Python

# API Client

def call\_api(

self, resource\_path, method, path\_params=None,

query\_params=None, header\_params=None, body=None, post\_params=None,

files=None, response\_type=None, auth\_settings=None,

\_return\_http\_data\_only=None, collection\_formats=None,

\_preload\_content=True, \_request\_timeout=None):

config = self.configuration

# header parameters

header\_params = header\_params or {}

header\_params.update(self.default\_headers)

if self.cookie:

header\_params['Cookie'] = self.cookie

if header\_params:

header\_params = self.sanitize\_for\_serialization(header\_params)

header\_params = dict(self.parameters\_to\_tuples(header\_params,

collection\_formats))

# path parameters

if path\_params:

path\_params = self.sanitize\_for\_serialization(path\_params)

path\_params = self.parameters\_to\_tuples(path\_params,

collection\_formats)

for k, v in path\_params:

# specified safe chars, encode everything

resource\_path = resource\_path.replace(

'{%s}' % k,

quote(str(v), safe=config.safe\_chars\_for\_path\_param)

)

# query parameters

if query\_params:

query\_params = self.sanitize\_for\_serialization(query\_params)

query\_params = self.parameters\_to\_tuples(query\_params,

collection\_formats)

# post parameters

if post\_params or files:

post\_params = self.prepare\_post\_parameters(post\_params, files)

post\_params = self.sanitize\_for\_serialization(post\_params)

post\_params = self.parameters\_to\_tuples(post\_params,

collection\_formats)

# auth setting

self.update\_params\_for\_auth(header\_params, query\_params, auth\_settings)

# body

if body:

body = self.sanitize\_for\_serialization(body)

# request url

url = self.configuration.host + resource\_path

# perform request and return response

response\_data = self.request(

method, url, query\_params=query\_params, headers=header\_params,

post\_params=post\_params, body=body,

\_preload\_content=\_preload\_content,

\_request\_timeout=\_request\_timeout)

self.last\_response = response\_data

return\_data = response\_data

if \_preload\_content:

# deserialize response data

if response\_type:

return\_data = self.deserialize(response\_data, response\_type)

else:

return\_data = None

if \_return\_http\_data\_only:

return (return\_data)

else:

return (return\_data, response\_data.status,

response\_data.getheaders())

# Account API

## /api/v1/Account/GetAccountSettings

def api\_v1\_account\_get\_account\_settings\_get\_with\_http\_info(self, account, password, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_account\_get\_account\_settings\_get" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_account\_get\_account\_settings\_get`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_account\_get\_account\_settings\_get`") # noqa: E501

collection\_formats = {}

path\_params = {}

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Account/GetAccountSettings', 'GET',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Settings', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

# Communication API

## /api/v1/Communication/EmailReceipt

def api\_v1\_communication\_email\_receipt\_post\_with\_http\_info(self, account, password, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_communication\_email\_receipt\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_communication\_email\_receipt\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_communication\_email\_receipt\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Communication/EmailReceipt', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Response', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

# Reporting API

## /api/v1/Reporting/GetBatchesByDate/{batchDate}

def api\_v1\_reporting\_get\_batches\_by\_date\_batch\_date\_get\_with\_http\_info(self, batch\_date, account, password, \*\*kwargs): # noqa: E501

all\_params = ['batch\_date', 'account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_reporting\_get\_batches\_by\_date\_batch\_date\_get" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'batch\_date' is set

if ('batch\_date' not in params or

params['batch\_date'] is None):

raise ValueError("Missing the required parameter `batch\_date` when calling `api\_v1\_reporting\_get\_batches\_by\_date\_batch\_date\_get`") # noqa: E501

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_reporting\_get\_batches\_by\_date\_batch\_date\_get`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_reporting\_get\_batches\_by\_date\_batch\_date\_get`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'batch\_date' in params:

path\_params['batchDate'] = params['batch\_date'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Reporting/GetBatchesByDate/{batchDate}', 'GET',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='list[Batch]', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Reporting/GetCheckTerminals

def api\_v1\_reporting\_get\_check\_terminal\_settings\_by\_sec\_sec\_code\_get\_with\_http\_info(self, sec\_code, account, password, \*\*kwargs): # noqa: E501

all\_params = ['sec\_code', 'account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_reporting\_get\_check\_terminal\_settings\_by\_sec\_sec\_code\_get" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'sec\_code' is set

if ('sec\_code' not in params or

params['sec\_code'] is None):

raise ValueError("Missing the required parameter `sec\_code` when calling `api\_v1\_reporting\_get\_check\_terminal\_settings\_by\_sec\_sec\_code\_get`") # noqa: E501

