

---

# **python-fedex Documentation**

***Release 2.3.0***

**2015, Python Fedex Devs**

**Nov 07, 2017**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Quick Start</b>	<b>5</b>
2.1	Initialize Config Object . . . . .	5
2.2	Create and Delete a Shipment . . . . .	5
2.3	Get the Response . . . . .	6
2.4	Validate an Address . . . . .	7
2.5	Estimate Delivery Time . . . . .	7
2.6	Validate Postal Code . . . . .	7
2.7	FedEx Store Search . . . . .	7
2.8	Get Package Rate . . . . .	7
2.9	Track a Package . . . . .	8
2.10	Get JSON and Dictionary Response . . . . .	8
2.11	Advanced Requests . . . . .	8
<b>3</b>	<b>Implemented Services</b>	<b>11</b>
3.1	Ship Service . . . . .	11
3.2	Rate Service . . . . .	12
3.3	Validation Availability And Commitment Service . . . . .	13
3.4	Track Service . . . . .	14
3.5	Address Validation Service . . . . .	14
3.6	Location Service . . . . .	15
3.7	Country Service . . . . .	15
3.8	Pickup Service . . . . .	16
3.9	Package Movement Service . . . . .	16
<b>4</b>	<b>Format Conversion Tools</b>	<b>19</b>
4.1	Basic Suds Object to Dictionary . . . . .	19
4.2	Advanced Suds Object to Dictionary . . . . .	19
4.3	Suds Object to JSON . . . . .	19
<b>5</b>	<b>Config Object</b>	<b>21</b>
<b>6</b>	<b>Base Service Class</b>	<b>23</b>
<b>7</b>	<b>Exception Classes</b>	<b>25</b>
7.1	Fedex Base Service Exception . . . . .	25

7.2	Fedex Failure . . . . .	25
7.3	Fedex Error . . . . .	25
<b>8</b>	<b>General Suds Plugin</b>	<b>27</b>
<b>9</b>	<b>Change Log</b>	<b>29</b>
9.1	2.4.0 . . . . .	29
9.2	2.3.0 . . . . .	29
9.3	2.2.0 . . . . .	29
9.4	2.1.0 . . . . .	30
9.5	2.0.0 . . . . .	30
9.6	1.1.1 . . . . .	30
9.7	1.1.0 . . . . .	30
9.8	1.0.14 . . . . .	31
9.9	1.0.12 and 1.0.13 . . . . .	31
9.10	1.0.11 . . . . .	31
9.11	1.0.10 . . . . .	31
9.12	1.0.9 . . . . .	31
9.13	1.0.8 . . . . .	31
9.14	1.0.7 . . . . .	31
9.15	1.0.6 . . . . .	31
9.16	1.0.4 . . . . .	32
9.17	1.0.3 . . . . .	32
<b>10</b>	<b>Indices and tables</b>	<b>33</b>

Contents:



# CHAPTER 1

---

## Installation

---

python-fedex is compatible with Python 2.7+. The only other requirement is [suds\\_jurko](#) although the module is also compatible with the original suds module.

For those on Linux/Unix Mac OS, the easiest route will be **pip** or **easy\_install**:

```
pip install fedex
```





To get you started, a comprehensive set of examples and tests is included in the examples and tests directory respectively. The examples below are come from the included unit tests.

### 2.1 Initialize Config Object

```
from fedex.config import FedexConfig
CONFIG_OBJ = FedexConfig(key='<key>',
                          password='<pass>',
                          account_number='<account_no>',
                          meter_number='<meter_no>')
```

### 2.2 Create and Delete a Shipment

```
shipment = FedexProcessShipmentRequest(CONFIG_OBJ)

shipment.RequestedShipment.DropoffType = 'REGULAR_PICKUP'
shipment.RequestedShipment.ServiceType = 'FEDEX_GROUND'
shipment.RequestedShipment.PackagingType = 'YOUR_PACKAGING'

shipment.RequestedShipment.Shipper.Contact.PersonName = 'Sender Name'
shipment.RequestedShipment.Shipper.Contact.PhoneNumber = '9012638716'

shipment.RequestedShipment.Shipper.Address.StreetLines = ['Address Line 1']
shipment.RequestedShipment.Shipper.Address.City = convert_to_utf8('Herndon')
shipment.RequestedShipment.Shipper.Address.StateOrProvinceCode = 'VA'
shipment.RequestedShipment.Shipper.Address.PostalCode = '20171'
shipment.RequestedShipment.Shipper.Address.CountryCode = 'US'

shipment.RequestedShipment.Recipient.Contact.PersonName = 'Recipient Name'
shipment.RequestedShipment.Recipient.Contact.PhoneNumber = '9012637906'
```

```
shipment.RequestedShipment.Recipient.Address.StreetLines = ['Address Line 1']
shipment.RequestedShipment.Recipient.Address.City = convert_to_utf8(
↳ 'TorontoâââHerndon')
shipment.RequestedShipment.Recipient.Address.StateOrProvinceCode = 'VA'
shipment.RequestedShipment.Recipient.Address.PostalCode = '20171'
shipment.RequestedShipment.Recipient.Address.CountryCode = 'US'
shipment.RequestedShipment.EdtRequestType = 'NONE'

shipment.RequestedShipment.ShippingChargesPayment.Payor.ResponsibleParty.
↳ AccountNumber \
    = CONFIG_OBJ.account_number

shipment.RequestedShipment.ShippingChargesPayment.PaymentType = 'SENDER'

shipment.RequestedShipment.LabelSpecification.LabelFormatType = 'COMMON2D'
shipment.RequestedShipment.LabelSpecification.ImageType = 'PNG'
shipment.RequestedShipment.LabelSpecification.LabelStockType = 'PAPER_7X4.75'
shipment.RequestedShipment.LabelSpecification.LabelPrintingOrientation = 'BOTTOM_EDGE_
↳ OF_TEXT_FIRST'

# Use order if setting multiple labels or delete
del shipment.RequestedShipment.LabelSpecification.LabelOrder

package1_weight = shipment.create_wsdl_object_of_type('Weight')
package1_weight.Value = 2.0
package1_weight.Units = "LB"
package1 = shipment.create_wsdl_object_of_type('RequestedPackageLineItem')
package1.PhysicalPackaging = 'ENVELOPE'
package1.Weight = package1_weight
shipment.add_package(package1)

shipment.send_validation_request()
shipment.send_request()

assert shipment.response
assert shipment.response.HighestSeverity == 'SUCCESS'
track_id = shipment.response.CompletedShipmentDetail.CompletedPackageDetails[0].
↳ TrackingIds[0].TrackingNumber
assert track_id

del_shipment = FedexDeleteShipmentRequest(CONFIG_OBJ)
del_shipment.DeletionControlType = "DELETE_ALL_PACKAGES"
del_shipment.TrackingId.TrackingNumber = track_id
del_shipment.TrackingId.TrackingIdType = 'EXPRESS'

del_shipment.send_request()
```

