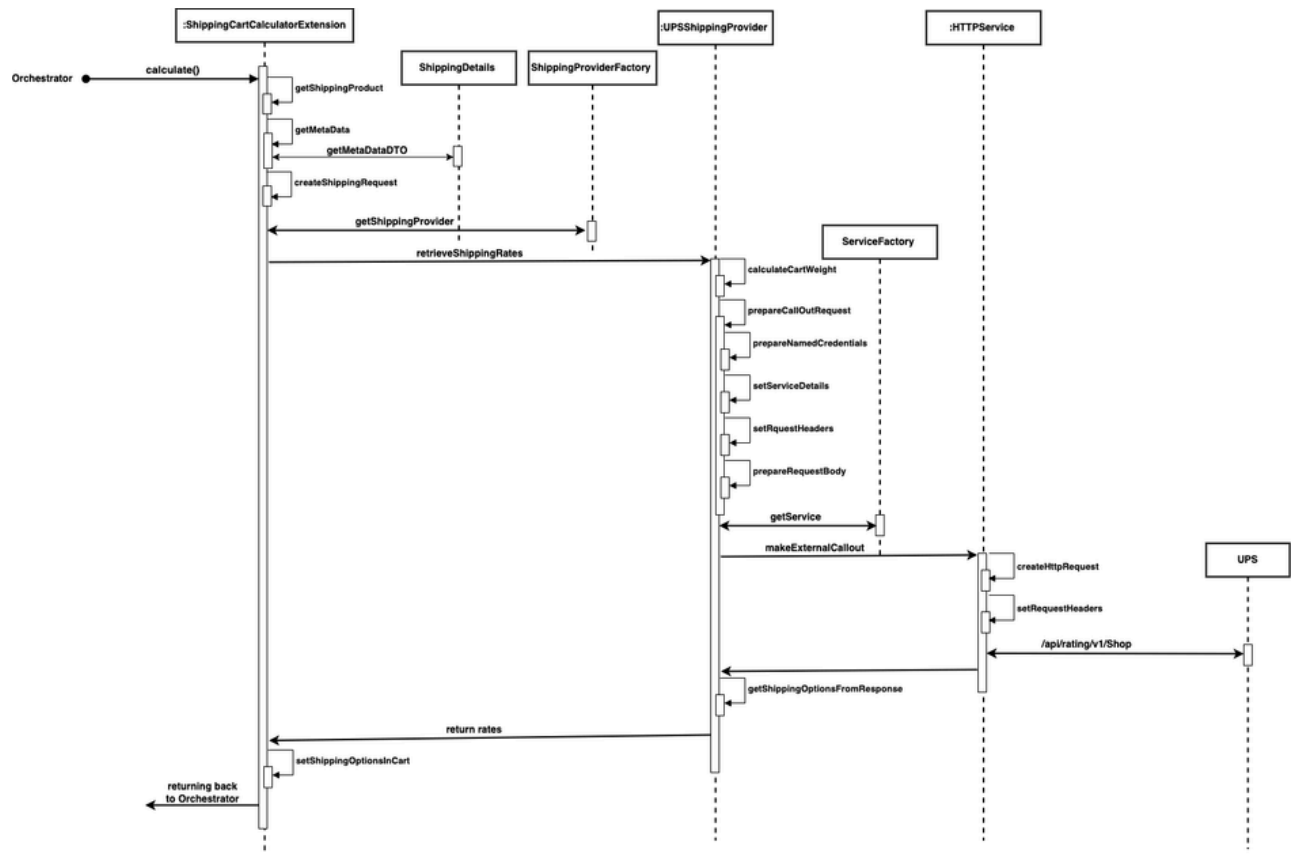
For common details you can refer to parent doc [here](#).

Technical Details

This integration implementation has two parts :

- **Configuration**
 - Custom Meta Data Records
 - UPS Shipping Provider
 - UPS Shipping Service
 - Named Credentials
 - UPS Shipping
 - External Credentials
 - UPS Authentication
 - UPSCredentials (Principal)
- **Source Code**
 - UPSShippingProvider
 - UPSShippingRateRequest
 - UPSShippingRateResponse

Sequence Diagram



Configuration details

	Section	Task	Details
1	Salesforce Org	Shipping Provider - Create a record for UPS	Create a record UPS Shipping for Custom Meta Data Type Shipping Provider
2	Salesforce Org	HTTP Service - Create a record for UPS	Create a record - UPS Shipping of above Custom Meta Data Type <div> <div>Shipping Provider Edit</div> <div> <div> <div>Label</div> <div>UPS Shipping Provider</div> </div> <div> <div>Shipping Provider Name</div> <div>UPS</div> </div> </div> <div> <div>Protected Component</div> <div><input type="checkbox"/></div> </div> <div> <div>Is Active?</div> <div><input checked="" type="checkbox"/></div> </div> <div> <div>Center</div> <div>UPS</div> </div> <div> <div>Display Order</div> <div>2</div> </div> <div> <div>Contract Name</div> <div>R1Training</div> </div> <div> <div>Contract Location</div> <div>en_US</div> </div> <div> <div>Impl Class</div> <div>UPSShippingProvider</div> </div> <div> <div>Max Weight Threshold</div> <div>150</div> </div> <div> <div>Min Weight Threshold</div> <div>5</div> </div> <div> <div>Shipping Options</div> <div> <div>"UPS Standard"</div> <div>"UPS 3 Day Select"</div> <div>"UPS Next Day Air Saver"</div> <div>"UPS Next Day Air Early"</div> </div> </div> <div> <div>Account Number</div> <div>HJ8214</div> </div> <div> <div>Request</div> <div> <div>"RateRequest": {</div> <div>"Request": {</div> <div>"TransactionReference": {</div> <div>"CustomerContext": {</div> <div>"Shipment": {</div> <div>"Shipper": {</div> <div>"Name": "ShippingName"</div> <div>}</div> <div>}</div> <div>}</div> <div>}</div> <div>}</div> <div>}</div> </div> </div> <div> <div>Service</div> <div>UPS Shipping Service</div> </div> <div> <div>Namespace Profile</div> <div></div> </div> </div>