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_reporting\_get\_check\_terminal\_settings\_by\_sec\_sec\_code\_get`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_reporting\_get\_check\_terminal\_settings\_by\_sec\_sec\_code\_get`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'sec\_code' in params:

path\_params['secCode'] = params['sec\_code'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Reporting/GetCheckTerminalSettingsBySEC/{secCode}', 'GET',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='list[TerminalSettings]', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Reporting/GetCheckTerminalSettingsBySEC/{secCode}

def api\_v1\_reporting\_get\_check\_terminal\_settings\_by\_sec\_sec\_code\_get\_with\_http\_info(self, sec\_code, account, password, \*\*kwargs): # noqa: E501

all\_params = ['sec\_code', 'account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_reporting\_get\_check\_terminal\_settings\_by\_sec\_sec\_code\_get" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'sec\_code' is set

if ('sec\_code' not in params or

params['sec\_code'] is None):

raise ValueError("Missing the required parameter `sec\_code` when calling `api\_v1\_reporting\_get\_check\_terminal\_settings\_by\_sec\_sec\_code\_get`") # noqa: E501

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_reporting\_get\_check\_terminal\_settings\_by\_sec\_sec\_code\_get`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_reporting\_get\_check\_terminal\_settings\_by\_sec\_sec\_code\_get`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'sec\_code' in params:

path\_params['secCode'] = params['sec\_code'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Reporting/GetCheckTerminalSettingsBySEC/{secCode}', 'GET',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='list[TerminalSettings]', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Reporting/GetCheckTerminalSettings/{terminalID}

def api\_v1\_reporting\_get\_check\_terminal\_settings\_terminal\_id\_get\_with\_http\_info(self, terminal\_id, account, password, \*\*kwargs): # noqa: E501

all\_params = ['terminal\_id', 'account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_reporting\_get\_check\_terminal\_settings\_terminal\_id\_get" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'terminal\_id' is set

if ('terminal\_id' not in params or

params['terminal\_id'] is None):

raise ValueError("Missing the required parameter `terminal\_id` when calling `api\_v1\_reporting\_get\_check\_terminal\_settings\_terminal\_id\_get`") # noqa: E501

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_reporting\_get\_check\_terminal\_settings\_terminal\_id\_get`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_reporting\_get\_check\_terminal\_settings\_terminal\_id\_get`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'terminal\_id' in params:

path\_params['terminalID'] = params['terminal\_id'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Reporting/GetCheckTerminalSettings/{terminalID}', 'GET',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='list[TerminalSettings]', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Reporting/GetTransactionsByBatch/{batch}

def api\_v1\_reporting\_get\_transactions\_by\_batch\_batch\_get\_with\_http\_info(self, batch, account, password, \*\*kwargs): # noqa: E501

all\_params = ['batch', 'account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_reporting\_get\_transactions\_by\_batch\_batch\_get" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'batch' is set

if ('batch' not in params or

params['batch'] is None):

raise ValueError("Missing the required parameter `batch` when calling `api\_v1\_reporting\_get\_transactions\_by\_batch\_batch\_get`") # noqa: E501

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_reporting\_get\_transactions\_by\_batch\_batch\_get`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_reporting\_get\_transactions\_by\_batch\_batch\_get`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'batch' in params:

path\_params['batch'] = params['batch'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Reporting/GetTransactionsByBatch/{batch}', 'GET',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='list[QueryTransaction]', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Reporting/GetTransactionsByDate/{transactionDate}

def api\_v1\_reporting\_get\_transactions\_by\_date\_transaction\_date\_get\_with\_http\_info(self, transaction\_date, account, password, \*\*kwargs): # noqa: E501

all\_params = ['transaction\_date', 'account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_reporting\_get\_transactions\_by\_date\_transaction\_date\_get" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'transaction\_date' is set

if ('transaction\_date' not in params or

params['transaction\_date'] is None):

raise ValueError("Missing the required parameter `transaction\_date` when calling `api\_v1\_reporting\_get\_transactions\_by\_date\_transaction\_date\_get`") # noqa: E501