## 2.3 Get the Response

Once you perform `send_request()`, you can access the `response` attribute of the service request object. To access the response from the above delete shipment example:

```
request_response = del_shipment.response
```

## 2.4 Validate an Address

```
avs_request = FedexAddressValidationRequest (CONFIG_OBJ)
address1 = avs_request.create_wsdl_object_of_type('AddressToValidate')
address1.Address.StreetLines = ['155 Old Greenville Hwy', 'Suite 103']
address1.Address.City = 'Clemson'
address1.Address.StateOrProvinceCode = 'SC'
address1.Address.PostalCode = 29631
address1.Address.CountryCode = 'US'
address1.Address.Residential = False
avs_request.add_address(address1)

avs_request.send_request()
```

## 2.5 Estimate Delivery Time

```
avc_request = FedexAvailabilityCommitmentRequest (CONFIG_OBJ)
avc_request.Origin.PostalCode = 'M5V 3A4'
avc_request.Origin.CountryCode = 'CA'
avc_request.Destination.PostalCode = '27577' # 29631
avc_request.Destination.CountryCode = 'US'
```

## 2.6 Validate Postal Code

```
inquiry = FedexValidatePostalRequest (CONFIG_OBJ)
inquiry.Address.PostalCode = '29631'
inquiry.Address.CountryCode = 'US'

inquiry.send_request()
```

## 2.7 FedEx Store Search

```
location_request = FedexSearchLocationRequest (CONFIG_OBJ)
location_request.Address.PostalCode = '38119'
location_request.Address.CountryCode = 'US'
```

## 2.8 Get Package Rate

```
rate = FedexRateServiceRequest (CONFIG_OBJ)

rate.RequestedShipment.DropoffType = 'REGULAR_PICKUP'
rate.RequestedShipment.ServiceType = 'FEDEX_GROUND'
rate.RequestedShipment.PackagingType = 'YOUR_PACKAGING'

rate.RequestedShipment.Shipper.Address.StateOrProvinceCode = 'SC'
rate.RequestedShipment.Shipper.Address.PostalCode = '29631'
```

```
rate.RequestedShipment.Shipper.Address.CountryCode = 'US'

rate.RequestedShipment.Recipient.Address.StateOrProvinceCode = 'NC'
rate.RequestedShipment.Recipient.Address.PostalCode = '27577'
rate.RequestedShipment.Recipient.Address.CountryCode = 'US'

rate.RequestedShipment.EdtRequestType = 'NONE'
rate.RequestedShipment.ShippingChargesPayment.PaymentType = 'SENDER'

package1_weight = rate.create_wsdl_object_of_type('Weight')
package1_weight.Value = 1.0
package1_weight.Units = "LB"
package1 = rate.create_wsdl_object_of_type('RequestedPackageLineItem')
package1.Weight = package1_weight
package1.PhysicalPackaging = 'BOX'
package1.GroupPackageCount = 1
rate.add_package(package1)

rate.send_request()
```

## 2.9 Track a Package

```
track = FedexTrackRequest(CONFIG_OBJ)
tracking_num = '781820562774'
track.SelectionDetails.PackageIdentifier.Type = 'TRACKING_NUMBER_OR_DOORTAG'
track.SelectionDetails.PackageIdentifier.Value = tracking_num
track.send_request()
```

## 2.10 Get JSON and Dictionary Response

```
from fedex.tools.conversion import subject_to_dict
from fedex.tools.conversion import subject_to_json
response_dict = subject_to_dict(shipment.response)
response_json = subject_to_json(shipment.response)
```

## 2.11 Advanced Requests

Since this module is a suds wrapper, you can browse each service's WSDL in `fedex/wsdl`s and add any required objects to your request.