3	Salesforce Org	Create External Credentials	Create External Credentials for OAuth 2.0,
---	----------------	-----------------------------	---

SETUP		
Named Credentials		
<div> Named Credentials External Credentials </div>		
1 Items - Sorted by Label		
Label	Authentication Protocol	Actions
UPS Authentication	OAuth 2.0	

Provide details , like

Name - UPSAuth

Authentication Protocol - AAuth 2.0

Authentication Flow Type - Client Credentials with Secret Flow

Identity Provider URL - <https://wwwcie.ups.com/security/v1/oauth/token>

Edit UPS Authentication

* Label

UPS Authentication

* Name

UPSAuth

* Authentication Protocol

OAuth 2.0

* Authentication Flow Type

Client Credentials with Client Secret Flow

Scope

* Identity Provider URL

<https://wwwcie.ups.com/security/v1/oauth/token>

☐

Pass client credentials in request body

Cancel

Save

SETUP > NAMED CREDENTIALS												
UPS Authentication												
<div> <div> <div>Label</div> <div>UPS Authentication</div> </div> <div> <div>Name</div> <div>UPSAuth</div> </div> </div> <div> <div> <div>Authentication Protocol</div> <div>OAuth 2.0</div> </div> <div> <div>Authentication Flow Type</div> <div>Client Credentials with Client Secret Flow</div> </div> <div> <div>Identity Provider URL</div> <div>https://wwwcie.ups.com/security/v1/oauth/token</div> </div> <div> <input type="checkbox"/> <div>Pass client credentials in request body</div> </div> </div> <div> <div>Managed Package Access</div> <div>Created By Namespace</div> </div>												
<div>Related Named Credentials</div> <table> <tr> <th>Label</th><th>Name</th><th>URL</th></tr> <tr> <td>UPS Shipping</td><td>UPSShipping</td><td>https://wwwcie.ups.com</td></tr> </table>			Label	Name	URL	UPS Shipping	UPSShipping	https://wwwcie.ups.com				
Label	Name	URL										
UPS Shipping	UPSShipping	https://wwwcie.ups.com										
<div>Principals</div> <table> <tr> <th>Se...</th><th>Parameter Name</th><th>Client ID</th><th>Authentication Status</th><th>Actions</th></tr> <tr> <td>1</td><td>UPSCredentials</td><td>e6LnmA4GRLS5sqcYUmfYF38RUd8yA2vckA/GacaSYQjpe6</td><td>Configured</td><td></td></tr> </table>			Se...	Parameter Name	Client ID	Authentication Status	Actions	1	UPSCredentials	e6LnmA4GRLS5sqcYUmfYF38RUd8yA2vckA/GacaSYQjpe6	Configured	
Se...	Parameter Name	Client ID	Authentication Status	Actions								
1	UPSCredentials	e6LnmA4GRLS5sqcYUmfYF38RUd8yA2vckA/GacaSYQjpe6	Configured									

Also create a Principal which will store Client ID & Client Secret

Name - UPSCredentials

Edit Principal

Parameter Name

UPSCredentials

Sequence Number

1

Client ID

e6LNmA4GRLS5ccqYUtnfyF3tlRUdByAGxlcArGace5YGj

Client Secret

.....

Principal Access

Name	Entity Type	ID
B2BBuyer	PermissionSet	OPSHn000001yYV7OAM

Cancel

Save

4

Salesforce Org

Create Named Credentials

Create a Named Credentials for storing UPS API URL

Set URL - <https://wwwcie.ups.com><https://onlinetools.ups.com>

Also to associate with External Credentials (created in previous step) select "UPS Credentials" under External Credentials

Under **Callout Options**

Turn on - Generate Authorization Header

Edit UPS Shipping

Label

UPS Shipping

Name

UPSShipping

URL

<https://wwwcie.ups.com>

Enabled for Callouts

☒

Authentication

External Credential

UPS Authentication

Client Certificate

Search Certificates...

Callout Options

Generate Authorization Header

☒

Allow Formulas in HTTP Header

☐

Allow Formulas in HTTP Body

☐

Outbound Network Connection

Select an Option

Cancel

Save

SETUP

Named Credentials

Named Credentials

External Credentials

1 Items - Sorted by Label

New

Label	Type	URL	External Credential	Actions
UPS Shipping	Secured Endpoint	https://wwwcie.ups.com	UPS Authentication	

Here are the details

SETUP > NAMED CREDENTIALS

UPS Shipping

EditDelete

Label

UPS Shipping

Name

UPSShipping

URL

https://www.ups.com

Enabled for Callouts

Authentication

External Credential

UPS Authentication

Client Certificate

Callout Options

Generate Authorization Header

Allow Formulas in HTTP Header

Allow Formulas in HTTP Body

Outbound Network Connection

Managed Package Access

Created By Namespace

Allowed Namespaces for Callouts

5

Salesforce Org

Assign Permission

To make callouts that use a named credential, you must enable principal access to permission sets or profiles for the users who need access.

Setup > Permission Sets > B2BBuyer

Click **External Credential Principal Access**

B2BBuyer

Find Settings...CloneEdit PropertiesManage AssignmentsView Summary (Beta)

Permission Set Overview

Description

Allows access to the store. Lets users see products and categories, make purchases, and add products to a cart.