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_reporting\_get\_transactions\_by\_date\_transaction\_date\_get`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_reporting\_get\_transactions\_by\_date\_transaction\_date\_get`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'transaction\_date' in params:

path\_params['transactionDate'] = params['transaction\_date'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Reporting/GetTransactionsByDate/{transactionDate}', 'GET',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='list[QueryTransaction]', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

# Vault API

## /api/v1/Vault/GetVaultRecord/{reference}

def api\_v1\_vault\_get\_vault\_record\_reference\_get\_with\_http\_info(self, reference, account, password, \*\*kwargs): # noqa: E501

all\_params = ['reference', 'account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_vault\_get\_vault\_record\_reference\_get" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'reference' is set

if ('reference' not in params or

params['reference'] is None):

raise ValueError("Missing the required parameter `reference` when calling `api\_v1\_vault\_get\_vault\_record\_reference\_get`") # noqa: E501

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_vault\_get\_vault\_record\_reference\_get`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_vault\_get\_vault\_record\_reference\_get`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'reference' in params:

path\_params['reference'] = params['reference'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Vault/GetVaultRecord/{reference}', 'GET',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Record', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Vault/SaveVault

def api\_v1\_vault\_save\_vault\_card\_vault\_id\_post\_with\_http\_info(self, account, password, vault\_id, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'vault\_id', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_vault\_save\_vault\_card\_vault\_id\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_vault\_save\_vault\_card\_vault\_id\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_vault\_save\_vault\_card\_vault\_id\_post`") # noqa: E501

# verify the required parameter 'vault\_id' is set

if ('vault\_id' not in params or

params['vault\_id'] is None):

raise ValueError("Missing the required parameter `vault\_id` when calling `api\_v1\_vault\_save\_vault\_card\_vault\_id\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'vault\_id' in params:

path\_params['vaultID'] = params['vault\_id'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Vault/SaveVaultCard/{vaultID}', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Response', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Vault/SaveVaultCard/{vaultID}

def api\_v1\_vault\_save\_vault\_card\_vault\_id\_post\_with\_http\_info(self, account, password, vault\_id, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'vault\_id', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_vault\_save\_vault\_card\_vault\_id\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_vault\_save\_vault\_card\_vault\_id\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_vault\_save\_vault\_card\_vault\_id\_post`") # noqa: E501

# verify the required parameter 'vault\_id' is set

if ('vault\_id' not in params or

params['vault\_id'] is None):

raise ValueError("Missing the required parameter `vault\_id` when calling `api\_v1\_vault\_save\_vault\_card\_vault\_id\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'vault\_id' in params:

path\_params['vaultID'] = params['vault\_id'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Vault/SaveVaultCard/{vaultID}', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Response', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Vault/SearchVault

def api\_v1\_vault\_search\_vault\_post\_with\_http\_info(self, account, password, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_vault\_search\_vault\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_vault\_search\_vault\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_vault\_search\_vault\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Vault/SearchVault', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='list[Record]', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Vault/DeleteVaultCardByID/{vaultCardID}

def api\_v1\_vault\_delete\_vault\_card\_by\_id\_vault\_card\_id\_post\_with\_http\_info(self, vault\_card\_id, account, password, \*\*kwargs): # noqa: E501

all\_params = ['vault\_card\_id', 'account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_vault\_delete\_vault\_card\_by\_id\_vault\_card\_id\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'vault\_card\_id' is set

if ('vault\_card\_id' not in params or

params['vault\_card\_id'] is None):

raise ValueError("Missing the required parameter `vault\_card\_id` when calling `api\_v1\_vault\_delete\_vault\_card\_by\_id\_vault\_card\_id\_post`") # noqa: E501