For example, the *Ship Service* WSDL has a *simpleType* definition for *ServiceType* with the following options:

- 'STANDARD\_OVERNIGHT'
- 'PRIORITY\_OVERNIGHT'
- 'FEDEX\_GROUND'
- 'FEDEX\_EXPRESS\_SAVER'
- 'FEDEX\_2\_DAY'

- 'INTERNATIONAL\_PRIORITY'
- 'SAME\_DAY'
- 'INTERNATIONAL\_ECONOMY'

You can see all what is available for a specific definition by browsing a service's WSDL. It is possible to customize your request beyond what is included in the examples but still within the confines of this package. If it's something common enough, please bring it forward so that it can be included along as an example.



---

## Implemented Services

---

python-fedex implements various [FedEx Web Services](#). These services along with the implemented functionality is described below.

### 3.1 Ship Service

Generates (FedEx Waybill) and deletes shipments.

**class** `fedex.services.ship_service.FedexProcessShipmentRequest` (*config\_obj*, \*args, \*\*kwargs)

Bases: `fedex.base_service.FedexBaseService`

This class allows you to process (create) a new FedEx shipment. You will need to populate the data structures in `self.RequestedShipment`, then send the request. Label printing is supported and very configurable, returning an ASCII representation with the response as well.

The optional keyword args detailed on `L{FedexBaseService}` apply here as well.

@type `config_obj`: `L{FedexConfig}` @param `config_obj`: A valid `FedexConfig` object.

**add\_package** (*package\_item*)

Adds a package to the ship request.

@type `package_item`: WSDL object, type of `RequestedPackageLineItem` WSDL object.

@keyword `package_item`: A `RequestedPackageLineItem`, created by calling `create_wsdl_object_of_type('RequestedPackageLineItem')` on this `ShipmentRequest` object. See `examples/create_shipment.py` for more details.

**create\_wsdl\_object\_of\_type** (*type\_name*)

Creates and returns a WSDL object of the specified type. :param `type_name`: specifies the object's type name from WSDL.

**send\_request** (*send\_function=None*)

Sends the assembled request on the child object. @type `send_function`: function reference @keyword `send_function`: A function reference (passed without the

parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

**send\_validation\_request()**

This is very similar to just sending the shipment via the typical `send_request()` function, but this doesn't create a shipment. It is used to make sure "good" values are given by the user or the application using the library.

**RequestedShipment = None**

@ivar: Holds the RequestedShipment WSDL object.

**class** `fedex.services.ship_service.FedexDeleteShipmentRequest` (*config\_obj*, *\*args*,  
*\*\*kwargs*)

Bases: `fedex.base_service.FedexBaseService`

This class allows you to delete a shipment, given a tracking number.

Deletes a shipment via a tracking number.

**create\_wsdl\_object\_of\_type** (*type\_name*)

Creates and returns a WSDL object of the specified type. :param *type\_name*: specifies the object's type name from WSDL.

**send\_request** (*send\_function=None*)

Sends the assembled request on the child object. @type *send\_function*: function reference @keyword *send\_function*: A function reference (passed without the

parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

**DeletionControlType = None**

@ivar: Holds the DeletionControlType WSDL object.

**TrackingId = None**

@ivar: Holds the TrackingId WSDL object.

## 3.2 Rate Service

Calculates the cost of a shipment.

**class** `fedex.services.rate_service.FedexRateServiceRequest` (*config\_obj*, *\*args*,  
*\*\*kwargs*)

Bases: `fedex.base_service.FedexBaseService`

This class allows you to get the shipping charges for a particular address. You will need to populate the data structures in `self.RequestedShipment`, then send the request.

The optional keyword args detailed on `L{FedexBaseService}` apply here as well.

@type *config\_obj*: `L{FedexConfig}` @param *config\_obj*: A valid `FedexConfig` object.

**add\_package** (*package\_item*)

Adds a package to the ship request.

@type *package\_item*: WSDL object, type of `RequestedPackageLineItem` WSDL object.

@keyword *package\_item*: A `RequestedPackageLineItem`, created by calling `create_wsdl_object_of_type('RequestedPackageLineItem')` on this `ShipmentRequest` object. See `examples/create_shipment.py` for more details.



**create\_wsdl\_object\_of\_type** (*type\_name*)

Creates and returns a WSDL object of the specified type. :param type\_name: specifies the object's type name from WSDL.

**send\_request** (*send\_function=None*)

Sends the assembled request on the child object. @type send\_function: function reference @keyword send\_function: A function reference (passed without the parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

**RequestedShipment = None**

@ivar: Holds the RequestedShipment WSDL object including the shipper, recipient and shipt time.

### 3.3 Validation Availability And Commitment Service

Calculates the estimated arrival time of a shipment.

```
class fedex.services.availability_commitment_service.FedexAvailabilityCommitmentRequest (config_obj, *args, **kwargs)
```

Bases: *fedex.base\_service.FedexBaseService*

This class allows you validate service availability

@type config\_obj: L{FedexConfig} @param config\_obj: A valid FedexConfig object.

