Python Models

# Address

# coding: utf-8

class Address(object):

swagger\_types = {

'address\_line1': 'str',

'address\_line2': 'str',

'city': 'str',

'state': 'str',

'zip\_code': 'str'

}

attribute\_map = {

'address\_line1': 'AddressLine1',

'address\_line2': 'AddressLine2',

'city': 'City',

'state': 'State',

'zip\_code': 'ZipCode'

}

def \_\_init\_\_(self, address\_line1=None, address\_line2=None, city=None, state=None, zip\_code=None): # noqa: E501

"""Address - a model defined in Swagger""" # noqa: E501

self.\_address\_line1 = None

self.\_address\_line2 = None

self.\_city = None

self.\_state = None

self.\_zip\_code = None

self.discriminator = None

if address\_line1 is not None:

self.address\_line1 = address\_line1

if address\_line2 is not None:

self.address\_line2 = address\_line2

if city is not None:

self.city = city

if state is not None:

self.state = state

if zip\_code is not None:

self.zip\_code = zip\_code

@property

def address\_line1(self):

return self.\_address\_line1

@address\_line1.setter

def address\_line1(self, address\_line1):

self.\_address\_line1 = address\_line1

@property

def address\_line2(self):

return self.\_address\_line2

@address\_line2.setter

def address\_line2(self, address\_line2):

self.\_address\_line2 = address\_line2

@property

def city(self):

return self.\_city

@city.setter

def city(self, city):

self.\_city = city

@property

def state(self):

return self.\_state

@state.setter

def state(self, state):

self.\_state = state

@property

def zip\_code(self):

return self.\_zip\_code

@zip\_code.setter

def zip\_code(self, zip\_code):

self.\_zip\_code = zip\_code

# Batch

# coding: utf-8

class Batch(object):

swagger\_types = {

'batch\_number': 'str',

'started': 'datetime',

'completed': 'datetime',

'sales\_count': 'str',

'sales\_sum': 'float',

'return\_count': 'str',

'return\_sum': 'float',

'batch\_error': 'bool'

}

attribute\_map = {

'batch\_number': 'BatchNumber',

'started': 'Started',

'completed': 'Completed',

'sales\_count': 'SalesCount',

'sales\_sum': 'SalesSum',

'return\_count': 'ReturnCount',

'return\_sum': 'ReturnSum',

'batch\_error': 'BatchError'

}

def \_\_init\_\_(self, batch\_number=None, started=None, completed=None, sales\_count=None, sales\_sum=None, return\_count=None, return\_sum=None, batch\_error=None): # noqa: E501

"""Batch - a model defined in Swagger""" # noqa: E501

self.\_batch\_number = None

self.\_started = None

self.\_completed = None

self.\_sales\_count = None

self.\_sales\_sum = None

self.\_return\_count = None

self.\_return\_sum = None

self.\_batch\_error = None

self.discriminator = None

if batch\_number is not None:

self.batch\_number = batch\_number

if started is not None:

self.started = started

if completed is not None:

self.completed = completed

if sales\_count is not None:

self.sales\_count = sales\_count

if sales\_sum is not None:

self.sales\_sum = sales\_sum

if return\_count is not None:

self.return\_count = return\_count

if return\_sum is not None:

self.return\_sum = return\_sum

if batch\_error is not None:

self.batch\_error = batch\_error

@property

def batch\_number(self):

return self.\_batch\_number

@batch\_number.setter

def batch\_number(self, batch\_number):

self.\_batch\_number = batch\_number

@property

def started(self):

return self.\_started

@started.setter

def started(self, started):

self.\_started = started

@property

def completed(self):

return self.\_completed

@completed.setter

def completed(self, completed):

self.\_completed = completed

@property

def sales\_count(self):

return self.\_sales\_count

@sales\_count.setter

def sales\_count(self, sales\_count):

self.\_sales\_count = sales\_count

@property

def sales\_sum(self):

return self.\_sales\_sum

@sales\_sum.setter

def sales\_sum(self, sales\_sum):

self.\_sales\_sum = sales\_sum

@property

def return\_count(self):

return self.\_return\_count

@return\_count.setter

def return\_count(self, return\_count):

self.\_return\_count = return\_count

@property

def return\_sum(self):

return self.\_return\_sum

@return\_sum.setter

def return\_sum(self, return\_sum):

self.\_return\_sum = return\_sum

@property

def batch\_error(self):

return self.\_batch\_error

@batch\_error.setter

def batch\_error(self, batch\_error):

self.\_batch\_error = batch\_error

# Business

# coding: utf-8

class Business(object):

swagger\_types = {

'address': 'Address',

'name': 'str',

'phone': 'str',

'fax': 'str',

'web\_site': 'str'

}

attribute\_map = {

'address': 'Address',

'name': 'Name',

'phone': 'Phone',

'fax': 'Fax',

'web\_site': 'WebSite'

}

def \_\_init\_\_(self, address=None, name=None, phone=None, fax=None, web\_site=None): # noqa: E501

"""Business - a model defined in Swagger""" # noqa: E501

self.\_address = None

self.\_name = None

self.\_phone = None

self.\_fax = None

self.\_web\_site = None

self.discriminator = None

if address is not None:

self.address = address

if name is not None:

self.name = name

if phone is not None:

self.phone = phone

if fax is not None:

self.fax = fax

if web\_site is not None:

self.web\_site = web\_site

@property

def address(self):

return self.\_address

@address.setter

def address(self, address):

self.\_address = address

@property

def name(self):

return self.\_name

@name.setter

def name(self, name):

self.\_name = name

@property

def phone(self):

return self.\_phone

@phone.setter

def phone(self, phone):

self.\_phone = phone

@property

def fax(self):

return self.\_fax

@fax.setter

def fax(self, fax):

self.\_fax = fax

@property

def web\_site(self):

return self.\_web\_site

@web\_site.setter

def web\_site(self, web\_site):

self.\_web\_site = web\_site

# CheckAccountType

# coding: utf-8

class CheckAccountType(object):

UNKNOWN = "Unknown"

CHECKING = "Checking"

SAVINGS = "Savings"

swagger\_types = {

}

attribute\_map = {

}

def \_\_init\_\_(self): # noqa: E501

"""CheckAccountType - a model defined in Swagger""" # noqa: E501

self.discriminator = None

# CheckTransaction

# coding: utf-8

class CheckTransaction(object):

swagger\_types = {

'terminal\_id': 'str',

'account\_number': 'str',

'account\_type': 'CheckAccountType',

'check\_number': 'str',

'routing\_number': 'str',

'micr\_data': 'str',

'first\_name': 'str',

'last\_name': 'str',

'company\_name': 'str',

'address1': 'str',

'address2': 'str',

'city': 'str',

'state': 'str',

'zip': 'str',

'phone\_number': 'str',

'dl\_number': 'str',

'dl\_state': 'str',

'courtesy\_card\_id': 'str',

'ssn4': 'str',

'dob\_year': 'str',

'check\_amount': 'str',

'tran\_type': 'CheckTransactionType',

'proc\_id': 'str',

'ip\_address': 'str',

'customer\_number': 'str',

'po\_number': 'str',

'invoice\_number': 'str',

'tran\_note': 'str',

'email\_address': 'str',

'sec\_code': 'str',

'proc\_resp': 'str',

'proc\_code': 'str',

'proc\_mess': 'str'

}

attribute\_map = {

'terminal\_id': 'Terminal\_ID',

'account\_number': 'Account\_Number',

'account\_type': 'Account\_Type',

'check\_number': 'Check\_Number',

'routing\_number': 'Routing\_Number',

'micr\_data': 'MICR\_Data',

'first\_name': 'First\_Name',

'last\_name': 'Last\_Name',

'company\_name': 'Company\_Name',

'address1': 'Address1',

'address2': 'Address2',

'city': 'City',

'state': 'State',

'zip': 'Zip',

'phone\_number': 'Phone\_Number',

'dl\_number': 'DL\_Number',

'dl\_state': 'DL\_State',

'courtesy\_card\_id': 'Courtesy\_Card\_ID',

'ssn4': 'SSN4',

'dob\_year': 'DOB\_Year',

'check\_amount': 'Check\_Amount',

'tran\_type': 'Tran\_Type',

'proc\_id': 'Proc\_ID',

'ip\_address': 'IP\_Address',

'customer\_number': 'CustomerNumber',

'po\_number': 'PONumber',

'invoice\_number': 'InvoiceNumber',

'tran\_note': 'TranNote',

'email\_address': 'EmailAddress',

'sec\_code': 'SEC\_Code',

'proc\_resp': 'Proc\_Resp',

'proc\_code': 'Proc\_Code',

'proc\_mess': 'Proc\_Mess'

}

def \_\_init\_\_(self, terminal\_id=None, account\_number=None, account\_type=None, check\_number=None, routing\_number=None, micr\_data=None, first\_name=None, last\_name=None, company\_name=None, address1=None, address2=None, city=None, state=None, zip=None, phone\_number=None, dl\_number=None, dl\_state=None, courtesy\_card\_id=None, ssn4=None, dob\_year=None, check\_amount=None, tran\_type=None, proc\_id=None, ip\_address=None, customer\_number=None, po\_number=None, invoice\_number=None, tran\_note=None, email\_address=None, sec\_code=None, proc\_resp=None, proc\_code=None, proc\_mess=None): # noqa: E501

"""CheckTransaction - a model defined in Swagger""" # noqa: E501

self.\_terminal\_id = None

self.\_account\_number = None

self.\_account\_type = None

self.\_check\_number = None

self.\_routing\_number = None

self.\_micr\_data = None

self.\_first\_name = None

self.\_last\_name = None

self.\_company\_name = None

self.\_address1 = None

self.\_address2 = None

self.\_city = None

self.\_state = None

self.\_zip = None

self.\_phone\_number = None

self.\_dl\_number = None

self.\_dl\_state = None

self.\_courtesy\_card\_id = None

self.\_ssn4 = None

self.\_dob\_year = None

self.\_check\_amount = None

self.\_tran\_type = None

self.\_proc\_id = None

self.\_ip\_address = None

self.\_customer\_number = None

self.\_po\_number = None

self.\_invoice\_number = None

self.\_tran\_note = None

self.\_email\_address = None

self.\_sec\_code = None

self.\_proc\_resp = None

self.\_proc\_code = None

self.\_proc\_mess = None

self.discriminator = None

if terminal\_id is not None:

self.terminal\_id = terminal\_id

if account\_number is not None:

self.account\_number = account\_number

if account\_type is not None:

self.account\_type = account\_type

if check\_number is not None:

self.check\_number = check\_number

if routing\_number is not None:

self.routing\_number = routing\_number

if micr\_data is not None:

self.micr\_data = micr\_data

if first\_name is not None:

self.first\_name = first\_name

if last\_name is not None:

self.last\_name = last\_name

if company\_name is not None:

self.company\_name = company\_name

if address1 is not None:

self.address1 = address1

if address2 is not None:

self.address2 = address2

if city is not None:

self.city = city

if state is not None:

self.state = state

if zip is not None:

self.zip = zip

if phone\_number is not None:

self.phone\_number = phone\_number

if dl\_number is not None:

self.dl\_number = dl\_number

if dl\_state is not None:

self.dl\_state = dl\_state

if courtesy\_card\_id is not None:

self.courtesy\_card\_id = courtesy\_card\_id

if ssn4 is not None:

self.ssn4 = ssn4

if dob\_year is not None:

self.dob\_year = dob\_year

if check\_amount is not None:

self.check\_amount = check\_amount

if tran\_type is not None:

self.tran\_type = tran\_type

if proc\_id is not None:

self.proc\_id = proc\_id

if ip\_address is not None:

self.ip\_address = ip\_address

if customer\_number is not None:

self.customer\_number = customer\_number

if po\_number is not None:

self.po\_number = po\_number

if invoice\_number is not None:

self.invoice\_number = invoice\_number

if tran\_note is not None:

self.tran\_note = tran\_note

if email\_address is not None:

self.email\_address = email\_address

if sec\_code is not None:

self.sec\_code = sec\_code

if proc\_resp is not None:

self.proc\_resp = proc\_resp

if proc\_code is not None:

self.proc\_code = proc\_code

if proc\_mess is not None:

self.proc\_mess = proc\_mess

@property

def terminal\_id(self):