API Name

B2BBuyer

License

B2B Buyer Permission Set One Seat

Namespace Prefix

Session Authentication Required

Created By

Mass_Tyagi

2/9/2024, 11:25 AM

Permission Set Groups Added To

0

Last Modified By

Mass_Tyagi

2/9/2024, 7:08 PM

Apps

Object Settings

Permissions to access objects and fields, and settings such as tab availability

App Permissions

Permissions to perform app-specific actions, such as "Manage Call Centers"

Flow Access

Permissions to execute flows

External Credential Principal Access

Permissions to authenticate with external credential principal mappings

Custom Metadata Types

Permissions to access custom metadata types

Custom Setting Definitions

Permissions to access custom settings

Settings that apply to Salesforce apps, such as Sales, and custom apps built on the Lightning Platform

Learn More

Now click edit and select principal created for UPS in above steps(step # 3)

SETUP

Permission Sets

View Tutorial / Help for this Page

Permission Set

B2BBuyer

Find Settings...CloneEdit PropertiesManage AssignmentsView Summary (Beta)

Permission Set Overview

External Credential Principal Access

External Credential Principal Access

Available External Credential Principals

Enabled External Credential Principals

FedExAuth - FedExCredentials

UPSAuth - UPSCredentials

--None--

AddRemove

Permission Set

B2BBuyer

Find Settings...CloneEdit PropertiesManage AssignmentsView Summary (Beta)

Permission Set Overview

External Credential Principal Access

Items per page: 50 Items

View: All

Page 1 of 1

External Credential Principal Access

External Credential Principal Name

Installed Package

UPSAuth - UPSCredentials

More details can be found here - https://help.salesforce.com/s/articleView?id=sf.nc_enable_ext_cred_principal.htm&type=5

Record - UPS Shipping (Shipping Provider)

Field Label	Field Value
Context Name	<Webstore name> e.g. R1Training
Context Locale	<locale> e.g. en_US
Impl Class	UPSShippingProvider
Max Weight Threshold	150
Min Weight Threshold	5
Service	UPS Shipping Service
Account Number	<account number > provider by shipping provider with other credentials
Shipping Options	<pre>1 { 2 "11": "UPS Standard", 3 "12": "UPS 3 Day Select", 4 "13": "UPS Next Day Air Saver", 5 "14": "UPS Next Day Air Early", 6 "54": "UPS Worldwide Express Plus", 7 "59": "UPS 2nd Day Air A.M.", 8 "65": "UPS Worldwide Saver", 9 "70": "UPS Access Point Economy", 10 "82": "UPS Today Standard", 11 "83": "UPS Today Dedicated Courier", 12 "84": "UPS Today Intercity", 13 "85": "UPS Today Express", 14 "86": "UPS Today Express Saver", 15 "96": "UPS Worldwide Express Freight", 16 "01": "UPS Next Day Air", 17 "02": "UPS 2nd Day Air", 18 "03": "UPS Ground ", 19 "07": "UPS Worldwide Express", 20 "08": "UPS Worldwide Expedited" 21 }</pre>
Request	<pre>1 { 2 "RateRequest": { 3 "Request": { 4 "TransactionReference": { 5 "CustomerContext": "" 6 } 7 }, 8 "Shipment": { 9 "Shipper": { 10 "Name": "ShipperName", 11 "Address": { 12 "AddressLine": [13 "ShipperAddressLine" 14],</pre>

```

15         "City": "TIMONIUM",
16         "StateProvinceCode": "MD",
17         "PostalCode": "21093",
18         "CountryCode": "US"
19     }
20 },
21 "ShipTo": {},
22 "ShipFrom": {
23     "Name": "ShipFromName",
24     "Address": {
25         "AddressLine": [
26             "ShipFromAddressLine"
27         ],
28         "City": "TIMONIUM",
29         "StateProvinceCode": "MD",
30         "PostalCode": "21093",
31         "CountryCode": "US"
32     }
33 },
34 "Package": [{
35     "PackagingType": {
36         "Code": "02",
37         "Description": "Packaging"
38     },
39     "PackageWeight": {
40         "UnitOfMeasurement": {
41             "Code": "LBS",
42             "Description": "Pounds"
43         },
44         "Weight": "0"
45     }
46 }]
47 }
48 }
49 }

```

Record - UPS Shipping (HTTP Service)

Field Label	Field Value
End Point	/api/rating/v1/Shop
HTTP Method	GET POST PUT DELETE PATCH HEAD TRACE
Impl Class	<i>This is optional so could be left blank</i>
Named Credentials	UPSShipping
Service Mode	Live