**create\_wsdl\_object\_of\_type** (*type\_name*)

Creates and returns a WSDL object of the specified type. :param type\_name: specifies the object's type name from WSDL.

**send\_request** (*send\_function=None*)

Sends the assembled request on the child object. @type send\_function: function reference @keyword send\_function: A function reference (passed without the parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

**CarrierCode = None**

@ivar: Carrier Code Default to Fedex (FDXE), or can bbe FDXG.

**Destination = None**

@ivar: Holds Destination Address WSDL object.

**Origin = None**

@ivar: Holds Origin Address WSDL object.

**Packaging = None**

@ivar: Type of packaging to narrow down available shipping options or defaults to YOUR\_PACKAGING.

**Service = None**

@ivar: Service type, if set to None will get all available service information.

**ShipDate = None**

@ivar: Ship Date date WSDL object.

## 3.4 Track Service

Returns tracking information for a given tracking number.

**class** `fedex.services.track_service.FedexTrackRequest` (*config\_obj*, \**args*, \*\**kwargs*)

Bases: `fedex.base_service.FedexBaseService`

This class allows you to track shipments by providing a tracking number or other identifying features. By default, you can simply pass a tracking number to the constructor. If you would like to query shipments based on something other than tracking number, you will want to read the documentation for the `L{__init__}` method. Particularly, the `tracking_value` and `package_identifier` arguments.

Sends a shipment tracking request. The optional keyword args detailed on `L{FedexBaseService}` apply here as well.

@type `config_obj`: `L{FedexConfig}` @param `config_obj`: A valid `FedexConfig` object.

**create\_wsdl\_object\_of\_type** (*type\_name*)

Creates and returns a WSDL object of the specified type. :param `type_name`: specifies the object's type name from WSDL.

**send\_request** (*send\_function=None*)

Sends the assembled request on the child object. @type `send_function`: function reference @keyword `send_function`: A function reference (passed without the

parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

**ProcessingOptions = None**

@ivar: Holds the `TrackRequestProcessingOptionType` WSDL object that defaults to `None`.

**SelectionDetails = None**

@ivar: Holds the `SelectionDetails` WSDL object that includes tracking type and value.

## 3.5 Address Validation Service

Validates and cleans a given address.

**class** `fedex.services.address_validation_service.FedexAddressValidationRequest` (*config\_obj*, \**args*, \*\**kwargs*)

Bases: `fedex.base_service.FedexBaseService`

This class allows you validate anywhere from one to a hundred addresses in one go. Create `AddressToValidate` WSDL objects and add them to each instance of this request using `add_address()`.

@type `config_obj`: `L{FedexConfig}` @param `config_obj`: A valid `FedexConfig` object.

**add\_address** (*address\_item*)

Adds an address to `self.AddressesToValidate`.

@type `address_item`: WSDL object, type of `AddressToValidate` WSDL object. @keyword `address_item`: A `AddressToValidate`, created by

calling `create_wsdl_object_of_type('AddressToValidate')` on this `FedexAddressValidationRequest` object. See `examples/create_shipment.py` for more details.

**create\_wsdl\_object\_of\_type** (*type\_name*)

Creates and returns a WSDL object of the specified type. :param `type_name`: specifies the object's type name from WSDL.

**send\_request** (*send\_function=None*)

Sends the assembled request on the child object. @type send\_function: function reference @keyword send\_function: A function reference (passed without the parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

**AddressesToValidate = None**

@ivar: Holds the AddressToValidate WSDL object.

## 3.6 Location Service

Returns FedEx store locations based on a given location query.

```
class fedex.services.location_service.FedexSearchLocationRequest (config_obj,
                                                                *args,
                                                                **kwargs)
```

Bases: *fedex.base\_service.FedexBaseService*

This class allows you to figure out a FedEx location closest to a specified location criteria, based on location type. The response includes location details like operating times, directions and a map link and more.

@type config\_obj: L{FedexConfig} @param config\_obj: A valid FedexConfig object.

**create\_wSDL\_object\_of\_type** (*type\_name*)

Creates and returns a WSDL object of the specified type. :param type\_name: specifies the object's type name from WSDL.

**send\_request** (*send\_function=None*)

Sends the assembled request on the child object. @type send\_function: function reference @keyword send\_function: A function reference (passed without the parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

**Address = None**

@ivar: Holds the Address WSDL object.

**Constraints = None**

@ivar: Holds a list of SearchLocationConstraints WSDL objects.

**LocationsSearchCriterion = None**

@ivar: Holds the LocationsSearchCriteriaType WSDL object.

**MultipleMatchesAction = None**

@ivar: Holds the MultipleMatchesActionType WSDL object.

**PhoneNumber = None**

@ivar: Holds the PhoneNumber string object.

**SortDetail = None**

@ivar: Holds the LocationSortDetail WSDL object.

## 3.7 Country Service

Validates the postal codes for a given address.

```
class fedex.services.country_service.FedexValidatePostalRequest (config_obj, *args,
                                                                **kwargs)
```

Bases: *fedex.base\_service.FedexBaseService*

This class allows you validate an address. [https://www.fedex.com/us/developer/WebHelp/ws/2015/html/WebServicesHelp/WSDVG/47\\_Country\\_Service.htm](https://www.fedex.com/us/developer/WebHelp/ws/2015/html/WebServicesHelp/WSDVG/47_Country_Service.htm)

@type config\_obj: L{FedexConfig} @param config\_obj: A valid FedexConfig object.