return self.\_terminal\_id

@terminal\_id.setter

def terminal\_id(self, terminal\_id):

self.\_terminal\_id = terminal\_id

@property

def account\_number(self):

return self.\_account\_number

@account\_number.setter

def account\_number(self, account\_number):

self.\_account\_number = account\_number

@property

def account\_type(self):

return self.\_account\_type

@account\_type.setter

def account\_type(self, account\_type):

self.\_account\_type = account\_type

@property

def check\_number(self):

return self.\_check\_number

@check\_number.setter

def check\_number(self, check\_number):

self.\_check\_number = check\_number

@property

def routing\_number(self):

return self.\_routing\_number

@routing\_number.setter

def routing\_number(self, routing\_number):

self.\_routing\_number = routing\_number

@property

def micr\_data(self):

return self.\_micr\_data

@micr\_data.setter

def micr\_data(self, micr\_data):

self.\_micr\_data = micr\_data

@property

def first\_name(self):

return self.\_first\_name

@first\_name.setter

def first\_name(self, first\_name):

self.\_first\_name = first\_name

@property

def last\_name(self):

return self.\_last\_name

@last\_name.setter

def last\_name(self, last\_name):

self.\_last\_name = last\_name

@property

def company\_name(self):

return self.\_company\_name

@company\_name.setter

def company\_name(self, company\_name):

self.\_company\_name = company\_name

@property

def address1(self):

return self.\_address1

@address1.setter

def address1(self, address1):

self.\_address1 = address1

@property

def address2(self):

return self.\_address2

@address2.setter

def address2(self, address2):

self.\_address2 = address2

@property

def city(self):

return self.\_city

@city.setter

def city(self, city):

self.\_city = city

@property

def state(self):

return self.\_state

@state.setter

def state(self, state):

self.\_state = state

@property

def zip(self):

return self.\_zip

@zip.setter

def zip(self, zip):

self.\_zip = zip

@property

def phone\_number(self):

return self.\_phone\_number

@phone\_number.setter

def phone\_number(self, phone\_number):

self.\_phone\_number = phone\_number

@property

def dl\_number(self):

return self.\_dl\_number

@dl\_number.setter

def dl\_number(self, dl\_number):

self.\_dl\_number = dl\_number

@property

def dl\_state(self):

return self.\_dl\_state

@dl\_state.setter

def dl\_state(self, dl\_state):

self.\_dl\_state = dl\_state

@property

def courtesy\_card\_id(self):

return self.\_courtesy\_card\_id

@courtesy\_card\_id.setter

def courtesy\_card\_id(self, courtesy\_card\_id):

self.\_courtesy\_card\_id = courtesy\_card\_id

@property

def ssn4(self):

return self.\_ssn4

@ssn4.setter

def ssn4(self, ssn4):

self.\_ssn4 = ssn4

@property

def dob\_year(self):

return self.\_dob\_year

@dob\_year.setter

def dob\_year(self, dob\_year):

self.\_dob\_year = dob\_year

@property

def check\_amount(self):

return self.\_check\_amount

@check\_amount.setter

def check\_amount(self, check\_amount):

self.\_check\_amount = check\_amount

@property

def tran\_type(self):

return self.\_tran\_type

@tran\_type.setter

def tran\_type(self, tran\_type):

self.\_tran\_type = tran\_type

@property

def proc\_id(self):

return self.\_proc\_id

@proc\_id.setter

def proc\_id(self, proc\_id):

self.\_proc\_id = proc\_id

@property

def ip\_address(self):

return self.\_ip\_address

@ip\_address.setter

def ip\_address(self, ip\_address):

self.\_ip\_address = ip\_address

@property

def customer\_number(self):

return self.\_customer\_number

@customer\_number.setter

def customer\_number(self, customer\_number):

self.\_customer\_number = customer\_number

@property

def po\_number(self):

return self.\_po\_number

@po\_number.setter

def po\_number(self, po\_number):

self.\_po\_number = po\_number

@property

def invoice\_number(self):

return self.\_invoice\_number

@invoice\_number.setter

def invoice\_number(self, invoice\_number):

self.\_invoice\_number = invoice\_number

@property

def tran\_note(self):

return self.\_tran\_note

@tran\_note.setter

def tran\_note(self, tran\_note):

self.\_tran\_note = tran\_note

@property

def email\_address(self):

return self.\_email\_address

@email\_address.setter

def email\_address(self, email\_address):

self.\_email\_address = email\_address

@property

def sec\_code(self):

return self.\_sec\_code

@sec\_code.setter

def sec\_code(self, sec\_code):

self.\_sec\_code = sec\_code

@property

def proc\_resp(self):

return self.\_proc\_resp

@proc\_resp.setter

def proc\_resp(self, proc\_resp):

self.\_proc\_resp = proc\_resp

@property

def proc\_code(self):

return self.\_proc\_code

@proc\_code.setter

def proc\_code(self, proc\_code):

self.\_proc\_code = proc\_code

@property

def proc\_mess(self):

return self.\_proc\_mess

@proc\_mess.setter

def proc\_mess(self, proc\_mess):

self.\_proc\_mess = proc\_mess

# CheckTransactionSettings

# coding: utf-8

class CheckTransactionSettings(object):

swagger\_types = {

'reader': 'FieldInfo',

'contact\_name': 'FieldInfo',

'contact\_title': 'FieldInfo',

'email\_address': 'FieldInfo',

'phone\_default': 'str',

'fax': 'FieldInfo',

'billing\_address\_default': 'str',

'billing\_address2': 'FieldInfo',

'billing\_city\_default': 'str',

'billing\_state\_default': 'str',

'billing\_zip\_default': 'str',

'po\_number': 'FieldInfo',

'invoice\_number': 'FieldInfo',

'business\_name': 'FieldInfo',

'customer\_number': 'FieldInfo',

'reference\_note': 'FieldInfo',

'default\_sec\_code': 'str'

}

attribute\_map = {

'reader': 'Reader',

'contact\_name': 'ContactName',

'contact\_title': 'ContactTitle',

'email\_address': 'EmailAddress',

'phone\_default': 'PhoneDefault',

'fax': 'Fax',

'billing\_address\_default': 'BillingAddressDefault',

'billing\_address2': 'BillingAddress2',

'billing\_city\_default': 'BillingCityDefault',

'billing\_state\_default': 'BillingStateDefault',

'billing\_zip\_default': 'BillingZipDefault',

'po\_number': 'PONumber',

'invoice\_number': 'InvoiceNumber',

'business\_name': 'BusinessName',

'customer\_number': 'CustomerNumber',

'reference\_note': 'ReferenceNote',

'default\_sec\_code': 'DefaultSECCode'

}

def \_\_init\_\_(self, reader=None, contact\_name=None, contact\_title=None, email\_address=None, phone\_default=None, fax=None, billing\_address\_default=None, billing\_address2=None, billing\_city\_default=None, billing\_state\_default=None, billing\_zip\_default=None, po\_number=None, invoice\_number=None, business\_name=None, customer\_number=None, reference\_note=None, default\_sec\_code=None): # noqa: E501

"""CheckTransactionSettings - a model defined in Swagger""" # noqa: E501

self.\_reader = None

self.\_contact\_name = None

self.\_contact\_title = None

self.\_email\_address = None

self.\_phone\_default = None

self.\_fax = None

self.\_billing\_address\_default = None

self.\_billing\_address2 = None

self.\_billing\_city\_default = None

self.\_billing\_state\_default = None

self.\_billing\_zip\_default = None

self.\_po\_number = None

self.\_invoice\_number = None

self.\_business\_name = None

self.\_customer\_number = None

self.\_reference\_note = None

self.\_default\_sec\_code = None

self.discriminator = None

if reader is not None:

self.reader = reader

if contact\_name is not None:

self.contact\_name = contact\_name

if contact\_title is not None:

self.contact\_title = contact\_title

if email\_address is not None:

self.email\_address = email\_address

if phone\_default is not None:

self.phone\_default = phone\_default

if fax is not None:

self.fax = fax

if billing\_address\_default is not None:

self.billing\_address\_default = billing\_address\_default

if billing\_address2 is not None:

self.billing\_address2 = billing\_address2

if billing\_city\_default is not None:

self.billing\_city\_default = billing\_city\_default

if billing\_state\_default is not None:

self.billing\_state\_default = billing\_state\_default

if billing\_zip\_default is not None:

self.billing\_zip\_default = billing\_zip\_default

if po\_number is not None:

self.po\_number = po\_number

if invoice\_number is not None:

self.invoice\_number = invoice\_number

if business\_name is not None:

self.business\_name = business\_name

if customer\_number is not None:

self.customer\_number = customer\_number

if reference\_note is not None:

self.reference\_note = reference\_note

if default\_sec\_code is not None:

self.default\_sec\_code = default\_sec\_code

@property

def reader(self):

return self.\_reader

@reader.setter

def reader(self, reader):

self.\_reader = reader

@property

def contact\_name(self):

return self.\_contact\_name

@contact\_name.setter

def contact\_name(self, contact\_name):

self.\_contact\_name = contact\_name

@property

def contact\_title(self):

return self.\_contact\_title

@contact\_title.setter

def contact\_title(self, contact\_title):

self.\_contact\_title = contact\_title

@property

def email\_address(self):

return self.\_email\_address

@email\_address.setter

def email\_address(self, email\_address):

self.\_email\_address = email\_address

@property

def phone\_default(self):

return self.\_phone\_default

@phone\_default.setter

def phone\_default(self, phone\_default):

self.\_phone\_default = phone\_default

@property

def fax(self):

return self.\_fax

@fax.setter

def fax(self, fax):

self.\_fax = fax

@property

def billing\_address\_default(self):

return self.\_billing\_address\_default

@billing\_address\_default.setter

def billing\_address\_default(self, billing\_address\_default):

self.\_billing\_address\_default = billing\_address\_default

@property

def billing\_address2(self):

return self.\_billing\_address2

@billing\_address2.setter

def billing\_address2(self, billing\_address2):

self.\_billing\_address2 = billing\_address2

@property

def billing\_city\_default(self):

return self.\_billing\_city\_default

@billing\_city\_default.setter

def billing\_city\_default(self, billing\_city\_default):

self.\_billing\_city\_default = billing\_city\_default

@property

def billing\_state\_default(self):

return self.\_billing\_state\_default

@billing\_state\_default.setter

def billing\_state\_default(self, billing\_state\_default):

self.\_billing\_state\_default = billing\_state\_default

@property

def billing\_zip\_default(self):

return self.\_billing\_zip\_default

@billing\_zip\_default.setter

def billing\_zip\_default(self, billing\_zip\_default):

self.\_billing\_zip\_default = billing\_zip\_default

@property

def po\_number(self):

return self.\_po\_number

@po\_number.setter

def po\_number(self, po\_number):

self.\_po\_number = po\_number

@property

def invoice\_number(self):

return self.\_invoice\_number

@invoice\_number.setter

def invoice\_number(self, invoice\_number):

self.\_invoice\_number = invoice\_number

@property

def business\_name(self):

return self.\_business\_name

@business\_name.setter

def business\_name(self, business\_name):

self.\_business\_name = business\_name

@property

def customer\_number(self):

return self.\_customer\_number

@customer\_number.setter

def customer\_number(self, customer\_number):

self.\_customer\_number = customer\_number

@property

def reference\_note(self):

return self.\_reference\_note

@reference\_note.setter

def reference\_note(self, reference\_note):

self.\_reference\_note = reference\_note

@property

def default\_sec\_code(self):

return self.\_default\_sec\_code

@default\_sec\_code.setter

def default\_sec\_code(self, default\_sec\_code):

self.\_default\_sec\_code = default\_sec\_code

# CheckTransactionType

# coding: utf-8

class CheckTransactionType(object):

UNKNOWN = "Unknown"

AUTHORIZE = "Authorize"

VOID = "Void"

OVERRIDE = "Override"

PAYROLL = "Payroll"

RECURRING = "Recurring"

REFUND = "Refund"

swagger\_types = {

}

attribute\_map = {

}

def \_\_init\_\_(self): # noqa: E501

"""CheckTransactionType - a model defined in Swagger""" # noqa: E501

self.discriminator = None

# Contact

# coding: utf-8

class Contact(object):

swagger\_types = {

'first\_name': 'str',

'last\_name': 'str',

'name': 'str',

'title': 'str',

'phone': 'str',

'email': 'str'