	Mocked
Timeout	5000 (ms)
Success Codes	200,201
Error Codes	400,403,404,500,503
Auth Codes	401
Mocked Response	<pre> { "RateResponse": { "Response": { "ResponseStatus": { "Code": "1", "Description": "Success" }, "Alert": [{ "Code": "119001", "Description": "Additional Handling has automatically been set on Package 1." }, { "Code": "110971", "Description": "Your invoice may vary from the displayed reference rates" }], "TransactionReference": { "CustomerContext": "CustomerContext" } }, "RatedShipment": [{ "Service": { "Code": "01", "Description": "" }, "RatedShipmentAlert": [{ "Code": "119001", "Description": "Additional Handling has automatically been set on Package 1." }, { "Code": "110971", "Description": "Your invoice may vary from the displayed reference rates" }], "BillingWeight": { "UnitOfMeasurement": { "Code": "LBS", "Description": "Pounds" }, "Weight": "150.0" } }] } </pre>

```
"TransportationCharges": {
  "CurrencyCode": "USD",
  "MonetaryValue": "1253.30"
},
"ServiceOptionsCharges": {
  "CurrencyCode": "USD",
  "MonetaryValue": "41.00"
},
"TotalCharges": {
  "CurrencyCode": "USD",
  "MonetaryValue": "1294.30"
},
"GuaranteedDelivery": {
  "BusinessDaysInTransit": "1",
  "DeliveryByTime": "10:30 A.M."
},
"RatedPackage": {
  "TransportationCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "1253.30"
  },
  "ServiceOptionsCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "41.00"
  },
  "TotalCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "1294.30"
  },
  "Weight": "150.0",
  "BillingWeight": {
    "UnitOfMeasurement": {
      "Code": "LBS",
      "Description": "Pounds"
    },
    "Weight": "150.0"
  }
},
{
  "Service": {
    "Code": "02",
    "Description": ""
  },
  "RatedShipmentAlert": [
    {
      "Code": "119001",
      "Description": "Additional Handling has automatically been set on Package 1."
    },
    {
      "Code": "110971",
      "Description": "Your invoice may vary from the displayed reference rates"
    }
  ]
}
```

```
    }
  ],
  "BillingWeight": {
    "UnitOfMeasurement": {
      "Code": "LBS",
      "Description": "Pounds"
    },
    "Weight": "150.0"
  },
  "TransportationCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "520.06"
  },
  "ServiceOptionsCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "41.00"
  },
  "TotalCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "561.06"
  },
  "GuaranteedDelivery": {
    "BusinessDaysInTransit": "2"
  },
  "RatedPackage": {
    "TransportationCharges": {
      "CurrencyCode": "USD",
      "MonetaryValue": "520.06"
    },
    "ServiceOptionsCharges": {
      "CurrencyCode": "USD",
      "MonetaryValue": "41.00"
    },
    "TotalCharges": {
      "CurrencyCode": "USD",
      "MonetaryValue": "561.06"
    },
    "Weight": "150.0",
    "BillingWeight": {
      "UnitOfMeasurement": {
        "Code": "LBS",
        "Description": "Pounds"
      },
      "Weight": "150.0"
    }
  }
},
{
  "Service": {
    "Code": "03",
    "Description": ""
  },
}
```

```
"RatedShipmentAlert": [  
  {  
    "Code": "119001",  
    "Description": "Additional Handling has automatically been set on Package 1."  
  },  
  {  
    "Code": "110971",  
    "Description": "Your invoice may vary from the displayed reference rates"  
  }  
],  
"BillingWeight": {  
  "UnitOfMeasurement": {  
    "Code": "LBS",  
    "Description": "Pounds"  
  },  
  "Weight": "150.0"  
},  
"TransportationCharges": {  
  "CurrencyCode": "USD",  
  "MonetaryValue": "144.14"  
},  
"ServiceOptionsCharges": {  
  "CurrencyCode": "USD",  
  "MonetaryValue": "41.00"  
},  
"TotalCharges": {  
  "CurrencyCode": "USD",  
  "MonetaryValue": "185.14"  
},  
"RatedPackage": {  
  "TransportationCharges": {  
    "CurrencyCode": "USD",  
    "MonetaryValue": "144.14"  
  },  
  "ServiceOptionsCharges": {  
    "CurrencyCode": "USD",  
    "MonetaryValue": "41.00"  
  },  
  "TotalCharges": {  
    "CurrencyCode": "USD",  
    "MonetaryValue": "185.14"  
  },  
  "Weight": "150.0",  
  "BillingWeight": {  
    "UnitOfMeasurement": {  
      "Code": "LBS",  
      "Description": "Pounds"  
    },  
    "Weight": "150.0"  
  }  
}  
}
```

```
{
  "Service": {
    "Code": "12",
    "Description": ""
  },
  "RatedShipmentAlert": [
    {
      "Code": "119001",
      "Description": "Additional Handling has automatically been set on Package 1."
    },
    {
      "Code": "110971",
      "Description": "Your invoice may vary from the displayed reference rates"
    }
  ],
  "BillingWeight": {
    "UnitOfMeasurement": {
      "Code": "LBS",
      "Description": "Pounds"
    },
    "Weight": "150.0"
  },
  "TransportationCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "368.79"
  },
  "ServiceOptionsCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "41.00"
  },
  "TotalCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "409.79"
  },
  "GuaranteedDelivery": {
    "BusinessDaysInTransit": "3"
  },
  "RatedPackage": {
    "TransportationCharges": {
      "CurrencyCode": "USD",
      "MonetaryValue": "368.79"
    },
    "ServiceOptionsCharges": {
      "CurrencyCode": "USD",
      "MonetaryValue": "41.00"
    },
    "TotalCharges": {
      "CurrencyCode": "USD",
      "MonetaryValue": "409.79"
    },
    "Weight": "150.0",
    "BillingWeight": {
```

```
        "UnitOfMeasurement": {
            "Code": "LBS",
            "Description": "Pounds"
        },
        "Weight": "150.0"
    }
}
},
{
    "Service": {
        "Code": "13",
        "Description": ""
    },
    "RatedShipmentAlert": [
        {
            "Code": "119001",
            "Description": "Additional Handling has automatically been set on Package 1."
        },
        {
            "Code": "110971",
            "Description": "Your invoice may vary from the displayed reference rates"
        }
    ],
    "BillingWeight": {
        "UnitOfMeasurement": {
            "Code": "LBS",
            "Description": "Pounds"
        },
        "Weight": "150.0"
    },
    "TransportationCharges": {
        "CurrencyCode": "USD",
        "MonetaryValue": "1226.58"
    },
    "ServiceOptionsCharges": {
        "CurrencyCode": "USD",
        "MonetaryValue": "41.00"
    },
    "TotalCharges": {
        "CurrencyCode": "USD",
        "MonetaryValue": "1267.58"
    },
    "GuaranteedDelivery": {
        "BusinessDaysInTransit": "1",
        "DeliveryByTime": "3:00 P.M."
    },
    "RatedPackage": {
        "TransportationCharges": {
            "CurrencyCode": "USD",
            "MonetaryValue": "1226.58"
        },
        "ServiceOptionsCharges": {
```

```
        "CurrencyCode": "USD",
        "MonetaryValue": "41.00"
    },
    "TotalCharges": {
        "CurrencyCode": "USD",
        "MonetaryValue": "1267.58"
    },
    "Weight": "150.0",
    "BillingWeight": {
        "UnitOfMeasurement": {
            "Code": "LBS",
            "Description": "Pounds"
        },
        "Weight": "150.0"
    }
}
},
{
    "Service": {
        "Code": "14",
        "Description": ""
    },
    "RatedShipmentAlert": [
        {
            "Code": "119001",
            "Description": "Additional Handling has automatically been set on Package 1."
        },
        {
            "Code": "110971",
            "Description": "Your invoice may vary from the displayed reference rates"
        }
    ],
    "BillingWeight": {
        "UnitOfMeasurement": {
            "Code": "LBS",
            "Description": "Pounds"
        },
        "Weight": "150.0"
    },
    "TransportationCharges": {
        "CurrencyCode": "USD",
        "MonetaryValue": "1288.55"
    },
    "ServiceOptionsCharges": {
        "CurrencyCode": "USD",
        "MonetaryValue": "41.00"
    },
    "TotalCharges": {
        "CurrencyCode": "USD",
        "MonetaryValue": "1329.55"
    },
    "GuaranteedDelivery": {
```

```
"BusinessDaysInTransit": "1",
"DeliveryByTime": "8:30 A.M."
},
"RatedPackage": {
  "TransportationCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "1288.55"
  },
  "ServiceOptionsCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "41.00"
  },
  "TotalCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "1329.55"
  },
  "Weight": "150.0",
  "BillingWeight": {
    "UnitOfMeasurement": {
      "Code": "LBS",
      "Description": "Pounds"
    },
    "Weight": "150.0"
  }
},
{
  "Service": {
    "Code": "59",
    "Description": ""
  },
  "RatedShipmentAlert": [
    {
      "Code": "119001",
      "Description": "Additional Handling has automatically been set on Package 1."
    },
    {
      "Code": "110971",
      "Description": "Your invoice may vary from the displayed reference rates"
    }
  ],
  "BillingWeight": {
    "UnitOfMeasurement": {
      "Code": "LBS",
      "Description": "Pounds"
    },
    "Weight": "150.0"
  },
  "TransportationCharges": {
    "CurrencyCode": "USD",
    "MonetaryValue": "634.06"
  },
}
```