**create\_wsdl\_object\_of\_type** (type\_name)

Creates and returns a WSDL object of the specified type. :param type\_name: specifies the object's type name from WSDL.

**send\_request** (send\_function=None)

Sends the assembled request on the child object. @type send\_function: function reference @keyword send\_function: A function reference (passed without the parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

**Address = None**

@ivar: Holds Address WSDL objects.

**CarrierCode = None**

@ivar: Carrier Code Default to Fedex (FDXE), or can be FDYG.

**CheckForMismatch = None**

@ivar: Holds the CheckForMismatch boolean objects.

**RoutingCode = None**

@ivar: Routing Code Default to FDSY.

**ShipDateTime = None**

@ivar: Holds the ShipDateTime date time objects.

## 3.8 Pickup Service

Creates a fedex pickup request.

```
class fedex.services.pickup_service.FedexCreatePickupRequest (config_obj, *args,
                                                                **kwargs)
```

Bases: *fedex.base\_service.FedexBaseService*

**create\_wsdl\_object\_of\_type** (type\_name)

Creates and returns a WSDL object of the specified type. :param type\_name: specifies the object's type name from WSDL.

**send\_request** (send\_function=None)

Sends the assembled request on the child object. @type send\_function: function reference @keyword send\_function: A function reference (passed without the parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

## 3.9 Package Movement Service

DEPRECATED service that was used to validate postal codes and calculate shipment arrival times. Replaced with two distinct services: Validation Availability And Commitment Service, and Country Service.

**class** `fedex.services.country_service.FedexValidatePostalRequest` (*config\_obj*, \**args*,  
\*\**kwargs*)

Bases: `fedex.base_service.FedexBaseService`

This class allows you validate an address. [https://www.fedex.com/us/developer/WebHelp/ws/2015/html/WebServicesHelp/WSDVG/47\\_Country\\_Service.htm](https://www.fedex.com/us/developer/WebHelp/ws/2015/html/WebServicesHelp/WSDVG/47_Country_Service.htm)

@type config\_obj: L{FedexConfig} @param config\_obj: A valid FedexConfig object.

**create\_wsdl\_object\_of\_type** (*type\_name*)

Creates and returns a WSDL object of the specified type. :param type\_name: specifies the object's type name from WSDL.

**send\_request** (*send\_function=None*)

Sends the assembled request on the child object. @type send\_function: function reference @keyword send\_function: A function reference (passed without the parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

**Address = None**

@ivar: Holds Address WSDL objects.

**CarrierCode = None**

@ivar: Carrier Code Default to Fedex (FDXE), or can bbe FDXG.

**CheckForMismatch = None**

@ivar: Holds the CheckForMismatch boolean objects.

**RoutingCode = None**

@ivar: Routing Code Default to FDSD.

**ShipDateTime = None**

@ivar: Holds the ShipDateTime date time objects.



---

## Format Conversion Tools

---

Additional utility functions for converting suds xml object into other formats.

### 4.1 Basic Suds Object to Dictionary

`fedex.tools.conversion.basic_object_to_dict(obj)`

Converts suds object to dict very quickly. Does not serialize date time or normalize key case. :param obj: suds object :return: dict object

### 4.2 Advanced Suds Object to Dictionary

`fedex.tools.conversion.subject_to_dict(obj, key_to_lower=False, json_serialize=False)`

Converts a suds object to a dict. Includes advanced features. :param json\_serialize: If set, changes date and time types to iso string. :param key\_to\_lower: If set, changes index key name to lower case. :param obj: suds object :return: dict object

### 4.3 Suds Object to JSON

`fedex.tools.conversion.subject_to_json(obj, key_to_lower=False)`

Converts a suds object to a JSON string. :param obj: suds object :param key\_to\_lower: If set, changes index key name to lower case. :return: json object





---

## Config Object

---

A standardized configuration container object used by python-fedex.

```
class fedex.config.FedexConfig(key, password, account_number=None, meter_number=None,
                                freight_account_number=None, integrator_id=None,
                                wsdl_path=None, express_region_code=None,
                                use_test_server=False)
```

Bases: `object`

Base configuration class that is used for the different Fedex SOAP calls. These are generally passed to the Fedex request classes as arguments. You may instantiate a `L{FedexConfig}` object with the minimal `C{key}` and `C{password}` arguments and set the instance variables documented below at a later time if you must.

@type key: L{str} @param key: Developer test key. @type password: L{str} @param password: The Fedex-generated password for your Web Systems

account. This is generally emailed to you after registration.

@type account\_number: L{str} @keyword account\_number: The account number sent to you by Fedex after registering for Web Services.

@type meter\_number: L{str} @keyword meter\_number: The meter number sent to you by Fedex after registering for Web Services.

@type freight\_account\_number: L{str} @keyword freight\_account\_number: The freight account number sent to you

by Fedex after registering for Web Services.

@type integrator\_id: L{str} @keyword integrator\_id: The integrator string sent to you by Fedex after registering for Web Services.

@type wsdl\_path: L{str} @keyword wsdl\_path: In the event that you want to override the path to your WSDL directory, do so with this argument.