}

attribute\_map = {

'first\_name': 'FirstName',

'last\_name': 'LastName',

'name': 'Name',

'title': 'Title',

'phone': 'Phone',

'email': 'Email'

}

def \_\_init\_\_(self, first\_name=None, last\_name=None, name=None, title=None, phone=None, email=None): # noqa: E501

"""Contact - a model defined in Swagger""" # noqa: E501

self.\_first\_name = None

self.\_last\_name = None

self.\_name = None

self.\_title = None

self.\_phone = None

self.\_email = None

self.discriminator = None

if first\_name is not None:

self.first\_name = first\_name

if last\_name is not None:

self.last\_name = last\_name

if name is not None:

self.name = name

if title is not None:

self.title = title

if phone is not None:

self.phone = phone

if email is not None:

self.email = email

@property

def first\_name(self):

return self.\_first\_name

@first\_name.setter

def first\_name(self, first\_name):

self.\_first\_name = first\_name

@property

def last\_name(self):

return self.\_last\_name

@last\_name.setter

def last\_name(self, last\_name):

self.\_last\_name = last\_name

@property

def name(self):

return self.\_name

@name.setter

def name(self, name):

self.\_name = name

@property

def title(self):

return self.\_title

@title.setter

def title(self, title):

self.\_title = title

@property

def phone(self):

return self.\_phone

@phone.setter

def phone(self, phone):

self.\_phone = phone

@property

def email(self):

return self.\_email

@email.setter

def email(self, email):

self.\_email = email

# CreditCardTransactionSettings

# coding: utf-8

class CreditCardTransactionSettings(object):

swagger\_types = {

'cvv': 'FieldInfo',

'contact\_name': 'FieldInfo',

'contact\_title': 'FieldInfo',

'email\_address': 'FieldInfo',

'phone': 'FieldInfo',

'fax': 'FieldInfo',

'billing\_address': 'FieldInfo',

'billing\_address2': 'FieldInfo',

'billing\_city': 'FieldInfo',

'billing\_state': 'FieldInfo',

'billing\_zip': 'FieldInfo',

'po\_number': 'FieldInfo',

'invoice\_number': 'FieldInfo',

'business\_name': 'FieldInfo',

'customer\_number': 'FieldInfo',

'reference\_note': 'FieldInfo',

'sales\_tax': 'FieldInfo',

'tax\_exempt': 'FieldInfo',

'card\_tracks': 'FieldInfo',

'card\_zip': 'FieldInfo',

'card\_address': 'FieldInfo',

'item\_description': 'str',

'default\_transaction\_type': 'TransactionType'

}

attribute\_map = {

'cvv': 'CVV',

'contact\_name': 'ContactName',

'contact\_title': 'ContactTitle',

'email\_address': 'EmailAddress',

'phone': 'Phone',

'fax': 'Fax',

'billing\_address': 'BillingAddress',

'billing\_address2': 'BillingAddress2',

'billing\_city': 'BillingCity',

'billing\_state': 'BillingState',

'billing\_zip': 'BillingZip',

'po\_number': 'PONumber',

'invoice\_number': 'InvoiceNumber',

'business\_name': 'BusinessName',

'customer\_number': 'CustomerNumber',

'reference\_note': 'ReferenceNote',

'sales\_tax': 'SalesTax',

'tax\_exempt': 'TaxExempt',

'card\_tracks': 'CardTracks',

'card\_zip': 'CardZip',

'card\_address': 'CardAddress',

'item\_description': 'ItemDescription',

'default\_transaction\_type': 'DefaultTransactionType'

}

def \_\_init\_\_(self, cvv=None, contact\_name=None, contact\_title=None, email\_address=None, phone=None, fax=None, billing\_address=None, billing\_address2=None, billing\_city=None, billing\_state=None, billing\_zip=None, po\_number=None, invoice\_number=None, business\_name=None, customer\_number=None, reference\_note=None, sales\_tax=None, tax\_exempt=None, card\_tracks=None, card\_zip=None, card\_address=None, item\_description=None, default\_transaction\_type=None): # noqa: E501

"""CreditCardTransactionSettings - a model defined in Swagger""" # noqa: E501

self.\_cvv = None

self.\_contact\_name = None

self.\_contact\_title = None

self.\_email\_address = None

self.\_phone = None

self.\_fax = None

self.\_billing\_address = None

self.\_billing\_address2 = None

self.\_billing\_city = None

self.\_billing\_state = None

self.\_billing\_zip = None

self.\_po\_number = None

self.\_invoice\_number = None

self.\_business\_name = None

self.\_customer\_number = None

self.\_reference\_note = None

self.\_sales\_tax = None

self.\_tax\_exempt = None

self.\_card\_tracks = None

self.\_card\_zip = None

self.\_card\_address = None

self.\_item\_description = None

self.\_default\_transaction\_type = None

self.discriminator = None

if cvv is not None:

self.cvv = cvv

if contact\_name is not None:

self.contact\_name = contact\_name

if contact\_title is not None:

self.contact\_title = contact\_title

if email\_address is not None:

self.email\_address = email\_address

if phone is not None:

self.phone = phone

if fax is not None:

self.fax = fax

if billing\_address is not None:

self.billing\_address = billing\_address

if billing\_address2 is not None:

self.billing\_address2 = billing\_address2

if billing\_city is not None:

self.billing\_city = billing\_city

if billing\_state is not None:

self.billing\_state = billing\_state

if billing\_zip is not None:

self.billing\_zip = billing\_zip

if po\_number is not None:

self.po\_number = po\_number

if invoice\_number is not None:

self.invoice\_number = invoice\_number

if business\_name is not None:

self.business\_name = business\_name

if customer\_number is not None:

self.customer\_number = customer\_number

if reference\_note is not None:

self.reference\_note = reference\_note

if sales\_tax is not None:

self.sales\_tax = sales\_tax

if tax\_exempt is not None:

self.tax\_exempt = tax\_exempt

if card\_tracks is not None:

self.card\_tracks = card\_tracks

if card\_zip is not None:

self.card\_zip = card\_zip

if card\_address is not None:

self.card\_address = card\_address

if item\_description is not None:

self.item\_description = item\_description

if default\_transaction\_type is not None:

self.default\_transaction\_type = default\_transaction\_type

@property

def cvv(self):

return self.\_cvv

@cvv.setter

def cvv(self, cvv):

self.\_cvv = cvv

@property

def contact\_name(self):

return self.\_contact\_name

@contact\_name.setter

def contact\_name(self, contact\_name):

self.\_contact\_name = contact\_name

@property

def contact\_title(self):

return self.\_contact\_title

@contact\_title.setter

def contact\_title(self, contact\_title):

self.\_contact\_title = contact\_title

@property

def email\_address(self):

return self.\_email\_address

@email\_address.setter

def email\_address(self, email\_address):

self.\_email\_address = email\_address

@property

def phone(self):

return self.\_phone

@phone.setter

def phone(self, phone):

self.\_phone = phone

@property

def fax(self):

return self.\_fax

@fax.setter

def fax(self, fax):

self.\_fax = fax

@property

def billing\_address(self):

return self.\_billing\_address

@billing\_address.setter

def billing\_address(self, billing\_address):

self.\_billing\_address = billing\_address

@property

def billing\_address2(self):

return self.\_billing\_address2

@billing\_address2.setter

def billing\_address2(self, billing\_address2):

self.\_billing\_address2 = billing\_address2

@property

def billing\_city(self):

return self.\_billing\_city

@billing\_city.setter

def billing\_city(self, billing\_city):

self.\_billing\_city = billing\_city

@property

def billing\_state(self):

return self.\_billing\_state

@billing\_state.setter

def billing\_state(self, billing\_state):

self.\_billing\_state = billing\_state

@property

def billing\_zip(self):

return self.\_billing\_zip

@billing\_zip.setter

def billing\_zip(self, billing\_zip):

self.\_billing\_zip = billing\_zip

@property

def po\_number(self):

return self.\_po\_number

@po\_number.setter

def po\_number(self, po\_number):

self.\_po\_number = po\_number

@property

def invoice\_number(self):

return self.\_invoice\_number

@invoice\_number.setter

def invoice\_number(self, invoice\_number):

self.\_invoice\_number = invoice\_number

@property

def business\_name(self):

return self.\_business\_name

@business\_name.setter

def business\_name(self, business\_name):

self.\_business\_name = business\_name

@property

def customer\_number(self):

return self.\_customer\_number

@customer\_number.setter

def customer\_number(self, customer\_number):

self.\_customer\_number = customer\_number

@property

def reference\_note(self):

return self.\_reference\_note

@reference\_note.setter

def reference\_note(self, reference\_note):

self.\_reference\_note = reference\_note

@property

def sales\_tax(self):

return self.\_sales\_tax

@sales\_tax.setter

def sales\_tax(self, sales\_tax):

self.\_sales\_tax = sales\_tax

@property

def tax\_exempt(self):

return self.\_tax\_exempt

@tax\_exempt.setter

def tax\_exempt(self, tax\_exempt):

self.\_tax\_exempt = tax\_exempt

@property

def card\_tracks(self):

return self.\_card\_tracks

@card\_tracks.setter

def card\_tracks(self, card\_tracks):

self.\_card\_tracks = card\_tracks

@property

def card\_zip(self):

return self.\_card\_zip

@card\_zip.setter

def card\_zip(self, card\_zip):

self.\_card\_zip = card\_zip

@property

def card\_address(self):

return self.\_card\_address

@card\_address.setter

def card\_address(self, card\_address):

self.\_card\_address = card\_address

@property

def item\_description(self):

return self.\_item\_description

@item\_description.setter

def item\_description(self, item\_description):

self.\_item\_description = item\_description

@property

def default\_transaction\_type(self):

return self.\_default\_transaction\_type

@default\_transaction\_type.setter

def default\_transaction\_type(self, default\_transaction\_type):

self.\_default\_transaction\_type = default\_transaction\_type

# CVVPresenceType

# coding: utf-8

class CVVPresenceType(object):

NORMAL = "Normal"

BYPASS = "Bypass"

ILLEGIBLE = "Illegible"

NOTAVAILABLE = "NotAvailable"

swagger\_types = {

}

attribute\_map = {

}

def \_\_init\_\_(self): # noqa: E501

"""CVVPresenceType - a model defined in Swagger""" # noqa: E501

self.discriminator = None

# EmailReceipt

# coding: utf-8

class EmailReceipt(object):

swagger\_types = {

'transaction\_id': 'str',

'to\_addresses': 'str',

'subject': 'str',

'header': 'str',

'footer': 'str'

}

attribute\_map = {

'transaction\_id': 'TransactionID',

'to\_addresses': 'ToAddresses',

'subject': 'Subject',

'header': 'Header',

'footer': 'Footer'

}

def \_\_init\_\_(self, transaction\_id=None, to\_addresses=None, subject=None, header=None, footer=None): # noqa: E501

"""EmailReceipt - a model defined in Swagger""" # noqa: E501

self.\_transaction\_id = None

self.\_to\_addresses = None

self.\_subject = None

self.\_header = None

self.\_footer = None

self.discriminator = None

if transaction\_id is not None:

self.transaction\_id = transaction\_id

if to\_addresses is not None:

self.to\_addresses = to\_addresses

if subject is not None:

self.subject = subject

if header is not None:

self.header = header

if footer is not None:

self.footer = footer

@property

def transaction\_id(self):

return self.\_transaction\_id

@transaction\_id.setter

def transaction\_id(self, transaction\_id):

self.\_transaction\_id = transaction\_id

@property

def to\_addresses(self):

return self.\_to\_addresses

@to\_addresses.setter

def to\_addresses(self, to\_addresses):

self.\_to\_addresses = to\_addresses

@property

def subject(self):

return self.\_subject

@subject.setter

def subject(self, subject):

self.\_subject = subject

@property

def header(self):

return self.\_header

@header.setter

def header(self, header):

self.\_header = header

@property

def footer(self):

return self.\_footer

@footer.setter

def footer(self, footer):

self.\_footer = footer

# FieldAccess

# coding: utf-8

class FieldAccess(object):

DISABLED = "Disabled"

ENABLED = "Enabled"