	<pre>"ServiceOptionsCharges": { "CurrencyCode": "USD", "MonetaryValue": "41.00" }, "TotalCharges": { "CurrencyCode": "USD", "MonetaryValue": "675.06" }, "GuaranteedDelivery": { "BusinessDaysInTransit": "2", "DeliveryByTime": "10:30 A.M." }, "RatedPackage": { "TransportationCharges": { "CurrencyCode": "USD", "MonetaryValue": "634.06" }, "ServiceOptionsCharges": { "CurrencyCode": "USD", "MonetaryValue": "41.00" }, "TotalCharges": { "CurrencyCode": "USD", "MonetaryValue": "675.06" }, "Weight": "150.0", "BillingWeight": { "UnitOfMeasurement": { "Code": "LBS", "Description": "Pounds" }, "Weight": "150.0" } } }</pre>
--	--

Source Code

Class	Details	Code
UPSShippingProvider	<p>This class extends ShippingProvider class & implements following methods -</p> <p>prepareRequestBody</p> <p>getShippingOptionsFromResponse</p>	<pre>1 public class UPSShippingProvider extends ShippingProvider { 2 public UPSShippingProvider() { 3 } 4 public virtual override void prepareRequestBody(5 ShippingProviderRequest shippingRequest, 6 Map<String, String> callOutRequest 7) { 8 UPSShippingRateRequest requestObject = UPSShippingRateRequest 9 .shippingRequest.shippingMetadata.requestJSON 10 };</pre>

```

11     if (requestObject != null) {
12         requestObject.RateRequest.Shipment.Shipper.ShipperNumber
13         requestObject.RateRequest.Shipment.ShipTo = setShipToAddr
14             shippingRequest
15     };
16     requestObject.RateRequest.Shipment.Package_Z[0]
17         .PackageWeight.Weight = String.valueOf(shippingRequest.
18     String strRatingRequestBody = JSON.serialize(requestObjec
19     strRatingRequestBody = strRatingRequestBody.replaceAll(
20         'Package_Z',
21         'Package'
22     )
23         .replaceAll('Date_Z', 'Date')
24         .replaceAll('Time_Z', 'Time');
25
26     callOutRequest.put(Constants.SERVICE_REQUEST_BODY, strRat
27 }
28 }
29
30 public virtual UPSShippingRateRequest.ShipTo setShipToAddress
31     ShippingProviderRequest shippingRequest
32 ) {
33     UPSShippingRateRequest.Address address = new UPSShippingRat
34     address.AddressLine = new List<String>();
35     address.AddressLine.add(shippingRequest.street);
36     address.City = shippingRequest.city;
37     address.StateProvinceCode = shippingRequest.state;
38     address.PostalCode = shippingRequest.postalCode;
39     address.CountryCode = shippingRequest.country;
40     UPSShippingRateRequest.ShipTo shipTo = new UPSShippingRateR
41     shipTo.Address = address;
42     return shipTo;
43 }
44
45 public virtual override Map<String, ShippingProviderResponse>
46     List<String> responseList,
47     ShippingProviderRequest shippingRequest
48 ) {
49     Map<String, Object> shippingMethods = new Map<String, Objec
50     Map<String, ShippingProviderResponse> shippingMethodsWithRa
51         responseList,
52         shippingRequest
53     );
54     return shippingMethodsWithRate;
55 }
56
57 public virtual Map<String, ShippingProviderResponse> parseUPS
58     List<String> lstUPSResponseBody,
59     ShippingProviderRequest shippingRequest
60 ) {
61     Map<String, ShippingProviderResponse> shippingMethodsWithRa
62     ShippingProviderResponse shippingResponse;
63     if (lstUPSResponseBody != null && lstUPSResponseBody.size()
64         for (String strUPSResponseBody : lstUPSResponseBody) {
65             UPSShippingRateResponse upsShippingRateResponseObj = UP
66                 strUPSResponseBody
67             );
68             UPSShippingRateResponse.RateResponse upsShippingRateRes
69             for (
70                 UPSShippingRateResponse.RatedShipment shipmentObj : u