@type use\_test\_server: L{bool} @keyword use\_test\_server: When this is True, test server WSDLs are used

instead of the production server. You will also need to make sure that your L{FedexConfig} object has a production account number, meter number, authentication key, and password.

**account\_number = None**

@ivar: Web Services account number.

**express\_region\_code = None**

@ivar: Web services ExpressRegionCode

**freight\_account\_number = None**

@ivar: Web Services freight accountnumber.

**integrator\_id = None**

@ivar: Web services integrator ID.

**key = None**

@ivar: Developer test key.

**meter\_number = None**

@ivar: Web services meter number.

**password = None**

@ivar: Fedex Web Services password.

**use\_test\_server = None**

@ivar: When True, point to the test server.

---

## Base Service Class

---

```
class fedex.base_service.FedexBaseService (config_obj, wsdl_name, *args, **kwargs)
```

Bases: `object`

This class is the master class for all Fedex request objects. It gets all of the common SOAP objects created via suds and populates them with values from a `L{FedexConfig}` object, along with keyword arguments via `L{__init__}`.

**@note: This object should never be used directly, use one of the included** sub-classes.

This constructor should only be called by children of the class. As is such, only the optional keyword arguments caught by `C{**kwargs}` will be documented.

**@type** customer\_transaction\_id: `L{str}` **@keyword** customer\_transaction\_id: A user-specified identifier to differentiate this transaction from others. This value will be returned with the response from Fedex.

**create\_wsdl\_object\_of\_type** (*type\_name*)

Creates and returns a WSDL object of the specified type. :param type\_name: specifies the object's type name from WSDL.

**send\_request** (*send\_function=None*)

Sends the assembled request on the child object. **@type** send\_function: function reference **@keyword** send\_function: A function reference (passed without the

parenthesis) to a function that will send the request. This allows for overriding the default function in cases such as validation requests.

**ClientDetail = None**

**@ivar:** WSDL object that holds client account details.

**TransactionDetail = None**

**@ivar:** Holds customer-specified transaction IDs.

**VersionId = None**

**@ivar:** Holds details on the version numbers of the WSDL.

**WebAuthenticationDetail = None**

**@ivar:** WSDL object that holds authentication info.

**config\_obj = None**

@ivar: The FedExConfig object to pull auth info from.

**logger = None**

@ivar: Python logger instance with name 'fedex'.

**response = None**

@ivar: The response from Fedex. You will want to pick what you want out here here. This object does have a `__str__()` method, you'll want to print or log it to see what possible values you can pull.

python-fedex implements various custom exceptions.

### 7.1 Fedex Base Service Exception

```
class fedex.base_service.FedexBaseServiceException (error_code, value)  
    Bases: exceptions.Exception
```

Exception: Serves as the base exception that other service-related exception objects are sub-classed from.

### 7.2 Fedex Failure

```
class fedex.base_service.FedexFailure (error_code, value)  
    Bases: fedex.base_service.FedexBaseServiceException
```

Exception: The request could not be handled at this time. This is generally a server problem.

### 7.3 Fedex Error

```
class fedex.base_service.FedexError (error_code, value)  
    Bases: fedex.base_service.FedexBaseServiceException
```

Exception: These are generally problems with the client-provided data.



---

### General Suds Plugin

---

python-fedex uses the following suds plugin for suds xml requests.

```
class fedex.base_service.GeneralSudsPlugin (**kwargs)
```

```
    Bases: suds.plugin.MessagePlugin
```

General Suds Plugin: Adds logging request and response functionality and prunes empty WSDL objects before sending.

Initializes the request and response loggers.

```
marshalled (context)
```

Removes the WSDL objects that do not have a value before sending.

```
parsed (context)
```

Suds has sax parsed the received reply. Provides the plugin with the opportunity to inspect/modify the sax parsed DOM tree for the reply before it is unmarshalled. @param context: The reply context.

The I{reply} is DOM tree.

@type context: L{MessageContext}

```
received (context)
```

Logs the received response.

```
sending (context)
```

Logs the sent request.

```
unmarshalled (context)
```

Suds has unmarshalled the received reply. Provides the plugin with the opportunity to inspect/modify the unmarshalled reply object before it is returned. @param context: The reply context.

The I{reply} is unmarshalled suds object.

@type context: L{MessageContext}





#### 9.1 2.4.0

- Pickup Service using v11 WSDL (hornedbull)
- Added documentation and unit tests for Pickup Service. (radzhome)
- Update package data to include tools (noodlebreak)

#### 9.2 2.3.0

- Added Location Service using v3 WSDL. (radzhome)
- Added examples and unit tests for Location Service. (radzhome)
- Updated certification process scripts to work with latest WSDLs. (radzhome)
- Added warning logging for requests that come back with warning notes. (radzhome)
- Added PyPI, Travis, requires.io integration/badges. (radzhome)
- Organization change from gtaylor to python-fedex-devs. (gtaylor)
- Added deprecation message for movement service. (radzhome)
- Added conversion tools to convert suds xml object into python dict. (radzhome)
- Redirect logging to stdout for examples and tests when not ran via nose. (radzhome)

#### 9.3 2.2.0

- Added Country Service / Postal Code Validation service. (radzhome)
- Added CountryService\_v4.wsdl for Postal Code Validation. (radzhome)