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_vault\_delete\_vault\_card\_by\_id\_vault\_card\_id\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_vault\_delete\_vault\_card\_by\_id\_vault\_card\_id\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'vault\_card\_id' in params:

path\_params['vaultCardID'] = params['vault\_card\_id'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Vault/DeleteVaultCardByID/{vaultCardID}', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Response', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Vault/DeleteVaultCheckByID/{vaultCheckID}

def api\_v1\_vault\_delete\_vault\_check\_by\_id\_vault\_check\_id\_post\_with\_http\_info(self, vault\_check\_id, account, password, \*\*kwargs): # noqa: E501

all\_params = ['vault\_check\_id', 'account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_vault\_delete\_vault\_check\_by\_id\_vault\_check\_id\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'vault\_check\_id' is set

if ('vault\_check\_id' not in params or

params['vault\_check\_id'] is None):

raise ValueError("Missing the required parameter `vault\_check\_id` when calling `api\_v1\_vault\_delete\_vault\_check\_by\_id\_vault\_check\_id\_post`") # noqa: E501

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_vault\_delete\_vault\_check\_by\_id\_vault\_check\_id\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_vault\_delete\_vault\_check\_by\_id\_vault\_check\_id\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'vault\_check\_id' in params:

path\_params['vaultCheckID'] = params['vault\_check\_id'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Vault/DeleteVaultCheckByID/{vaultCheckID}', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Response', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Vault/SubmitCheckWithVaultCheckID/{vaultCheckID}

def api\_v1\_vault\_submit\_check\_with\_vault\_check\_id\_vault\_check\_id\_post\_with\_http\_info(self, account, password, vault\_check\_id, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'vault\_check\_id', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_vault\_submit\_check\_with\_vault\_check\_id\_vault\_check\_id\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_vault\_submit\_check\_with\_vault\_check\_id\_vault\_check\_id\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_vault\_submit\_check\_with\_vault\_check\_id\_vault\_check\_id\_post`") # noqa: E501

# verify the required parameter 'vault\_check\_id' is set

if ('vault\_check\_id' not in params or

params['vault\_check\_id'] is None):

raise ValueError("Missing the required parameter `vault\_check\_id` when calling `api\_v1\_vault\_submit\_check\_with\_vault\_check\_id\_vault\_check\_id\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'vault\_check\_id' in params:

path\_params['vaultCheckID'] = params['vault\_check\_id'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Vault/SubmitCheckWithVaultCheckID/{vaultCheckID}', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='CheckTransaction', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/Vault/SubmitWithVaultCardID/{vaultCardID}

def api\_v1\_vault\_submit\_with\_vault\_card\_id\_vault\_card\_id\_post\_with\_http\_info(self, account, password, vault\_card\_id, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'vault\_card\_id', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_vault\_submit\_with\_vault\_card\_id\_vault\_card\_id\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_vault\_submit\_with\_vault\_card\_id\_vault\_card\_id\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_vault\_submit\_with\_vault\_card\_id\_vault\_card\_id\_post`") # noqa: E501

# verify the required parameter 'vault\_card\_id' is set

if ('vault\_card\_id' not in params or

params['vault\_card\_id'] is None):

raise ValueError("Missing the required parameter `vault\_card\_id` when calling `api\_v1\_vault\_submit\_with\_vault\_card\_id\_vault\_card\_id\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'vault\_card\_id' in params:

path\_params['vaultCardID'] = params['vault\_card\_id'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/Vault/SubmitWithVaultCardID/{vaultCardID}', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Transaction', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

# VirtualTerminal API

## /api/v1/VirtualTerminal/Submit

def api\_v1\_virtual\_terminal\_submit\_check\_post\_with\_http\_info(self, account, password, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_virtual\_terminal\_submit\_check\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_virtual\_terminal\_submit\_check\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_virtual\_terminal\_submit\_check\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/VirtualTerminal/SubmitCheck', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='CheckTransaction', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/VirtualTerminal/SubmitCheck

def api\_v1\_virtual\_terminal\_submit\_check\_post\_with\_http\_info(self, account, password, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_virtual\_terminal\_submit\_check\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_virtual\_terminal\_submit\_check\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_virtual\_terminal\_submit\_check\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/VirtualTerminal/SubmitCheck', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='CheckTransaction', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/VirtualTerminal/Mark/{transactionID}

def api\_v1\_virtual\_terminal\_mark\_transaction\_id\_post\_with\_http\_info(self, transaction\_id, account, password, \*\*kwargs): # noqa: E501

all\_params = ['transaction\_id', 'account', 'password'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_virtual\_terminal\_mark\_transaction\_id\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'transaction\_id' is set

if ('transaction\_id' not in params or

params['transaction\_id'] is None):

raise ValueError("Missing the required parameter `transaction\_id` when calling `api\_v1\_virtual\_terminal\_mark\_transaction\_id\_post`") # noqa: E501