```

		<pre> 71) { 72 Decimal shippingAmount = shipmentObj.TotalCharges.Mon 73 null 74 ? Decimal.valueOf(shipmentObj.TotalCharges.Monetary 75 : 0; 76 if (shippingMethodsWithRate.containsKey(shipmentObj.S 77 shippingResponse = shippingMethodsWithRate.get(78 shipmentObj.Service.Code 79); 80 shippingResponse.cost = shippingResponse.cost + shi 81 } else { 82 shippingResponse = new ShippingProviderResponse(); 83 shippingResponse.serviceCode = shipmentObj.Service. 84 shippingResponse.cost = shippingAmount; 85 shippingResponse.carrier = shippingRequest.shipping 86 shippingResponse.displayOrder = shippingRequest.shi 87 shippingMethodsWithRate.put(88 shipmentObj.Service.Code, 89 shippingResponse 90); 91 } 92 } 93 } 94 } 95 return shippingMethodsWithRate; 96 } 97 } </pre>
UPSShippingRate Request	<p>This class is a DTO used to prepare UPS Shipping rate request which is posted to UPS to get shipping details</p>	<pre> 1 2 public class UPSShippingRateRequest { 3 4 public class TransactionReference { 5 public String CustomerContext {get;set;} 6 7 public TransactionReference(JSONParser parser) { 8 while (parser.nextToken() != System.JSONToken.END_ 9 if (parser.getCurrentToken() == System.JSONTok 10 String text = parser.getText(); 11 if (parser.nextToken() != System.JSONToken 12 if (text == 'CustomerContext') { 13 CustomerContext = parser.getText() 14 } else { 15 System.debug(LoggingLevel.WARN, 'T 16 consumeObject(parser); 17 } 18 } 19 } 20 } 21 } 22 } 23 24 public RateRequest RateRequest {get;set;} 25 26 public UPSShippingRateRequest(JSONParser parser) { 27 while (parser.nextToken() != System.JSONToken.END_OBJE 28 if (parser.getCurrentToken() == System.JSONToken.F 29 String text = parser.getText(); 30 if (parser.nextToken() != System.JSONToken.VAL 31 if (text == 'RateRequest') { 32 RateRequest = new RateRequest(parser); </pre>

```

33         } else {
34             System.debug(LoggingLevel.WARN, 'UPSSh
35             consumeObject(parser);
36         }
37     }
38 }
39 }
40 }
41
42 public class Address {
43     public List<String> AddressLine {get;set;}
44     public String City {get;set;}
45     public String StateProvinceCode {get;set;}
46     public String PostalCode {get;set;}
47     public String CountryCode {get;set;}
48     public Address() {}
49     public Address(JSONParser parser) {
50         while (parser.nextToken() != System.JSONToken.END_
51             if (parser.getCurrentToken() == System.JSONTok
52                 String text = parser.getText();
53                 if (parser.nextToken() != System.JSONToken
54                     if (text == 'AddressLine') {
55                         AddressLine = arrayOfString(parser
56                     } else if (text == 'City') {
57                         City = parser.getText();
58                     } else if (text == 'StateProvinceCode'
59                         StateProvinceCode = parser.getText
60                     } else if (text == 'PostalCode') {
61                         PostalCode = parser.getText();
62                     } else if (text == 'CountryCode') {
63                         CountryCode = parser.getText();
64                     } else {
65                         System.debug(LoggingLevel.WARN, 'A
66                         consumeObject(parser);
67                     }
68                 }
69             }
70         }
71     }
72 }
73
74 public class Pickup {
75     public String Date_Z {get;set;} // in json: Date
76     public String Time_Z {get;set;} // in json: Time
77
78     public Pickup(JSONParser parser) {
79         while (parser.nextToken() != System.JSONToken.END_
80             if (parser.getCurrentToken() == System.JSONTok
81                 String text = parser.getText();
82                 if (parser.nextToken() != System.JSONToken
83                     if (text == 'Date') {
84                         Date_Z = parser.getText();
85                     } else if (text == 'Time') {
86                         Time_Z = parser.getText();
87                     } else {
88                         System.debug(LoggingLevel.WARN, 'P
89                         consumeObject(parser);
90                     }
91                 }
92             }

```

```

93         }
94     }
95 }
96
97 public class ShipmentTotalWeight {
98     public CustomerClassification UnitOfMeasurement {get;set;}
99     public String Weight {get;set;}
100
101     public ShipmentTotalWeight(JSONParser parser) {
102         while (parser.nextToken() != System.JSONToken.END_
103             if (parser.getCurrentToken() == System.JSONTok
104                 String text = parser.getText();
105                 if (parser.nextToken() != System.JSONToken
106                     if (text == 'UnitOfMeasurement') {
107                         UnitOfMeasurement = new CustomerCl
108                     } else if (text == 'Weight') {
109                         Weight = parser.getText();
110                     } else {
111                         System.debug(LoggingLevel.WARN, 'S
112                             consumeObject(parser);
113                     }
114                 }
115             }
116         }
117     }
118 }
119
120 public class ShipmentCharges {
121     public BillShipper BillShipper {get;set;}
122
123     public ShipmentCharges(JSONParser parser) {
124         while (parser.nextToken() != System.JSONToken.END_
125             if (parser.getCurrentToken() == System.JSONTok
126                 String text = parser.getText();
127                 if (parser.nextToken() != System.JSONToken
128                     if (text == 'BillShipper') {
129                         BillShipper = new BillShipper(pars
130                     } else {
131                         System.debug(LoggingLevel.WARN, 'S
132                             consumeObject(parser);
133                     }
134                 }
135             }
136         }
137     }
138 }
139
140 public class Request {
141     public TransactionReference TransactionReference {get;
142
143     public Request(JSONParser parser) {
144         while (parser.nextToken() != System.JSONToken.END_
145             if (parser.getCurrentToken() == System.JSONTok
146                 String text = parser.getText();
147                 if (parser.nextToken() != System.JSONToken
148                     if (text == 'TransactionReference') {
149                         TransactionReference = new Transac
150                     } else {
151                         System.debug(LoggingLevel.WARN, 'R
152                             consumeObject(parser);