- Added unit tests and examples for Country service. (radzhome)
- Added Signature Option to ship example. (radzhome)
- Fix base service logging request and response. (radzhome)

## 9.4 2.1.0

- Added Validation, Availability and Commitment (AVC) service. (radzhome)
- Added [Validation]AvailabilityAndCommitmentService\_v4.wsdl for AVC service. (radzhome)
- Added examples and unit tests for AVC service.
- Refactored examples and documentation. (radzhome)
- A quick PEP8 pass using pycharm on most of the codebase (radzhome)
- Add travis yml (radzhome)

## 9.5 2.0.0

- Bump ShipService WSDL to v17 for create and delete shipment. (radzhome)
- Bump AddressValidation WSDL to v4. (radzhome)
- Bump RateService WSDL to v18. (radzhome)
- Bump TrackService WSDL to v10. (radzhome)
- General improvements to base class. (radzhome)
- Refactoring and updates to examples. (radzhome)
- Added test classes. (radzhome)
- Remove old and unused WSDLs. (radzhome)
- Change dependency to suds-jurko to include python 3 support. (radzhome)

## 9.6 1.1.1

- Made RateService\_v16.wsdl point at ws.fedex.com instead of wsbeta.fedex.com. Fixes issues in production. (ikks)

## 9.7 1.1.0

- A quick PEP8 pass on most of the codebase. Yucky. (gtaylor)
- Changing recommended install method to use pip + PyPi. (radzhome)
- Updated rate\_request and freight\_rate\_request examples for WSDL v16 compatibility. (foxxyz)
- Updated rate service WSDL from v8 to v16. (foxxyz)
- Added a freight rate request example (mwcbrent)

- Bump ShipService WSDL to v13. (mwcbrent)
- Update rate example to show multiple ServiceTypes. (danielatdattrixdotcom)

## 9.8 1.0.14

- Re-licensed under BSD.

## 9.9 1.0.12 and 1.0.13

- Forget that these ever existed.

## 9.10 1.0.11

- Bug fix of a bug fix for regions config. (jcartmell)

## 9.11 1.0.10

- Bug fix regarding regions and when they are sent. (jcartmell)

## 9.12 1.0.9

- Various fixes to RateRequest() FedEx API call. (jcartmell)
- Added this changelog. (gtaylor)

## 9.13 1.0.8

- Fixed some problems with the unit tests. (gtaylor)

## 9.14 1.0.7

- Fixed a bug with international rate request example. (gtaylor)

## 9.15 1.0.6

- Lots of documentation improvements. (gtaylor)
- FedEx RateRequest call implemented. (yahtib)
- FedEx Postal Inquiry call implemented. (yahtib)

## 9.16 1.0.4

- Removal of unnecessary files. (gtaylor)
- Documentation improvements. (gtaylor)

## 9.17 1.0.3

- Wrote some label certification modules that can help with the label certification process. (gtaylor)

### 9.17.1 initial

- 1.0

## CHAPTER 10

---

### Indices and tables

---

- `genindex`
- `search`



## A

account\_number (fedex.config.FedexConfig attribute), 22  
 add\_address() (fedex.services.address\_validation\_service.FedexAddressValidationRequest  
 method), 14  
 add\_package() (fedex.services.rate\_service.FedexRateServiceRequest (fedex.services.country\_service.FedexValidatePostalRequest  
 method), 12 method), 16, 17  
 add\_package() (fedex.services.ship\_service.FedexProcessShipmentRequest (fedex.services.location\_service.FedexSearchLocationRequest  
 method), 11 method), 15  
 Address (fedex.services.country\_service.FedexValidatePostalRequest create\_wsdL\_object\_of\_type()  
 attribute), 16, 17 (fedex.services.pickup\_service.FedexCreatePickupRequest  
 method), 16  
 Address (fedex.services.location\_service.FedexSearchLocationRequest create\_wsdL\_object\_of\_type()  
 attribute), 15 (fedex.services.rate\_service.FedexRateServiceRequest  
 method), 12  
 AddressesToValidate (fedex.services.address\_validation\_service.FedexAddressValidationRequest  
 attribute), 15

## B

basic\_sobject\_to\_dict() (in module create\_wsdL\_object\_of\_type()  
 fedex.tools.conversion), 19 (fedex.services.ship\_service.FedexDeleteShipmentRequest  
 method), 12

## C

CarrierCode (fedex.services.availability\_commitment\_service.FedexAvailabilityCommitmentRequest  
 attribute), 13 create\_wsdL\_object\_of\_type()  
 CarrierCode (fedex.services.country\_service.FedexValidatePostalRequest (fedex.services.track\_service.FedexTrackRequest  
 attribute), 16, 17 method), 14

CheckForMismatch (fedex.services.country\_service.FedexValidatePostalRequest  
 attribute), 16, 17

ClientDetail (fedex.base\_service.FedexBaseService attribute), 23 DeletionControlType (fedex.services.ship\_service.FedexDeleteShipmentRequest  
 attribute), 12

config\_obj (fedex.base\_service.FedexBaseService attribute), 23 Destination (fedex.services.availability\_commitment\_service.FedexAvailabilityCommitmentRequest  
 attribute), 13