REQUIRED = "Required"

swagger\_types = {

}

attribute\_map = {

}

def \_\_init\_\_(self): # noqa: E501

"""FieldAccess - a model defined in Swagger""" # noqa: E501

self.discriminator = None

# FieldInfo

# coding: utf-8

class FieldInfo(object):

swagger\_types = {

'access': 'FieldAccess',

'default': 'str'

}

attribute\_map = {

'access': 'Access',

'default': 'Default'

}

def \_\_init\_\_(self, access=None, default=None): # noqa: E501

"""FieldInfo - a model defined in Swagger""" # noqa: E501

self.\_access = None

self.\_default = None

self.discriminator = None

if access is not None:

self.access = access

if default is not None:

self.default = default

@property

def access(self):

return self.\_access

@access.setter

def access(self, access):

self.\_access = access

@property

def default(self):

return self.\_default

@default.setter

def default(self, default):

self.\_default = default

# QueryTransaction

# coding: utf-8

class QueryTransaction(object):

swagger\_types = {

'tran\_sequence': 'str',

'tran\_batch': 'str',

'tran\_modified': 'str',

'tran\_processed': 'str',

'tran\_settled': 'str',

'submit\_time': 'str',

'tran\_type': 'str',

'tran\_amt': 'float',

'tran\_tax': 'float',

'tran\_inv': 'str',

'tran\_po': 'str',

'tran\_c\_num': 'str',

'tran\_note': 'str',

'card\_num': 'str',

'card\_name': 'str',

'card\_exp': 'str',

'avs\_street': 'str',

'avs\_zip': 'str',

'cvv\_num': 'str',

'proc\_id': 'str',

'proc\_resp': 'str',

'proc\_code': 'str',

'proc\_mess': 'str',

'avs\_code': 'str',

'avs\_resp': 'str',

'cvv\_code': 'str',

'cvv\_resp': 'str',

'items': 'list[TransactionItem]',

'business\_name': 'str',

'business\_address': 'str',

'business\_address2': 'str',

'business\_city': 'str',

'business\_state': 'str',

'business\_zip': 'str',

'business\_country': 'str',

'business\_website': 'str',

'contact\_first\_name': 'str',

'contact\_last\_name': 'str',

'contact\_phone': 'str',

'contact\_fax': 'str',

'contact\_email': 'str',

'contact\_title': 'str',

'ip': 'str',

'tax\_exempt': 'bool'

}

attribute\_map = {

'tran\_sequence': 'Tran\_Sequence',

'tran\_batch': 'Tran\_Batch',

'tran\_modified': 'Tran\_Modified',

'tran\_processed': 'Tran\_Processed',

'tran\_settled': 'Tran\_Settled',

'submit\_time': 'Submit\_Time',

'tran\_type': 'Tran\_Type',

'tran\_amt': 'Tran\_Amt',

'tran\_tax': 'Tran\_Tax',

'tran\_inv': 'Tran\_Inv',

'tran\_po': 'Tran\_PO',

'tran\_c\_num': 'Tran\_CNum',

'tran\_note': 'Tran\_Note',

'card\_num': 'Card\_Num',

'card\_name': 'Card\_Name',

'card\_exp': 'Card\_Exp',

'avs\_street': 'AVS\_Street',

'avs\_zip': 'AVS\_Zip',

'cvv\_num': 'CVV\_Num',

'proc\_id': 'Proc\_ID',

'proc\_resp': 'Proc\_Resp',

'proc\_code': 'Proc\_Code',

'proc\_mess': 'Proc\_Mess',

'avs\_code': 'AVS\_Code',

'avs\_resp': 'AVS\_Resp',

'cvv\_code': 'CVV\_Code',

'cvv\_resp': 'CVV\_Resp',

'items': 'Items',

'business\_name': 'Business\_Name',

'business\_address': 'Business\_Address',

'business\_address2': 'Business\_Address2',

'business\_city': 'Business\_City',

'business\_state': 'Business\_State',

'business\_zip': 'Business\_Zip',

'business\_country': 'Business\_Country',

'business\_website': 'Business\_Website',

'contact\_first\_name': 'Contact\_FirstName',

'contact\_last\_name': 'Contact\_LastName',

'contact\_phone': 'Contact\_Phone',

'contact\_fax': 'Contact\_Fax',

'contact\_email': 'Contact\_Email',

'contact\_title': 'Contact\_Title',

'ip': 'IP',

'tax\_exempt': 'TaxExempt'

}

def \_\_init\_\_(self, tran\_sequence=None, tran\_batch=None, tran\_modified=None, tran\_processed=None, tran\_settled=None, submit\_time=None, tran\_type=None, tran\_amt=None, tran\_tax=None, tran\_inv=None, tran\_po=None, tran\_c\_num=None, tran\_note=None, card\_num=None, card\_name=None, card\_exp=None, avs\_street=None, avs\_zip=None, cvv\_num=None, proc\_id=None, proc\_resp=None, proc\_code=None, proc\_mess=None, avs\_code=None, avs\_resp=None, cvv\_code=None, cvv\_resp=None, items=None, business\_name=None, business\_address=None, business\_address2=None, business\_city=None, business\_state=None, business\_zip=None, business\_country=None, business\_website=None, contact\_first\_name=None, contact\_last\_name=None, contact\_phone=None, contact\_fax=None, contact\_email=None, contact\_title=None, ip=None, tax\_exempt=None): # noqa: E501

"""QueryTransaction - a model defined in Swagger""" # noqa: E501

self.\_tran\_sequence = None

self.\_tran\_batch = None

self.\_tran\_modified = None

self.\_tran\_processed = None

self.\_tran\_settled = None

self.\_submit\_time = None

self.\_tran\_type = None

self.\_tran\_amt = None

self.\_tran\_tax = None

self.\_tran\_inv = None

self.\_tran\_po = None

self.\_tran\_c\_num = None

self.\_tran\_note = None

self.\_card\_num = None

self.\_card\_name = None

self.\_card\_exp = None

self.\_avs\_street = None

self.\_avs\_zip = None

self.\_cvv\_num = None

self.\_proc\_id = None

self.\_proc\_resp = None

self.\_proc\_code = None

self.\_proc\_mess = None

self.\_avs\_code = None

self.\_avs\_resp = None

self.\_cvv\_code = None

self.\_cvv\_resp = None

self.\_items = None

self.\_business\_name = None

self.\_business\_address = None

self.\_business\_address2 = None

self.\_business\_city = None

self.\_business\_state = None

self.\_business\_zip = None

self.\_business\_country = None

self.\_business\_website = None

self.\_contact\_first\_name = None

self.\_contact\_last\_name = None

self.\_contact\_phone = None

self.\_contact\_fax = None

self.\_contact\_email = None

self.\_contact\_title = None

self.\_ip = None

self.\_tax\_exempt = None

self.discriminator = None

if tran\_sequence is not None:

self.tran\_sequence = tran\_sequence

if tran\_batch is not None:

self.tran\_batch = tran\_batch

if tran\_modified is not None:

self.tran\_modified = tran\_modified

if tran\_processed is not None:

self.tran\_processed = tran\_processed

if tran\_settled is not None:

self.tran\_settled = tran\_settled

if submit\_time is not None:

self.submit\_time = submit\_time

if tran\_type is not None:

self.tran\_type = tran\_type

if tran\_amt is not None:

self.tran\_amt = tran\_amt

if tran\_tax is not None:

self.tran\_tax = tran\_tax

if tran\_inv is not None:

self.tran\_inv = tran\_inv

if tran\_po is not None:

self.tran\_po = tran\_po

if tran\_c\_num is not None:

self.tran\_c\_num = tran\_c\_num

if tran\_note is not None:

self.tran\_note = tran\_note

if card\_num is not None:

self.card\_num = card\_num

if card\_name is not None:

self.card\_name = card\_name

if card\_exp is not None:

self.card\_exp = card\_exp

if avs\_street is not None:

self.avs\_street = avs\_street

if avs\_zip is not None:

self.avs\_zip = avs\_zip

if cvv\_num is not None:

self.cvv\_num = cvv\_num

if proc\_id is not None:

self.proc\_id = proc\_id

if proc\_resp is not None:

self.proc\_resp = proc\_resp

if proc\_code is not None:

self.proc\_code = proc\_code

if proc\_mess is not None:

self.proc\_mess = proc\_mess

if avs\_code is not None:

self.avs\_code = avs\_code

if avs\_resp is not None:

self.avs\_resp = avs\_resp

if cvv\_code is not None:

self.cvv\_code = cvv\_code

if cvv\_resp is not None:

self.cvv\_resp = cvv\_resp

if items is not None:

self.items = items

if business\_name is not None:

self.business\_name = business\_name

if business\_address is not None:

self.business\_address = business\_address

if business\_address2 is not None:

self.business\_address2 = business\_address2

if business\_city is not None:

self.business\_city = business\_city

if business\_state is not None:

self.business\_state = business\_state

if business\_zip is not None:

self.business\_zip = business\_zip

if business\_country is not None:

self.business\_country = business\_country

if business\_website is not None:

self.business\_website = business\_website

if contact\_first\_name is not None:

self.contact\_first\_name = contact\_first\_name

if contact\_last\_name is not None:

self.contact\_last\_name = contact\_last\_name

if contact\_phone is not None:

self.contact\_phone = contact\_phone

if contact\_fax is not None:

self.contact\_fax = contact\_fax

if contact\_email is not None:

self.contact\_email = contact\_email

if contact\_title is not None:

self.contact\_title = contact\_title

if ip is not None:

self.ip = ip

if tax\_exempt is not None:

self.tax\_exempt = tax\_exempt

@property

def tran\_sequence(self):

return self.\_tran\_sequence

@tran\_sequence.setter

def tran\_sequence(self, tran\_sequence):

self.\_tran\_sequence = tran\_sequence

@property

def tran\_batch(self):

return self.\_tran\_batch

@tran\_batch.setter

def tran\_batch(self, tran\_batch):

self.\_tran\_batch = tran\_batch

@property

def tran\_modified(self):

return self.\_tran\_modified

@tran\_modified.setter

def tran\_modified(self, tran\_modified):

self.\_tran\_modified = tran\_modified

@property

def tran\_processed(self):

return self.\_tran\_processed

@tran\_processed.setter

def tran\_processed(self, tran\_processed):

self.\_tran\_processed = tran\_processed

@property

def tran\_settled(self):

return self.\_tran\_settled

@tran\_settled.setter

def tran\_settled(self, tran\_settled):

self.\_tran\_settled = tran\_settled

@property

def submit\_time(self):

return self.\_submit\_time

@submit\_time.setter

def submit\_time(self, submit\_time):

self.\_submit\_time = submit\_time

@property

def tran\_type(self):

return self.\_tran\_type

@tran\_type.setter

def tran\_type(self, tran\_type):

self.\_tran\_type = tran\_type

@property

def tran\_amt(self):

return self.\_tran\_amt

@tran\_amt.setter

def tran\_amt(self, tran\_amt):

self.\_tran\_amt = tran\_amt

@property

def tran\_tax(self):

return self.\_tran\_tax

@tran\_tax.setter

def tran\_tax(self, tran\_tax):

self.\_tran\_tax = tran\_tax

@property

def tran\_inv(self):

return self.\_tran\_inv

@tran\_inv.setter

def tran\_inv(self, tran\_inv):

self.\_tran\_inv = tran\_inv

@property

def tran\_po(self):

return self.\_tran\_po

@tran\_po.setter

def tran\_po(self, tran\_po):

self.\_tran\_po = tran\_po

@property

def tran\_c\_num(self):

return self.\_tran\_c\_num

@tran\_c\_num.setter

def tran\_c\_num(self, tran\_c\_num):

self.\_tran\_c\_num = tran\_c\_num

@property

def tran\_note(self):

return self.\_tran\_note

@tran\_note.setter

def tran\_note(self, tran\_note):

self.\_tran\_note = tran\_note

@property

def card\_num(self):

return self.\_card\_num

@card\_num.setter

def card\_num(self, card\_num):

self.\_card\_num = card\_num

@property

def card\_name(self):

return self.\_card\_name

@card\_name.setter

def card\_name(self, card\_name):

self.\_card\_name = card\_name

@property

def card\_exp(self):

return self.\_card\_exp

@card\_exp.setter

def card\_exp(self, card\_exp):

self.\_card\_exp = card\_exp

@property

def avs\_street(self):