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_virtual\_terminal\_mark\_transaction\_id\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_virtual\_terminal\_mark\_transaction\_id\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'transaction\_id' in params:

path\_params['transactionID'] = params['transaction\_id'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/VirtualTerminal/Mark/{transactionID}', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Response', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/VirtualTerminal/MarkTransactions

def api\_v1\_virtual\_terminal\_mark\_transactions\_post\_with\_http\_info(self, account, password, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_virtual\_terminal\_mark\_transactions\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_virtual\_terminal\_mark\_transactions\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_virtual\_terminal\_mark\_transactions\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/VirtualTerminal/MarkTransactions', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Response', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/VirtualTerminal/Query

def api\_v1\_virtual\_terminal\_query\_post\_with\_http\_info(self, account, password, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_virtual\_terminal\_query\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_virtual\_terminal\_query\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_virtual\_terminal\_query\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/VirtualTerminal/Query', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='QueryTransaction', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)

## /api/v1/VirtualTerminal/UpdateTransactionInfo/{transactionID}

def api\_v1\_virtual\_terminal\_update\_transaction\_info\_transaction\_id\_post\_with\_http\_info(self, account, password, transaction\_id, \*\*kwargs): # noqa: E501

all\_params = ['account', 'password', 'transaction\_id', 'body'] # noqa: E501

all\_params.append('async\_req')

all\_params.append('\_return\_http\_data\_only')

all\_params.append('\_preload\_content')

all\_params.append('\_request\_timeout')

params = locals()

for key, val in six.iteritems(params['kwargs']):

if key not in all\_params:

raise TypeError(

"Got an unexpected keyword argument '%s'"

" to method api\_v1\_virtual\_terminal\_update\_transaction\_info\_transaction\_id\_post" % key

)

params[key] = val

del params['kwargs']

# verify the required parameter 'account' is set

if ('account' not in params or

params['account'] is None):

raise ValueError("Missing the required parameter `account` when calling `api\_v1\_virtual\_terminal\_update\_transaction\_info\_transaction\_id\_post`") # noqa: E501

# verify the required parameter 'password' is set

if ('password' not in params or

params['password'] is None):

raise ValueError("Missing the required parameter `password` when calling `api\_v1\_virtual\_terminal\_update\_transaction\_info\_transaction\_id\_post`") # noqa: E501

# verify the required parameter 'transaction\_id' is set

if ('transaction\_id' not in params or

params['transaction\_id'] is None):

raise ValueError("Missing the required parameter `transaction\_id` when calling `api\_v1\_virtual\_terminal\_update\_transaction\_info\_transaction\_id\_post`") # noqa: E501

collection\_formats = {}

path\_params = {}

if 'transaction\_id' in params:

path\_params['transactionID'] = params['transaction\_id'] # noqa: E501

query\_params = []

header\_params = {}

if 'account' in params:

header\_params['Account'] = params['account'] # noqa: E501

if 'password' in params:

header\_params['Password'] = params['password'] # noqa: E501

form\_params = []

local\_var\_files = {}

body\_params = None

if 'body' in params:

body\_params = params['body']

# HTTP header `Accept`

header\_params['Accept'] = self.api\_client.select\_header\_accept(

['text/plain', 'application/json', 'text/json']) # noqa: E501

# HTTP header `Content-Type`

header\_params['Content-Type'] = self.api\_client.select\_header\_content\_type( # noqa: E501

['application/json', 'text/json', 'application/\*+json']) # noqa: E501

# Authentication setting

auth\_settings = [] # noqa: E501

return self.api\_client.call\_api(

'/api/v1/VirtualTerminal/UpdateTransactionInfo/{transactionID}', 'POST',

path\_params,

query\_params,

header\_params,

body=body\_params,

post\_params=form\_params,

files=local\_var\_files,

response\_type='Response', # noqa: E501

auth\_settings=auth\_settings,

async\_req=params.get('async\_req'),

\_return\_http\_data\_only=params.get('\_return\_http\_data\_only'),

\_preload\_content=params.get('\_preload\_content', True),

\_request\_timeout=params.get('\_request\_timeout'),

collection\_formats=collection\_formats)