Constraints (fedex.services.location\_service.FedexSearchLocationRequest  
 attribute), 15

create\_wsdL\_object\_of\_type() express\_region\_code (fedex.config.FedexConfig attribute), 22  
 (fedex.base\_service.FedexBaseService  
 method), 23

create\_wsdL\_object\_of\_type() **F**  
 (fedex.services.address\_validation\_service.FedexAddressValidationRequest (class in  
 method), 14 fedex.services.address\_validation\_service),  
 14

- FedexAvailabilityCommitmentRequest (class in [FedexAvailabilityCommitmentService](#)), 13
- FedexBaseService (class in [FedexBaseService](#)), 23
- FedexBaseServiceException (class in [FedexBaseService](#)), 25
- FedexConfig (class in [FedexConfig](#)), 21
- FedexCreatePickupRequest (class in [FedexCreatePickupRequest](#)), 16
- FedexDeleteShipmentRequest (class in [FedexDeleteShipmentRequest](#)), 12
- FedexError (class in [FedexError](#)), 25
- FedexFailure (class in [FedexFailure](#)), 25
- FedexProcessShipmentRequest (class in [FedexProcessShipmentRequest](#)), 11
- FedexRateServiceRequest (class in [FedexRateServiceRequest](#)), 12
- FedexSearchLocationRequest (class in [FedexSearchLocationRequest](#)), 15
- FedexTrackRequest (class in [FedexTrackRequest](#)), 14
- FedexValidatePostalRequest (class in [FedexValidatePostalRequest](#)), 15, 16
- freight\_account\_number ([FedexConfig](#) attribute), 22
- ## G
- GeneralSudsPlugin (class in [FedexBaseService](#)), 27
- ## I
- integrator\_id ([FedexConfig](#) attribute), 22
- ## K
- key ([FedexConfig](#) attribute), 22
- ## L
- LocationsSearchCriterion ([FedexSearchLocationRequest](#) attribute), 15
- logger ([FedexBaseService](#) attribute), 24
- ## M
- marshalled() ([FedexBaseService.GeneralSudsPlugin](#) method), 27
- meter\_number ([FedexConfig](#) attribute), 22
- MultipleMatchesAction ([FedexSearchLocationRequest](#) attribute), 15
- ## O
- Origin ([FedexAvailabilityCommitmentService.FedexAvailabilityCommitmentRequest](#) attribute), 13
- Packaging ([FedexAvailabilityCommitmentService.FedexAvailabilityCommitmentRequest](#) attribute), 13
- parsed() ([FedexBaseService.GeneralSudsPlugin](#) method), 27
- password ([FedexConfig](#) attribute), 22
- PhoneNumber ([FedexSearchLocationRequest](#) attribute), 15
- ProcessingOptions ([FedexTrackRequest](#) attribute), 14
- ## R
- received() ([FedexBaseService.GeneralSudsPlugin](#) method), 27
- RequestedShipment ([FedexRateServiceRequest](#) attribute), 13
- RequestedShipment ([FedexProcessShipmentRequest](#) attribute), 12
- response ([FedexBaseService](#) attribute), 24
- RoutingCode ([FedexValidatePostalRequest](#) attribute), 16, 17
- ## S
- SelectionDetails ([FedexTrackRequest](#) attribute), 14
- send\_request() ([FedexBaseService](#) method), 23
- send\_request() ([FedexAddressValidationService.FedexAddressValidationRequest](#) method), 14
- send\_request() ([FedexAvailabilityCommitmentService.FedexAvailabilityCommitmentRequest](#) method), 13
- send\_request() ([FedexValidatePostalRequest](#) method), 16, 17
- send\_request() ([FedexSearchLocationRequest](#) method), 15
- send\_request() ([FedexCreatePickupRequest](#) method), 16
- send\_request() ([FedexRateServiceRequest](#) method), 13
- send\_request() ([FedexDeleteShipmentRequest](#) method), 12
- send\_request() ([FedexProcessShipmentRequest](#) method), 11
- send\_request() ([FedexTrackRequest](#) method), 14
- send\_validation\_request() ([FedexProcessShipmentRequest](#) method), 12
- sending() ([FedexBaseService.GeneralSudsPlugin](#) method), 27
- Service ([FedexAvailabilityCommitmentService.FedexAvailabilityCommitmentRequest](#) attribute), 13



ShipDate (fedex.services.availability\_commitment\_service.FedexAvailabilityCommitmentRequest attribute), [13](#)

ShipDateTime (fedex.services.country\_service.FedexValidatePostalRequest attribute), [16](#), [17](#)

subject\_to\_dict() (in module fedex.tools.conversion), [19](#)

subject\_to\_json() (in module fedex.tools.conversion), [19](#)

SortDetail (fedex.services.location\_service.FedexSearchLocationRequest attribute), [15](#)

## T

TrackingId (fedex.services.ship\_service.FedexDeleteShipmentRequest attribute), [12](#)

TransactionDetail (fedex.base\_service.FedexBaseService attribute), [23](#)

## U

unmarshalled() (fedex.base\_service.GeneralSudsPlugin method), [27](#)

use\_test\_server (fedex.config.FedexConfig attribute), [22](#)

## V

VersionId (fedex.base\_service.FedexBaseService attribute), [23](#)

## W

WebAuthenticationDetail  
(fedex.base\_service.FedexBaseService attribute), [23](#)