return self.\_avs\_street

@avs\_street.setter

def avs\_street(self, avs\_street):

self.\_avs\_street = avs\_street

@property

def avs\_zip(self):

return self.\_avs\_zip

@avs\_zip.setter

def avs\_zip(self, avs\_zip):

self.\_avs\_zip = avs\_zip

@property

def cvv\_num(self):

return self.\_cvv\_num

@cvv\_num.setter

def cvv\_num(self, cvv\_num):

self.\_cvv\_num = cvv\_num

@property

def proc\_id(self):

return self.\_proc\_id

@proc\_id.setter

def proc\_id(self, proc\_id):

self.\_proc\_id = proc\_id

@property

def proc\_resp(self):

return self.\_proc\_resp

@proc\_resp.setter

def proc\_resp(self, proc\_resp):

self.\_proc\_resp = proc\_resp

@property

def proc\_code(self):

return self.\_proc\_code

@proc\_code.setter

def proc\_code(self, proc\_code):

self.\_proc\_code = proc\_code

@property

def proc\_mess(self):

return self.\_proc\_mess

@proc\_mess.setter

def proc\_mess(self, proc\_mess):

self.\_proc\_mess = proc\_mess

@property

def avs\_code(self):

return self.\_avs\_code

@avs\_code.setter

def avs\_code(self, avs\_code):

self.\_avs\_code = avs\_code

@property

def avs\_resp(self):

return self.\_avs\_resp

@avs\_resp.setter

def avs\_resp(self, avs\_resp):

self.\_avs\_resp = avs\_resp

@property

def cvv\_code(self):

return self.\_cvv\_code

@cvv\_code.setter

def cvv\_code(self, cvv\_code):

self.\_cvv\_code = cvv\_code

@property

def cvv\_resp(self):

return self.\_cvv\_resp

@cvv\_resp.setter

def cvv\_resp(self, cvv\_resp):

self.\_cvv\_resp = cvv\_resp

@property

def items(self):

return self.\_items

@items.setter

def items(self, items):

self.\_items = items

@property

def business\_name(self):

return self.\_business\_name

@business\_name.setter

def business\_name(self, business\_name):

self.\_business\_name = business\_name

@property

def business\_address(self):

return self.\_business\_address

@business\_address.setter

def business\_address(self, business\_address):

self.\_business\_address = business\_address

@property

def business\_address2(self):

return self.\_business\_address2

@business\_address2.setter

def business\_address2(self, business\_address2):

self.\_business\_address2 = business\_address2

@property

def business\_city(self):

return self.\_business\_city

@business\_city.setter

def business\_city(self, business\_city):

self.\_business\_city = business\_city

@property

def business\_state(self):

return self.\_business\_state

@business\_state.setter

def business\_state(self, business\_state):

self.\_business\_state = business\_state

@property

def business\_zip(self):

return self.\_business\_zip

@business\_zip.setter

def business\_zip(self, business\_zip):

self.\_business\_zip = business\_zip

@property

def business\_country(self):

return self.\_business\_country

@business\_country.setter

def business\_country(self, business\_country):

self.\_business\_country = business\_country

@property

def business\_website(self):

return self.\_business\_website

@business\_website.setter

def business\_website(self, business\_website):

self.\_business\_website = business\_website

@property

def contact\_first\_name(self):

return self.\_contact\_first\_name

@contact\_first\_name.setter

def contact\_first\_name(self, contact\_first\_name):

self.\_contact\_first\_name = contact\_first\_name

@property

def contact\_last\_name(self):

return self.\_contact\_last\_name

@contact\_last\_name.setter

def contact\_last\_name(self, contact\_last\_name):

self.\_contact\_last\_name = contact\_last\_name

@property

def contact\_phone(self):

return self.\_contact\_phone

@contact\_phone.setter

def contact\_phone(self, contact\_phone):

self.\_contact\_phone = contact\_phone

@property

def contact\_fax(self):

return self.\_contact\_fax

@contact\_fax.setter

def contact\_fax(self, contact\_fax):

self.\_contact\_fax = contact\_fax

@property

def contact\_email(self):

return self.\_contact\_email

@contact\_email.setter

def contact\_email(self, contact\_email):

self.\_contact\_email = contact\_email

@property

def contact\_title(self):

return self.\_contact\_title

@contact\_title.setter

def contact\_title(self, contact\_title):

self.\_contact\_title = contact\_title

@property

def ip(self):

return self.\_ip

@ip.setter

def ip(self, ip):

self.\_ip = ip

@property

def tax\_exempt(self):

return self.\_tax\_exempt

@tax\_exempt.setter

def tax\_exempt(self, tax\_exempt):

self.\_tax\_exempt = tax\_exempt

# Receipt

# coding: utf-8

class Receipt(object):

swagger\_types = {

'header': 'ReceiptCustomLines',

'footer': 'ReceiptCustomLines'

}

attribute\_map = {

'header': 'Header',

'footer': 'Footer'

}

def \_\_init\_\_(self, header=None, footer=None): # noqa: E501

"""Receipt - a model defined in Swagger""" # noqa: E501

self.\_header = None

self.\_footer = None

self.discriminator = None

if header is not None:

self.header = header

if footer is not None:

self.footer = footer

@property

def header(self):

return self.\_header

@header.setter

def header(self, header):

self.\_header = header

@property

def footer(self):

return self.\_footer

@footer.setter

def footer(self, footer):

self.\_footer = footer

# ReceiptCustomLines

# coding: utf-8

class ReceiptCustomLines(object):

swagger\_types = {

'custom\_line1': 'str',

'custom\_line2': 'str',

'custom\_line3': 'str',

'custom\_line4': 'str'

}

attribute\_map = {

'custom\_line1': 'CustomLine1',

'custom\_line2': 'CustomLine2',

'custom\_line3': 'CustomLine3',

'custom\_line4': 'CustomLine4'

}

def \_\_init\_\_(self, custom\_line1=None, custom\_line2=None, custom\_line3=None, custom\_line4=None): # noqa: E501

"""ReceiptCustomLines - a model defined in Swagger""" # noqa: E501

self.\_custom\_line1 = None

self.\_custom\_line2 = None

self.\_custom\_line3 = None

self.\_custom\_line4 = None

self.discriminator = None

if custom\_line1 is not None:

self.custom\_line1 = custom\_line1

if custom\_line2 is not None:

self.custom\_line2 = custom\_line2

if custom\_line3 is not None:

self.custom\_line3 = custom\_line3

if custom\_line4 is not None:

self.custom\_line4 = custom\_line4

@property

def custom\_line1(self):

return self.\_custom\_line1

@custom\_line1.setter

def custom\_line1(self, custom\_line1):

self.\_custom\_line1 = custom\_line1

@property

def custom\_line2(self):

return self.\_custom\_line2

@custom\_line2.setter

def custom\_line2(self, custom\_line2):

self.\_custom\_line2 = custom\_line2

@property

def custom\_line3(self):

return self.\_custom\_line3

@custom\_line3.setter

def custom\_line3(self, custom\_line3):

self.\_custom\_line3 = custom\_line3

@property

def custom\_line4(self):

return self.\_custom\_line4

@custom\_line4.setter

def custom\_line4(self, custom\_line4):

self.\_custom\_line4 = custom\_line4

# Record

# coding: utf-8

class Record(object):

swagger\_types = {

'response': 'Response',

'id': 'int',

'reference': 'str',

'business': 'Business',

'contact': 'Contact',

'po': 'str',

'invoice': 'str',

'customer': 'str',

'tax\_exempt': 'bool',

'note': 'str',

'credit\_cards': 'list[VaultCreditCard]',

'checks': 'list[VaultCheck]'

}

attribute\_map = {

'response': 'Response',

'id': 'ID',

'reference': 'Reference',

'business': 'Business',

'contact': 'Contact',

'po': 'PO',

'invoice': 'Invoice',

'customer': 'Customer',

'tax\_exempt': 'TaxExempt',

'note': 'Note',

'credit\_cards': 'CreditCards',

'checks': 'Checks'

}

def \_\_init\_\_(self, response=None, id=None, reference=None, business=None, contact=None, po=None, invoice=None, customer=None, tax\_exempt=None, note=None, credit\_cards=None, checks=None): # noqa: E501

"""Record - a model defined in Swagger""" # noqa: E501

self.\_response = None

self.\_id = None

self.\_reference = None

self.\_business = None

self.\_contact = None

self.\_po = None

self.\_invoice = None

self.\_customer = None

self.\_tax\_exempt = None

self.\_note = None

self.\_credit\_cards = None

self.\_checks = None

self.discriminator = None

if response is not None:

self.response = response

if id is not None:

self.id = id

if reference is not None:

self.reference = reference

if business is not None:

self.business = business

if contact is not None:

self.contact = contact

if po is not None:

self.po = po

if invoice is not None:

self.invoice = invoice

if customer is not None:

self.customer = customer

if tax\_exempt is not None:

self.tax\_exempt = tax\_exempt

if note is not None:

self.note = note

if credit\_cards is not None:

self.credit\_cards = credit\_cards

if checks is not None:

self.checks = checks

@property

def response(self):

return self.\_response

@response.setter

def response(self, response):

self.\_response = response

@property

def id(self):

return self.\_id

@id.setter

def id(self, id):

self.\_id = id

@property

def reference(self):

return self.\_reference

@reference.setter

def reference(self, reference):

self.\_reference = reference

@property

def business(self):

return self.\_business

@business.setter

def business(self, business):

self.\_business = business

@property

def contact(self):

return self.\_contact

@contact.setter

def contact(self, contact):

self.\_contact = contact

@property

def po(self):

return self.\_po

@po.setter

def po(self, po):

self.\_po = po

@property

def invoice(self):

return self.\_invoice

@invoice.setter

def invoice(self, invoice):

self.\_invoice = invoice

@property

def customer(self):

return self.\_customer

@customer.setter

def customer(self, customer):

self.\_customer = customer

@property

def tax\_exempt(self):

return self.\_tax\_exempt

@tax\_exempt.setter

def tax\_exempt(self, tax\_exempt):

self.\_tax\_exempt = tax\_exempt

@property

def note(self):

return self.\_note

@note.setter

def note(self, note):

self.\_note = note

@property

def credit\_cards(self):

return self.\_credit\_cards

@credit\_cards.setter

def credit\_cards(self, credit\_cards):

self.\_credit\_cards = credit\_cards

@property

def checks(self):

return self.\_checks

@checks.setter

def checks(self, checks):

self.\_checks = checks

# Response

# coding: utf-8

class Response(object):

swagger\_types = {

'error': 'bool',

'message': 'str'

}

attribute\_map = {

'error': 'Error',

'message': 'Message'

}

def \_\_init\_\_(self, error=None, message=None): # noqa: E501

"""Response - a model defined in Swagger""" # noqa: E501

self.\_error = None

self.\_message = None

self.discriminator = None

if error is not None:

self.error = error

if message is not None:

self.message = message

@property

def error(self):

return self.\_error

@error.setter

def error(self, error):

self.\_error = error

@property

def message(self):

return self.\_message

@message.setter

def message(self, message):

self.\_message = message

# SearchVault

# coding: utf-8

class SearchVault(object):

swagger\_types = {

'reference': 'str',

'business\_name': 'str',

'first\_name': 'str',

'last\_name': 'str'

}

attribute\_map = {

'reference': 'Reference',

'business\_name': 'BusinessName',

'first\_name': 'FirstName',

'last\_name': 'LastName'

}

def \_\_init\_\_(self, reference=None, business\_name=None, first\_name=None, last\_name=None): # noqa: E501

"""SearchVault - a model defined in Swagger""" # noqa: E501

self.\_reference = None

self.\_business\_name = None

self.\_first\_name = None

self.\_last\_name = None

self.discriminator = None

if reference is not None:

self.reference = reference

if business\_name is not None:

self.business\_name = business\_name

if first\_name is not None:

self.first\_name = first\_name

if last\_name is not None:

self.last\_name = last\_name

@property

def reference(self):

return self.\_reference

@reference.setter

def reference(self, reference):

self.\_reference = reference

@property

def business\_name(self):

return self.\_business\_name

@business\_name.setter

def business\_name(self, business\_name):

self.\_business\_name = business\_name

@property

def first\_name(self):

return self.\_first\_name

@first\_name.setter

def first\_name(self, first\_name):

self.\_first\_name = first\_name

@property

def last\_name(self):

return self.\_last\_name

@last\_name.setter

def last\_name(self, last\_name):

self.\_last\_name = last\_name

# Settings

# coding: utf-8

class Settings(object):

swagger\_types = {

'credit\_card\_transaction\_settings': 'CreditCardTransactionSettings',

'check\_transaction\_settings': 'CheckTransactionSettings',

'business': 'Business',

'main\_contact': 'Contact',

'receipt40': 'Receipt',

'receipt80': 'Receipt',

'print\_signature\_line': 'bool',

'response': 'Response',

'account\_id': 'str',

'copy\_email\_receipt': 'FieldInfo'