```



```

213         } else if (text == 'PaymentDetails') {
214             PaymentDetails = arrayOfPaymentDet
215         } else if (text == 'ShipmentRatingOpti
216             ShipmentRatingOptions = new Shipme
217         } else if (text == 'Service') {
218             Service = new CustomerClassificati
219         } else if (text == 'ShipmentTotalWeigh
220             ShipmentTotalWeight = new Shipment
221         } else if (text == 'NumOfPieces') {
222             NumOfPieces = parser.getText();
223         } else if (text == 'Package') {
224             Package_Z = arrayOfPackage_Z(parse
225         } else if (text == 'DeliveryTimeInform
226             DeliveryTimeInformation = new Deli
227         } else if (text == 'ShipmentCharges')
228             ShipmentCharges = new ShipmentChar
229         } else {
230             System.debug(LoggingLevel.WARN, 'S
231             consumeObject(parser);
232         }
233     }
234 }
235 }
236 }
237 }
238
239 public class ShipmentCharge {
240     public String Type {get;set;}
241     public BillThirdParty BillThirdParty {get;set;}
242
243     public ShipmentCharge(JSONParser parser) {
244         while (parser.nextToken() != System.JSONToken.END_
245             if (parser.getCurrentToken() == System.JSONTok
246                 String text = parser.getText();
247                 if (parser.nextToken() != System.JSONToken
248                     if (text == 'Type') {
249                         Type = parser.getText();
250                     } else if (text == 'BillThirdParty') {
251                         BillThirdParty = new BillThirdPart
252                     } else {
253                         System.debug(LoggingLevel.WARN, 'S
254                         consumeObject(parser);
255                     }
256                 }
257             }
258         }
259     }
260 }
261
262 public class CustomerClassification {
263     public String Code {get;set;}
264     public String Description {get;set;}
265
266     public CustomerClassification(JSONParser parser) {
267         while (parser.nextToken() != System.JSONToken.END_
268             if (parser.getCurrentToken() == System.JSONTok
269                 String text = parser.getText();
270                 if (parser.nextToken() != System.JSONToken
271                     if (text == 'Code') {
272                         Code = parser.getText();

```

```

273         } else if (text == 'Description') {
274             Description = parser.getText();
275         } else {
276             System.debug(LoggingLevel.WARN, 'C
277             consumeObject(parser);
278         }
279     }
280 }
281 }
282 }
283 }
284
285 public class BillShipper {
286     public String AccountNumber {get;set;}
287
288     public BillShipper(JSONParser parser) {
289         while (parser.nextToken() != System.JSONToken.END_
290             if (parser.getCurrentToken() == System.JSONTok
291                 String text = parser.getText();
292                 if (parser.nextToken() != System.JSONToken
293                     if (text == 'AccountNumber') {
294                         AccountNumber = parser.getText();
295                     } else {
296                         System.debug(LoggingLevel.WARN, 'B
297                         consumeObject(parser);
298                     }
299                 }
300             }
301         }
302     }
303 }
304
305 public class DeliveryTimeInformation {
306     public String PackageBillType {get;set;}
307     public Pickup Pickup {get;set;}
308
309     public DeliveryTimeInformation(JSONParser parser) {
310         while (parser.nextToken() != System.JSONToken.END_
311             if (parser.getCurrentToken() == System.JSONTok
312                 String text = parser.getText();
313                 if (parser.nextToken() != System.JSONToken
314                     if (text == 'PackageBillType') {
315                         PackageBillType = parser.getText()
316                     } else if (text == 'Pickup') {
317                         Pickup = new Pickup(parser);
318                     } else {
319                         System.debug(LoggingLevel.WARN, 'D
320                         consumeObject(parser);
321                     }
322                 }
323             }
324         }
325     }
326 }
327
328 public class Shipper {
329     public String Name {get;set;}
330     public String ShipperNumber {get;set;}
331     public Address Address {get;set;}
332

```



```

333     public Shipper(JSONParser parser) {
334         while (parser.nextToken() != System.JSONToken.END_)
335             if (parser.getCurrentToken() == System.JSONTok
336                 String text = parser.getText();
337                 if (parser.nextToken() != System.JSONToken
338                     if (text == 'Name') {
339                         Name = parser.getText();
340                     } else if (text == 'ShipperNumber') {
341                         ShipperNumber = parser.getText();
342                     } else if (text == 'Address') {
343                         Address = new Address(parser);
344                     } else {
345                         System.debug(LoggingLevel.WARN, 'S
346                         consumeObject(parser);
347                     }
348                 }
349             }
350         }
351     }
352 }
353
354 public class ShipTo {
355     public String Name {get;set;}
356     public Address Address {get;set;}
357     public ShipTo() {
358     }
359     public ShipTo(JSONParser parser) {
360         while (parser.nextToken() != System.JSONToken.END_)
361             if (parser.getCurrentToken() == System.JSONTok
362                 String text = parser.getText();
363                 if (parser.nextToken() != System.JSONToken
364                     if (text == 'Name') {
365                         Name = parser.getText();
366                     } else if (text == 'Address') {
367                         Address = new Address(parser);
368                     } else {
369                         System.debug(LoggingLevel.WARN, 'S
370                         consumeObject(parser);
371                     }
372                 }
373             }
374         }
375     }
376 }
377
378 public class PaymentDetails {
379     public ShipmentCharge ShipmentCharge {get;set;}
380
381     public PaymentDetails(JSONParser parser) {
382         while (parser.nextToken() != System.JSONToken.END_)
383             if (parser.getCurrentToken() == System.JSONTok
384                 String text = parser.getText();
385                 if (parser.nextToken() != System.JSONToken
386                     if (text == 'ShipmentCharge') {
387                         ShipmentCharge = new ShipmentCharg
388                     } else {
389                         System.debug(LoggingLevel.WARN, 'P
390                         consumeObject(parser);
391                     }
392             }