}

attribute\_map = {

'credit\_card\_transaction\_settings': 'CreditCardTransactionSettings',

'check\_transaction\_settings': 'CheckTransactionSettings',

'business': 'Business',

'main\_contact': 'MainContact',

'receipt40': 'Receipt40',

'receipt80': 'Receipt80',

'print\_signature\_line': 'PrintSignatureLine',

'response': 'Response',

'account\_id': 'AccountID',

'copy\_email\_receipt': 'CopyEmailReceipt'

}

def \_\_init\_\_(self, credit\_card\_transaction\_settings=None, check\_transaction\_settings=None, business=None, main\_contact=None, receipt40=None, receipt80=None, print\_signature\_line=None, response=None, account\_id=None, copy\_email\_receipt=None): # noqa: E501

"""Settings - a model defined in Swagger""" # noqa: E501

self.\_credit\_card\_transaction\_settings = None

self.\_check\_transaction\_settings = None

self.\_business = None

self.\_main\_contact = None

self.\_receipt40 = None

self.\_receipt80 = None

self.\_print\_signature\_line = None

self.\_response = None

self.\_account\_id = None

self.\_copy\_email\_receipt = None

self.discriminator = None

if credit\_card\_transaction\_settings is not None:

self.credit\_card\_transaction\_settings = credit\_card\_transaction\_settings

if check\_transaction\_settings is not None:

self.check\_transaction\_settings = check\_transaction\_settings

if business is not None:

self.business = business

if main\_contact is not None:

self.main\_contact = main\_contact

if receipt40 is not None:

self.receipt40 = receipt40

if receipt80 is not None:

self.receipt80 = receipt80

if print\_signature\_line is not None:

self.print\_signature\_line = print\_signature\_line

if response is not None:

self.response = response

if account\_id is not None:

self.account\_id = account\_id

if copy\_email\_receipt is not None:

self.copy\_email\_receipt = copy\_email\_receipt

@property

def credit\_card\_transaction\_settings(self):

return self.\_credit\_card\_transaction\_settings

@credit\_card\_transaction\_settings.setter

def credit\_card\_transaction\_settings(self, credit\_card\_transaction\_settings):

self.\_credit\_card\_transaction\_settings = credit\_card\_transaction\_settings

@property

def check\_transaction\_settings(self):

return self.\_check\_transaction\_settings

@check\_transaction\_settings.setter

def check\_transaction\_settings(self, check\_transaction\_settings):

self.\_check\_transaction\_settings = check\_transaction\_settings

@property

def business(self):

return self.\_business

@business.setter

def business(self, business):

self.\_business = business

@property

def main\_contact(self):

return self.\_main\_contact

@main\_contact.setter

def main\_contact(self, main\_contact):

self.\_main\_contact = main\_contact

@property

def receipt40(self):

return self.\_receipt40

@receipt40.setter

def receipt40(self, receipt40):

self.\_receipt40 = receipt40

@property

def receipt80(self):

return self.\_receipt80

@receipt80.setter

def receipt80(self, receipt80):

self.\_receipt80 = receipt80

@property

def print\_signature\_line(self):

return self.\_print\_signature\_line

@print\_signature\_line.setter

def print\_signature\_line(self, print\_signature\_line):

self.\_print\_signature\_line = print\_signature\_line

@property

def response(self):

return self.\_response

@response.setter

def response(self, response):

self.\_response = response

@property

def account\_id(self):

return self.\_account\_id

@account\_id.setter

def account\_id(self, account\_id):

self.\_account\_id = account\_id

@property

def copy\_email\_receipt(self):

return self.\_copy\_email\_receipt

@copy\_email\_receipt.setter

def copy\_email\_receipt(self, copy\_email\_receipt):

self.\_copy\_email\_receipt = copy\_email\_receipt

# TerminalSettings

# coding: utf-8

class TerminalSettings(object):

swagger\_types = {

'id': 'int',

'terminal\_id': 'int',

'sec\_code': 'str',

'gateway\_terminal': 'bool',

'consumer\_credits': 'bool',

'dl\_required': 'bool',

'check\_verify': 'bool',

'identity\_verify': 'bool'

}

attribute\_map = {

'id': 'ID',

'terminal\_id': 'Terminal\_ID',

'sec\_code': 'SEC\_Code',

'gateway\_terminal': 'Gateway\_Terminal',

'consumer\_credits': 'Consumer\_Credits',

'dl\_required': 'DL\_Required',

'check\_verify': 'Check\_Verify',

'identity\_verify': 'Identity\_Verify'

}

def \_\_init\_\_(self, id=None, terminal\_id=None, sec\_code=None, gateway\_terminal=None, consumer\_credits=None, dl\_required=None, check\_verify=None, identity\_verify=None): # noqa: E501

"""TerminalSettings - a model defined in Swagger""" # noqa: E501

self.\_id = None

self.\_terminal\_id = None

self.\_sec\_code = None

self.\_gateway\_terminal = None

self.\_consumer\_credits = None

self.\_dl\_required = None

self.\_check\_verify = None

self.\_identity\_verify = None

self.discriminator = None

if id is not None:

self.id = id

if terminal\_id is not None:

self.terminal\_id = terminal\_id

if sec\_code is not None:

self.sec\_code = sec\_code

if gateway\_terminal is not None:

self.gateway\_terminal = gateway\_terminal

if consumer\_credits is not None:

self.consumer\_credits = consumer\_credits

if dl\_required is not None:

self.dl\_required = dl\_required

if check\_verify is not None:

self.check\_verify = check\_verify

if identity\_verify is not None:

self.identity\_verify = identity\_verify

@property

def id(self):

return self.\_id

@id.setter

def id(self, id):

self.\_id = id

@property

def terminal\_id(self):

return self.\_terminal\_id

@terminal\_id.setter

def terminal\_id(self, terminal\_id):

self.\_terminal\_id = terminal\_id

@property

def sec\_code(self):

return self.\_sec\_code

@sec\_code.setter

def sec\_code(self, sec\_code):

self.\_sec\_code = sec\_code

@property

def gateway\_terminal(self):

return self.\_gateway\_terminal

@gateway\_terminal.setter

def gateway\_terminal(self, gateway\_terminal):

self.\_gateway\_terminal = gateway\_terminal

@property

def consumer\_credits(self):

return self.\_consumer\_credits

@consumer\_credits.setter

def consumer\_credits(self, consumer\_credits):

self.\_consumer\_credits = consumer\_credits

@property

def dl\_required(self):

return self.\_dl\_required

@dl\_required.setter

def dl\_required(self, dl\_required):

self.\_dl\_required = dl\_required

@property

def check\_verify(self):

return self.\_check\_verify

@check\_verify.setter

def check\_verify(self, check\_verify):

self.\_check\_verify = check\_verify

@property

def identity\_verify(self):

return self.\_identity\_verify

@identity\_verify.setter

def identity\_verify(self, identity\_verify):

self.\_identity\_verify = identity\_verify

# Transaction

# coding: utf-8

class Transaction(object):

swagger\_types = {

'submit\_time': 'datetime',

'tran\_type': 'str',

'tran\_amt': 'float',

'tran\_tax': 'float',

'tran\_inv': 'str',

'tran\_po': 'str',

'tran\_c\_num': 'str',

'tran\_note': 'str',

'card\_num': 'str',

'card\_name': 'str',

'card\_exp': 'str',

'card\_type': 'str',

'avs\_street': 'str',

'avs\_zip': 'str',

'cvv\_num': 'str',

'proc\_id': 'str',

'proc\_resp': 'str',

'proc\_code': 'str',

'proc\_mess': 'str',

'avs\_code': 'str',

'avs\_resp': 'str',

'cvv\_code': 'str',

'cvv\_resp': 'str',

'items': 'list[TransactionItem]',

'business\_name': 'str',

'business\_address': 'str',

'business\_address2': 'str',

'business\_city': 'str',

'business\_state': 'str',

'business\_zip': 'str',

'business\_country': 'str',

'business\_website': 'str',

'contact\_first\_name': 'str',

'contact\_last\_name': 'str',

'contact\_phone': 'str',

'contact\_fax': 'str',

'contact\_email': 'str',

'contact\_title': 'str',

'ip': 'str',

'tax\_exempt': 'bool'

}

attribute\_map = {

'submit\_time': 'Submit\_Time',

'tran\_type': 'Tran\_Type',

'tran\_amt': 'Tran\_Amt',

'tran\_tax': 'Tran\_Tax',

'tran\_inv': 'Tran\_Inv',

'tran\_po': 'Tran\_PO',

'tran\_c\_num': 'Tran\_CNum',

'tran\_note': 'Tran\_Note',

'card\_num': 'Card\_Num',

'card\_name': 'Card\_Name',

'card\_exp': 'Card\_Exp',

'card\_type': 'Card\_Type',

'avs\_street': 'AVS\_Street',

'avs\_zip': 'AVS\_Zip',

'cvv\_num': 'CVV\_Num',

'proc\_id': 'Proc\_ID',

'proc\_resp': 'Proc\_Resp',

'proc\_code': 'Proc\_Code',

'proc\_mess': 'Proc\_Mess',

'avs\_code': 'AVS\_Code',

'avs\_resp': 'AVS\_Resp',

'cvv\_code': 'CVV\_Code',

'cvv\_resp': 'CVV\_Resp',

'items': 'Items',

'business\_name': 'Business\_Name',

'business\_address': 'Business\_Address',

'business\_address2': 'Business\_Address2',

'business\_city': 'Business\_City',

'business\_state': 'Business\_State',

'business\_zip': 'Business\_Zip',

'business\_country': 'Business\_Country',

'business\_website': 'Business\_Website',

'contact\_first\_name': 'Contact\_FirstName',

'contact\_last\_name': 'Contact\_LastName',

'contact\_phone': 'Contact\_Phone',

'contact\_fax': 'Contact\_Fax',

'contact\_email': 'Contact\_Email',

'contact\_title': 'Contact\_Title',

'ip': 'IP',

'tax\_exempt': 'TaxExempt'

}

def \_\_init\_\_(self, submit\_time=None, tran\_type=None, tran\_amt=None, tran\_tax=None, tran\_inv=None, tran\_po=None, tran\_c\_num=None, tran\_note=None, card\_num=None, card\_name=None, card\_exp=None, card\_type=None, avs\_street=None, avs\_zip=None, cvv\_num=None, proc\_id=None, proc\_resp=None, proc\_code=None, proc\_mess=None, avs\_code=None, avs\_resp=None, cvv\_code=None, cvv\_resp=None, items=None, business\_name=None, business\_address=None, business\_address2=None, business\_city=None, business\_state=None, business\_zip=None, business\_country=None, business\_website=None, contact\_first\_name=None, contact\_last\_name=None, contact\_phone=None, contact\_fax=None, contact\_email=None, contact\_title=None, ip=None, tax\_exempt=None): # noqa: E501

"""Transaction - a model defined in Swagger""" # noqa: E501

self.\_submit\_time = None

self.\_tran\_type = None

self.\_tran\_amt = None

self.\_tran\_tax = None

self.\_tran\_inv = None

self.\_tran\_po = None

self.\_tran\_c\_num = None

self.\_tran\_note = None

self.\_card\_num = None

self.\_card\_name = None

self.\_card\_exp = None

self.\_card\_type = None

self.\_avs\_street = None

self.\_avs\_zip = None

self.\_cvv\_num = None

self.\_proc\_id = None

self.\_proc\_resp = None

self.\_proc\_code = None

self.\_proc\_mess = None

self.\_avs\_code = None

self.\_avs\_resp = None

self.\_cvv\_code = None

self.\_cvv\_resp = None

self.\_items = None

self.\_business\_name = None

self.\_business\_address = None

self.\_business\_address2 = None

self.\_business\_city = None

self.\_business\_state = None

self.\_business\_zip = None

self.\_business\_country = None

self.\_business\_website = None

self.\_contact\_first\_name = None

self.\_contact\_last\_name = None

self.\_contact\_phone = None

self.\_contact\_fax = None

self.\_contact\_email = None

self.\_contact\_title = None

self.\_ip = None

self.\_tax\_exempt = None

self.discriminator = None

if submit\_time is not None:

self.submit\_time = submit\_time

if tran\_type is not None:

self.tran\_type = tran\_type

if tran\_amt is not None:

self.tran\_amt = tran\_amt

if tran\_tax is not None:

self.tran\_tax = tran\_tax

if tran\_inv is not None:

self.tran\_inv = tran\_inv

if tran\_po is not None:

self.tran\_po = tran\_po

if tran\_c\_num is not None:

self.tran\_c\_num = tran\_c\_num

if tran\_note is not None:

self.tran\_note = tran\_note

if card\_num is not None:

self.card\_num = card\_num

if card\_name is not None:

self.card\_name = card\_name

if card\_exp is not None:

self.card\_exp = card\_exp

if card\_type is not None:

self.card\_type = card\_type

if avs\_street is not None:

self.avs\_street = avs\_street

if avs\_zip is not None:

self.avs\_zip = avs\_zip

if cvv\_num is not None:

self.cvv\_num = cvv\_num

if proc\_id is not None:

self.proc\_id = proc\_id

if proc\_resp is not None:

self.proc\_resp = proc\_resp

if proc\_code is not None:

self.proc\_code = proc\_code

if proc\_mess is not None:

self.proc\_mess = proc\_mess

if avs\_code is not None:

self.avs\_code = avs\_code

if avs\_resp is not None:

self.avs\_resp = avs\_resp

if cvv\_code is not None:

self.cvv\_code = cvv\_code

if cvv\_resp is not None:

self.cvv\_resp = cvv\_resp

if items is not None:

self.items = items

if business\_name is not None:

self.business\_name = business\_name

if business\_address is not None:

self.business\_address = business\_address

if business\_address2 is not None:

self.business\_address2 = business\_address2

if business\_city is not None:

self.business\_city = business\_city

if business\_state is not None:

self.business\_state = business\_state

if business\_zip is not None:

self.business\_zip = business\_zip

if business\_country is not None:

self.business\_country = business\_country

if business\_website is not None:

self.business\_website = business\_website

if contact\_first\_name is not None:

self.contact\_first\_name = contact\_first\_name

if contact\_last\_name is not None:

self.contact\_last\_name = contact\_last\_name

if contact\_phone is not None:

self.contact\_phone = contact\_phone

if contact\_fax is not None:

self.contact\_fax = contact\_fax

if contact\_email is not None:

self.contact\_email = contact\_email

if contact\_title is not None:

self.contact\_title = contact\_title

if ip is not None:

self.ip = ip

if tax\_exempt is not None:

self.tax\_exempt = tax\_exempt

@property

def submit\_time(self):

return self.\_submit\_time

@submit\_time.setter

def submit\_time(self, submit\_time):

self.\_submit\_time = submit\_time

@property

def tran\_type(self):

return self.\_tran\_type

@tran\_type.setter

def tran\_type(self, tran\_type):

self.\_tran\_type = tran\_type

@property

def tran\_amt(self):

return self.\_tran\_amt

@tran\_amt.setter

def tran\_amt(self, tran\_amt):

self.\_tran\_amt = tran\_amt

@property

def tran\_tax(self):

return self.\_tran\_tax

@tran\_tax.setter

def tran\_tax(self, tran\_tax):

self.\_tran\_tax = tran\_tax

@property

def tran\_inv(self):

return self.\_tran\_inv

@tran\_inv.setter

def tran\_inv(self, tran\_inv):

self.\_tran\_inv = tran\_inv

@property

def tran\_po(self):

return self.\_tran\_po

@tran\_po.setter

def tran\_po(self, tran\_po):

self.\_tran\_po = tran\_po

@property

def tran\_c\_num(self):

return self.\_tran\_c\_num

@tran\_c\_num.setter

def tran\_c\_num(self, tran\_c\_num):

self.\_tran\_c\_num = tran\_c\_num

@property

def tran\_note(self):

return self.\_tran\_note

@tran\_note.setter

def tran\_note(self, tran\_note):

self.\_tran\_note = tran\_note

@property

def card\_num(self):

return self.\_card\_num

@card\_num.setter

def card\_num(self, card\_num):

self.\_card\_num = card\_num

@property

def card\_name(self):

return self.\_card\_name

@card\_name.setter

def card\_name(self, card\_name):

self.\_card\_name = card\_name

@property

def card\_exp(self):

return self.\_card\_exp

@card\_exp.setter

def card\_exp(self, card\_exp):

self.\_card\_exp = card\_exp

@property

def card\_type(self):

return self.\_card\_type

@card\_type.setter

def card\_type(self, card\_type):

self.\_card\_type = card\_type

@property

def avs\_street(self):

return self.\_avs\_street

@avs\_street.setter

def avs\_street(self, avs\_street):

self.\_avs\_street = avs\_street

@property

def avs\_zip(self):

return self.\_avs\_zip

@avs\_zip.setter

def avs\_zip(self, avs\_zip):

self.\_avs\_zip = avs\_zip

@property

def cvv\_num(self):

return self.\_cvv\_num

@cvv\_num.setter

def cvv\_num(self, cvv\_num):

self.\_cvv\_num = cvv\_num

@property

def proc\_id(self):

return self.\_proc\_id

@proc\_id.setter

def proc\_id(self, proc\_id):

self.\_proc\_id = proc\_id

@property

def proc\_resp(self):

return self.\_proc\_resp

@proc\_resp.setter

def proc\_resp(self, proc\_resp):

self.\_proc\_resp = proc\_resp

@property

def proc\_code(self):

return self.\_proc\_code

@proc\_code.setter

def proc\_code(self, proc\_code):

self.\_proc\_code = proc\_code

@property

def proc\_mess(self):

return self.\_proc\_mess

@proc\_mess.setter

def proc\_mess(self, proc\_mess):

self.\_proc\_mess = proc\_mess

@property

def avs\_code(self):

return self.\_avs\_code

@avs\_code.setter

def avs\_code(self, avs\_code):

self.\_avs\_code = avs\_code

@property

def avs\_resp(self):

return self.\_avs\_resp

@avs\_resp.setter

def avs\_resp(self, avs\_resp):

self.\_avs\_resp = avs\_resp

@property

def cvv\_code(self):

return self.\_cvv\_code

@cvv\_code.setter

def cvv\_code(self, cvv\_code):

self.\_cvv\_code = cvv\_code

@property

def cvv\_resp(self):

return self.\_cvv\_resp

@cvv\_resp.setter

def cvv\_resp(self, cvv\_resp):

self.\_cvv\_resp = cvv\_resp

@property

def items(self):

return self.\_items

@items.setter

def items(self, items):

self.\_items = items

@property

def business\_name(self):

return self.\_business\_name

@business\_name.setter

def business\_name(self, business\_name):

self.\_business\_name = business\_name

@property

def business\_address(self):

return self.\_business\_address

@business\_address.setter

def business\_address(self, business\_address):

self.\_business\_address = business\_address

@property

def business\_address2(self):

return self.\_business\_address2

@business\_address2.setter

def business\_address2(self, business\_address2):

self.\_business\_address2 = business\_address2

@property

def business\_city(self):

return self.\_business\_city

@business\_city.setter

def business\_city(self, business\_city):

self.\_business\_city = business\_city

@property

def business\_state(self):

return self.\_business\_state

@business\_state.setter

def business\_state(self, business\_state):

self.\_business\_state = business\_state

@property

def business\_zip(self):

return self.\_business\_zip

@business\_zip.setter

def business\_zip(self, business\_zip):

self.\_business\_zip = business\_zip

@property

def business\_country(self):

return self.\_business\_country

@business\_country.setter

def business\_country(self, business\_country):

self.\_business\_country = business\_country

@property

def business\_website(self):

return self.\_business\_website

@business\_website.setter

def business\_website(self, business\_website):

self.\_business\_website = business\_website

@property

def contact\_first\_name(self):

return self.\_contact\_first\_name

@contact\_first\_name.setter

def contact\_first\_name(self, contact\_first\_name):

self.\_contact\_first\_name = contact\_first\_name

@property

def contact\_last\_name(self):

return self.\_contact\_last\_name

@contact\_last\_name.setter

def contact\_last\_name(self, contact\_last\_name):

self.\_contact\_last\_name = contact\_last\_name

@property

def contact\_phone(self):

return self.\_contact\_phone

@contact\_phone.setter

def contact\_phone(self, contact\_phone):

self.\_contact\_phone = contact\_phone

@property

def contact\_fax(self):

return self.\_contact\_fax

@contact\_fax.setter

def contact\_fax(self, contact\_fax):

self.\_contact\_fax = contact\_fax

@property

def contact\_email(self):

return self.\_contact\_email

@contact\_email.setter

def contact\_email(self, contact\_email):

self.\_contact\_email = contact\_email

@property

def contact\_title(self):

return self.\_contact\_title

@contact\_title.setter

def contact\_title(self, contact\_title):

self.\_contact\_title = contact\_title

@property

def ip(self):

return self.\_ip

@ip.setter

def ip(self, ip):

self.\_ip = ip

@property

def tax\_exempt(self):

return self.\_tax\_exempt

@tax\_exempt.setter

def tax\_exempt(self, tax\_exempt):

self.\_tax\_exempt = tax\_exempt

# TransactionItem

# coding: utf-8

class TransactionItem(object):

swagger\_types = {

'tax\_included': 'bool',

'credit': 'bool',

'extended\_amount': 'float',

'item\_number': 'str',

'unit\_cost': 'float',

'quantity': 'int',

'description': 'str',

'discount': 'float',

'tax': 'float',

'commodity\_code': 'str',

'unit\_of\_measure': 'str'

}

attribute\_map = {

'tax\_included': 'TaxIncluded',

'credit': 'Credit',

'extended\_amount': 'ExtendedAmount',

'item\_number': 'ItemNumber',

'unit\_cost': 'UnitCost',

'quantity': 'Quantity',

'description': 'Description',

'discount': 'Discount',

'tax': 'Tax',

'commodity\_code': 'CommodityCode',

'unit\_of\_measure': 'UnitOfMeasure'

}

def \_\_init\_\_(self, tax\_included=None, credit=None, extended\_amount=None, item\_number=None, unit\_cost=None, quantity=None, description=None, discount=None, tax=None, commodity\_code=None, unit\_of\_measure=None): # noqa: E501

"""TransactionItem - a model defined in Swagger""" # noqa: E501

self.\_tax\_included = None

self.\_credit = None

self.\_extended\_amount = None

self.\_item\_number = None

self.\_unit\_cost = None

self.\_quantity = None

self.\_description = None

self.\_discount = None

self.\_tax = None

self.\_commodity\_code = None

self.\_unit\_of\_measure = None

self.discriminator = None

if tax\_included is not None:

self.tax\_included = tax\_included

if credit is not None:

self.credit = credit

if extended\_amount is not None:

self.extended\_amount = extended\_amount

if item\_number is not None:

self.item\_number = item\_number

if unit\_cost is not None:

self.unit\_cost = unit\_cost

if quantity is not None:

self.quantity = quantity

if description is not None:

self.description = description

if discount is not None:

self.discount = discount

if tax is not None:

self.tax = tax

if commodity\_code is not None:

self.commodity\_code = commodity\_code

if unit\_of\_measure is not None:

self.unit\_of\_measure = unit\_of\_measure

@property

def tax\_included(self):

return self.\_tax\_included

@tax\_included.setter

def tax\_included(self, tax\_included):

self.\_tax\_included = tax\_included

@property

def credit(self):

return self.\_credit

@credit.setter

def credit(self, credit):

self.\_credit = credit

@property

def extended\_amount(self):

return self.\_extended\_amount

@extended\_amount.setter

def extended\_amount(self, extended\_amount):

self.\_extended\_amount = extended\_amount

@property

def item\_number(self):

return self.\_item\_number

@item\_number.setter

def item\_number(self, item\_number):

self.\_item\_number = item\_number

@property

def unit\_cost(self):

return self.\_unit\_cost

@unit\_cost.setter

def unit\_cost(self, unit\_cost):

self.\_unit\_cost = unit\_cost

@property

def quantity(self):

return self.\_quantity

@quantity.setter

def quantity(self, quantity):

self.\_quantity = quantity

@property

def description(self):

return self.\_description

@description.setter

def description(self, description):

self.\_description = description

@property

def discount(self):

return self.\_discount

@discount.setter

def discount(self, discount):

self.\_discount = discount

@property

def tax(self):

return self.\_tax

@tax.setter

def tax(self, tax):

self.\_tax = tax

@property

def commodity\_code(self):

return self.\_commodity\_code

@commodity\_code.setter

def commodity\_code(self, commodity\_code):

self.\_commodity\_code = commodity\_code

@property

def unit\_of\_measure(self):

return self.\_unit\_of\_measure

@unit\_of\_measure.setter

def unit\_of\_measure(self, unit\_of\_measure):

self.\_unit\_of\_measure = unit\_of\_measure

# UpdateTransactionInfo

# coding: utf-8

class UpdateTransactionInfo(object):

swagger\_types = {

'invoice\_number': 'str',

'customer\_number': 'str',

'po\_number': 'str',

'note': 'str'