```

```

393         }
394     }
395 }
396 }
397
398 public class Package_Z {
399     public CustomerClassification SimpleRate {get;set;}
400     public CustomerClassification PackagingType {get;set;}
401     public Dimensions Dimensions {get;set;}
402     public ShipmentTotalWeight PackageWeight {get;set;}
403     public String OversizeIndicator {get;set;}
404     public String MinimumBillableWeightIndicator {get;set;}
405
406     public Package_Z(JSONParser parser) {
407         while (parser.nextToken() != System.JSONToken.END_
408             if (parser.getCurrentToken() == System.JSONTok
409                 String text = parser.getText();
410                 if (parser.nextToken() != System.JSONToken
411                     if (text == 'SimpleRate') {
412                         SimpleRate = new CustomerClassific
413                     } else if (text == 'PackagingType') {
414                         PackagingType = new CustomerClassi
415                     } else if (text == 'Dimensions') {
416                         Dimensions = new Dimensions(parser
417                     } else if (text == 'PackageWeight') {
418                         PackageWeight = new ShipmentTotalw
419                     } else if (text == 'OversizeIndicator'
420                         OversizeIndicator = parser.getText
421                     } else if (text == 'MinimumBillableWei
422                         MinimumBillableWeightIndicator = p
423                     } else {
424                         System.debug(LoggingLevel.WARN, 'P
425                         consumeObject(parser);
426                     }
427                 }
428             }
429         }
430     }
431 }
432
433 public class RateRequest {
434     public Request Request {get;set;}
435     public CustomerClassification CustomerClassification {
436     public Shipment Shipment {get;set;}
437
438     public RateRequest(JSONParser parser) {
439         while (parser.nextToken() != System.JSONToken.END_
440             if (parser.getCurrentToken() == System.JSONTok
441                 String text = parser.getText();
442                 if (parser.nextToken() != System.JSONToken
443                     if (text == 'Request') {
444                         Request = new Request(parser);
445                     } else if (text == 'CustomerClassifica
446                         CustomerClassification = new Custo
447                     } else if (text == 'Shipment') {
448                         Shipment = new Shipment(parser);
449                     } else {
450                         System.debug(LoggingLevel.WARN, 'R
451                         consumeObject(parser);
452                     }

```



```

513         } else {
514             System.debug(LoggingLevel.WARN, 'A
515             consumeObject(parser);
516         }
517     }
518 }
519 }
520 }
521 }
522
523 public class ShipmentRatingOptions {
524     public String TPFCNegotiatedRatesIndicator {get;set;}
525     public String NegotiatedRatesIndicator {get;set;}
526
527     public ShipmentRatingOptions(JSONParser parser) {
528         while (parser.nextToken() != System.JSONToken.END_
529             if (parser.getCurrentToken() == System.JSONTok
530                 String text = parser.getText();
531                 if (parser.nextToken() != System.JSONToken
532                     if (text == 'TPFCNegotiatedRatesIndica
533                         TPFCNegotiatedRatesIndicator = par
534                     } else if (text == 'NegotiatedRatesInd
535                         NegotiatedRatesIndicator = parser.
536                     } else {
537                         System.debug(LoggingLevel.WARN, 'S
538                         consumeObject(parser);
539                     }
540                 }
541             }
542         }
543     }
544 }
545
546
547 public static UPSShippingRateRequest parse(String json) {
548     System.JSONParser parser = System.JSON.createParser(js
549     return new UPSShippingRateRequest(parser);
550 }
551
552 public static void consumeObject(System.JSONParser parser)
553     Integer depth = 0;
554     do {
555         System.JSONToken curr = parser.getCurrentToken();
556         if (curr == System.JSONToken.START_OBJECT ||
557             curr == System.JSONToken.START_ARRAY) {
558             depth++;
559         } else if (curr == System.JSONToken.END_OBJECT ||
560             curr == System.JSONToken.END_ARRAY) {
561             depth--;
562         }
563     } while (depth > 0 && parser.nextToken() != null);
564 }
565
566
567
568
569
570 private static List<Package_Z> arrayOfPackage_Z(System.JSO
571     List<Package_Z> res = new List<Package_Z>();
572     if (p.getCurrentToken() == null) p.nextToken();

```

		<pre> 573 while (p.nextToken() != System.JSONToken.END_ARRAY) { 574 res.add(new Package_Z(p)); 575 } 576 return res; 577 } 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 private static List<String> arrayOfString(System.JSONParse 595 List<String> res = new List<String>(); 596 if (p.getCurrentToken() == null) p.nextToken(); 597 while (p.nextToken() != System.JSONToken.END_ARRAY) { 598 res.add(p.getText()); 599 } 600 return res; 601 } 602 603 604 605 606 607 private static List<PaymentDetails> arrayOfPaymentDetails(608 List<PaymentDetails> res = new List<PaymentDetails>(); 609 if (p.getCurrentToken() == null) p.nextToken(); 610 while (p.nextToken() != System.JSONToken.END_ARRAY) { 611 res.add(new PaymentDetails(p)); 612 } 613 return res; 614 } 615 616 617 618 } 619 </pre>
UPSShippingRate Response	This class is a DTO used to parse shipping rate response returned by UPS	<pre> 1 global with sharing class UPSShippingRateResponse { 2 global class Response { 3 global ResponseStatus ResponseStatus; 4 //global List<ResponseStatus> Alert; 5 global TransactionReference TransactionReference; 6 } 7 8 global class TransactionReference { 9 global String CustomerContext; 10 } 11 12 global RateResponse RateResponse; </pre>

```

13
14 global class ResponseStatus {
15     global String Code;
16     global String Description;
17 }
18
19 global class BillingWeight {
20     global ResponseStatus UnitOfMeasurement;
21     global String Weight;
22 }
23
24 global class GuaranteedDelivery {
25     global String BusinessDaysInTransit;
26 }
27
28 global class RateResponse {
29     global Response Response;
30     global List<RatedShipment> RatedShipment;
31 }
32
33 global class TransportationCharges {
34     global String CurrencyCode;
35     global String MonetaryValue;
36 }
37
38 global class RatedPackage {
39     global TransportationCharges TransportationCharges;
40     global TransportationCharges ServiceOptionsCharges;
41     global TransportationCharges TotalCharges;
42     global String Weight;
43     global BillingWeight BillingWeight;
44 }
45
46 global class GuaranteedDelivery_Z {
47     global String BusinessDaysInTransit;
48     global String DeliveryByTime;
49 }
50
51 global class RatedShipment {
52     global ResponseStatus Service;
53     //global List<ResponseStatus> RatedShipmentAlert;
54     global BillingWeight BillingWeight;
55     global TransportationCharges TransportationCharges;
56     global TransportationCharges ServiceOptionsCharges;
57     global TransportationCharges TotalCharges;
58     global RatedPackage RatedPackage;
59     global GuaranteedDelivery GuaranteedDelivery;
60 }
61
62 global static UPSShippingRateResponse parse(String json) {
63     return (UPSShippingRateResponse) System.JSON.deserialize(
64         json,
65         UPSShippingRateResponse.class
66     );
67 }
68 }
69

```