}

attribute\_map = {

'invoice\_number': 'InvoiceNumber',

'customer\_number': 'CustomerNumber',

'po\_number': 'PONumber',

'note': 'Note'

}

def \_\_init\_\_(self, invoice\_number=None, customer\_number=None, po\_number=None, note=None): # noqa: E501

"""UpdateTransactionInfo - a model defined in Swagger""" # noqa: E501

self.\_invoice\_number = None

self.\_customer\_number = None

self.\_po\_number = None

self.\_note = None

self.discriminator = None

if invoice\_number is not None:

self.invoice\_number = invoice\_number

if customer\_number is not None:

self.customer\_number = customer\_number

if po\_number is not None:

self.po\_number = po\_number

if note is not None:

self.note = note

@property

def invoice\_number(self):

return self.\_invoice\_number

@invoice\_number.setter

def invoice\_number(self, invoice\_number):

self.\_invoice\_number = invoice\_number

@property

def customer\_number(self):

return self.\_customer\_number

@customer\_number.setter

def customer\_number(self, customer\_number):

self.\_customer\_number = customer\_number

@property

def po\_number(self):

return self.\_po\_number

@po\_number.setter

def po\_number(self, po\_number):

self.\_po\_number = po\_number

@property

def note(self):

return self.\_note

@note.setter

def note(self, note):

self.\_note = note

# VaultCheck

# coding: utf-8

class VaultCheck(object):

swagger\_types = {

'response': 'Response',

'check\_id': 'int',

'main': 'bool',

'check\_routing': 'str',

'check\_account': 'str',

'check\_number': 'str',

'check\_first\_name': 'str',

'check\_last\_name': 'str',

'note': 'str',

'masked\_routing': 'str',

'masked\_account': 'str',

'account\_type': 'str'

}

attribute\_map = {

'response': 'Response',

'check\_id': 'CheckID',

'main': 'Main',

'check\_routing': 'CheckRouting',

'check\_account': 'CheckAccount',

'check\_number': 'CheckNumber',

'check\_first\_name': 'CheckFirstName',

'check\_last\_name': 'CheckLastName',

'note': 'Note',

'masked\_routing': 'MaskedRouting',

'masked\_account': 'MaskedAccount',

'account\_type': 'AccountType'

}

def \_\_init\_\_(self, response=None, check\_id=None, main=None, check\_routing=None, check\_account=None, check\_number=None, check\_first\_name=None, check\_last\_name=None, note=None, masked\_routing=None, masked\_account=None, account\_type=None): # noqa: E501

"""VaultCheck - a model defined in Swagger""" # noqa: E501

self.\_response = None

self.\_check\_id = None

self.\_main = None

self.\_check\_routing = None

self.\_check\_account = None

self.\_check\_number = None

self.\_check\_first\_name = None

self.\_check\_last\_name = None

self.\_note = None

self.\_masked\_routing = None

self.\_masked\_account = None

self.\_account\_type = None

self.discriminator = None

if response is not None:

self.response = response

if check\_id is not None:

self.check\_id = check\_id

if main is not None:

self.main = main

if check\_routing is not None:

self.check\_routing = check\_routing

if check\_account is not None:

self.check\_account = check\_account

if check\_number is not None:

self.check\_number = check\_number

if check\_first\_name is not None:

self.check\_first\_name = check\_first\_name

if check\_last\_name is not None:

self.check\_last\_name = check\_last\_name

if note is not None:

self.note = note

if masked\_routing is not None:

self.masked\_routing = masked\_routing

if masked\_account is not None:

self.masked\_account = masked\_account

if account\_type is not None:

self.account\_type = account\_type

@property

def response(self):

return self.\_response

@response.setter

def response(self, response):

self.\_response = response

@property

def check\_id(self):

return self.\_check\_id

@check\_id.setter

def check\_id(self, check\_id):

self.\_check\_id = check\_id

@property

def main(self):

return self.\_main

@main.setter

def main(self, main):

self.\_main = main

@property

def check\_routing(self):

return self.\_check\_routing

@check\_routing.setter

def check\_routing(self, check\_routing):

self.\_check\_routing = check\_routing

@property

def check\_account(self):

return self.\_check\_account

@check\_account.setter

def check\_account(self, check\_account):

self.\_check\_account = check\_account

@property

def check\_number(self):

return self.\_check\_number

@check\_number.setter

def check\_number(self, check\_number):

self.\_check\_number = check\_number

@property

def check\_first\_name(self):

return self.\_check\_first\_name

@check\_first\_name.setter

def check\_first\_name(self, check\_first\_name):

self.\_check\_first\_name = check\_first\_name

@property

def check\_last\_name(self):

return self.\_check\_last\_name

@check\_last\_name.setter

def check\_last\_name(self, check\_last\_name):

self.\_check\_last\_name = check\_last\_name

@property

def note(self):

return self.\_note

@note.setter

def note(self, note):

self.\_note = note

@property

def masked\_routing(self):

return self.\_masked\_routing

@masked\_routing.setter

def masked\_routing(self, masked\_routing):

self.\_masked\_routing = masked\_routing

@property

def masked\_account(self):

return self.\_masked\_account

@masked\_account.setter

def masked\_account(self, masked\_account):

self.\_masked\_account = masked\_account

@property

def account\_type(self):

return self.\_account\_type

@account\_type.setter

def account\_type(self, account\_type):

self.\_account\_type = account\_type

# VaultCreditCard

# coding: utf-8

class VaultCreditCard(object):

swagger\_types = {

'magnetic\_tracks': 'str',

'track1': 'str',

'track2': 'str',

'number': 'str',

'masked\_number': 'str',

'holder': 'str',

'exp\_date': 'str',

'cvv': 'str',

'address': 'str',

'city': 'str',

'state': 'str',

'zip': 'str',

'swiped': 'bool',

'type\_string': 'str',

'cvv\_type': 'CVVPresenceType',

'not\_present': 'bool',

'response': 'Response',

'card\_id': 'int',

'note': 'str',

'main': 'bool'

}

attribute\_map = {

'magnetic\_tracks': 'MagneticTracks',

'track1': 'Track1',

'track2': 'Track2',

'number': 'Number',

'masked\_number': 'MaskedNumber',

'holder': 'Holder',

'exp\_date': 'ExpDate',

'cvv': 'CVV',

'address': 'Address',

'city': 'City',

'state': 'State',

'zip': 'ZIP',

'swiped': 'Swiped',

'type\_string': 'TypeString',

'cvv\_type': 'CVVType',

'not\_present': 'NotPresent',

'response': 'Response',

'card\_id': 'CardID',

'note': 'Note',

'main': 'Main'

}

def \_\_init\_\_(self, magnetic\_tracks=None, track1=None, track2=None, number=None, masked\_number=None, holder=None, exp\_date=None, cvv=None, address=None, city=None, state=None, zip=None, swiped=None, type\_string=None, cvv\_type=None, not\_present=None, response=None, card\_id=None, note=None, main=None): # noqa: E501

"""VaultCreditCard - a model defined in Swagger""" # noqa: E501

self.\_magnetic\_tracks = None

self.\_track1 = None

self.\_track2 = None

self.\_number = None

self.\_masked\_number = None

self.\_holder = None

self.\_exp\_date = None

self.\_cvv = None

self.\_address = None

self.\_city = None

self.\_state = None

self.\_zip = None

self.\_swiped = None

self.\_type\_string = None

self.\_cvv\_type = None

self.\_not\_present = None

self.\_response = None

self.\_card\_id = None

self.\_note = None

self.\_main = None

self.discriminator = None

if magnetic\_tracks is not None:

self.magnetic\_tracks = magnetic\_tracks

if track1 is not None:

self.track1 = track1

if track2 is not None:

self.track2 = track2

if number is not None:

self.number = number

if masked\_number is not None:

self.masked\_number = masked\_number

if holder is not None:

self.holder = holder

if exp\_date is not None:

self.exp\_date = exp\_date

if cvv is not None:

self.cvv = cvv

if address is not None:

self.address = address

if city is not None:

self.city = city

if state is not None:

self.state = state

if zip is not None:

self.zip = zip

if swiped is not None:

self.swiped = swiped

if type\_string is not None:

self.type\_string = type\_string

if cvv\_type is not None:

self.cvv\_type = cvv\_type

if not\_present is not None:

self.not\_present = not\_present

if response is not None:

self.response = response

if card\_id is not None:

self.card\_id = card\_id

if note is not None:

self.note = note

if main is not None:

self.main = main

@property

def magnetic\_tracks(self):

return self.\_magnetic\_tracks

@magnetic\_tracks.setter

def magnetic\_tracks(self, magnetic\_tracks):

self.\_magnetic\_tracks = magnetic\_tracks

@property

def track1(self):

return self.\_track1

@track1.setter

def track1(self, track1):

self.\_track1 = track1

@property

def track2(self):

return self.\_track2

@track2.setter

def track2(self, track2):

self.\_track2 = track2

@property

def number(self):

return self.\_number

@number.setter

def number(self, number):

self.\_number = number

@property

def masked\_number(self):

return self.\_masked\_number

@masked\_number.setter

def masked\_number(self, masked\_number):

self.\_masked\_number = masked\_number

@property

def holder(self):

return self.\_holder

@holder.setter

def holder(self, holder):

self.\_holder = holder

@property

def exp\_date(self):

return self.\_exp\_date

@exp\_date.setter

def exp\_date(self, exp\_date):

self.\_exp\_date = exp\_date

@property

def cvv(self):

return self.\_cvv

@cvv.setter

def cvv(self, cvv):

self.\_cvv = cvv

@property

def address(self):

return self.\_address

@address.setter

def address(self, address):

self.\_address = address

@property

def city(self):

return self.\_city

@city.setter

def city(self, city):

self.\_city = city

@property

def state(self):

return self.\_state

@state.setter

def state(self, state):

self.\_state = state

@property

def zip(self):

return self.\_zip

@zip.setter

def zip(self, zip):

self.\_zip = zip

@property

def swiped(self):

return self.\_swiped

@swiped.setter

def swiped(self, swiped):

self.\_swiped = swiped

@property

def type\_string(self):

return self.\_type\_string

@type\_string.setter

def type\_string(self, type\_string):

self.\_type\_string = type\_string

@property

def cvv\_type(self):

return self.\_cvv\_type

@cvv\_type.setter

def cvv\_type(self, cvv\_type):

self.\_cvv\_type = cvv\_type

@property

def not\_present(self):

return self.\_not\_present

@not\_present.setter

def not\_present(self, not\_present):

self.\_not\_present = not\_present

@property

def response(self):

return self.\_response

@response.setter

def response(self, response):

self.\_response = response

@property

def card\_id(self):

return self.\_card\_id

@card\_id.setter

def card\_id(self, card\_id):

self.\_card\_id = card\_id

@property

def note(self):

return self.\_note

@note.setter

def note(self, note):

self.\_note = note

@property

def main(self):

return self.\_main

@main.setter

def main(self, main):

self.\_main = main