C# Models

# Address

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class Address : IEquatable<Address>, IValidatableObject

{

public Address(string addressLine1 = default(string), string addressLine2 = default(string), string city = default(string), string state = default(string), string zipCode = default(string))

{

this.AddressLine1 = addressLine1;

this.AddressLine2 = addressLine2;

this.City = city;

this.State = state;

this.ZipCode = zipCode;

}

[DataMember(Name="AddressLine1", EmitDefaultValue=false)]

public string AddressLine1 { get; set; }

[DataMember(Name="AddressLine2", EmitDefaultValue=false)]

public string AddressLine2 { get; set; }

[DataMember(Name="City", EmitDefaultValue=false)]

public string City { get; set; }

[DataMember(Name="State", EmitDefaultValue=false)]

public string State { get; set; }

[DataMember(Name="ZipCode", EmitDefaultValue=false)]

public string ZipCode { get; set; }

}

}

# Batch

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class Batch : IEquatable<Batch>, IValidatableObject

{

public Batch(string batchNumber = default(string), DateTime? started = default(DateTime?), DateTime? completed = default(DateTime?), string salesCount = default(string), double? salesSum = default(double?), string returnCount = default(string), double? returnSum = default(double?), bool? batchError = default(bool?))

{

this.BatchNumber = batchNumber;

this.Started = started;

this.Completed = completed;

this.SalesCount = salesCount;

this.SalesSum = salesSum;

this.ReturnCount = returnCount;

this.ReturnSum = returnSum;

this.BatchError = batchError;

}

[DataMember(Name="BatchNumber", EmitDefaultValue=false)]

public string BatchNumber { get; set; }

[DataMember(Name="Started", EmitDefaultValue=false)]

public DateTime? Started { get; set; }

[DataMember(Name="Completed", EmitDefaultValue=false)]

public DateTime? Completed { get; set; }

[DataMember(Name="SalesCount", EmitDefaultValue=false)]

public string SalesCount { get; set; }

[DataMember(Name="SalesSum", EmitDefaultValue=false)]

public double? SalesSum { get; set; }

[DataMember(Name="ReturnCount", EmitDefaultValue=false)]

public string ReturnCount { get; set; }

[DataMember(Name="ReturnSum", EmitDefaultValue=false)]

public double? ReturnSum { get; set; }

[DataMember(Name="BatchError", EmitDefaultValue=false)]

public bool? BatchError { get; set; }

}

}

# Business

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class Business : IEquatable<Business>, IValidatableObject

{

public Business(Address address = default(Address), string name = default(string), string phone = default(string), string fax = default(string), string webSite = default(string))

{

this.Address = address;

this.Name = name;

this.Phone = phone;

this.Fax = fax;

this.WebSite = webSite;

}

[DataMember(Name="Address", EmitDefaultValue=false)]

public Address Address { get; set; }

[DataMember(Name="Name", EmitDefaultValue=false)]

public string Name { get; set; }

[DataMember(Name="Phone", EmitDefaultValue=false)]

public string Phone { get; set; }

[DataMember(Name="Fax", EmitDefaultValue=false)]

public string Fax { get; set; }

[DataMember(Name="WebSite", EmitDefaultValue=false)]

public string WebSite { get; set; }

}

}

# CheckAccountType

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[JsonConverter(typeof(StringEnumConverter))]

public enum CheckAccountType

{

[EnumMember(Value = "Unknown")]

Unknown = 1,

[EnumMember(Value = "Checking")]

Checking = 2,

[EnumMember(Value = "Savings")]

Savings = 3

}

}

# CheckTransaction

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class CheckTransaction : IEquatable<CheckTransaction>, IValidatableObject

{

public CheckTransaction(string terminalID = default(string), string accountNumber = default(string), CheckAccountType accountType = default(CheckAccountType), string checkNumber = default(string), string routingNumber = default(string), string mICRData = default(string), string firstName = default(string), string lastName = default(string), string companyName = default(string), string address1 = default(string), string address2 = default(string), string city = default(string), string state = default(string), string zip = default(string), string phoneNumber = default(string), string dLNumber = default(string), string dLState = default(string), string courtesyCardID = default(string), string sSN4 = default(string), string dOBYear = default(string), string checkAmount = default(string), CheckTransactionType tranType = default(CheckTransactionType), string procID = default(string), string iPAddress = default(string), string customerNumber = default(string), string pONumber = default(string), string invoiceNumber = default(string), string tranNote = default(string), string emailAddress = default(string), string sECCode = default(string), string procResp = default(string), string procCode = default(string), string procMess = default(string))

{

this.TerminalID = terminalID;

this.AccountNumber = accountNumber;

this.AccountType = accountType;

this.CheckNumber = checkNumber;

this.RoutingNumber = routingNumber;

this.MICRData = mICRData;

this.FirstName = firstName;

this.LastName = lastName;

this.CompanyName = companyName;

this.Address1 = address1;

this.Address2 = address2;

this.City = city;

this.State = state;

this.Zip = zip;

this.PhoneNumber = phoneNumber;

this.DLNumber = dLNumber;

this.DLState = dLState;

this.CourtesyCardID = courtesyCardID;

this.SSN4 = sSN4;

this.DOBYear = dOBYear;

this.CheckAmount = checkAmount;

this.TranType = tranType;

this.ProcID = procID;

this.IPAddress = iPAddress;

this.CustomerNumber = customerNumber;

this.PONumber = pONumber;

this.InvoiceNumber = invoiceNumber;

this.TranNote = tranNote;

this.EmailAddress = emailAddress;

this.SECCode = sECCode;

this.ProcResp = procResp;

this.ProcCode = procCode;

this.ProcMess = procMess;

}

[DataMember(Name="Terminal\_ID", EmitDefaultValue=false)]

public string TerminalID { get; set; }

[DataMember(Name="Account\_Number", EmitDefaultValue=false)]

public string AccountNumber { get; set; }

[DataMember(Name="Account\_Type", EmitDefaultValue=false)]

public CheckAccountType AccountType { get; set; }

[DataMember(Name="Check\_Number", EmitDefaultValue=false)]

public string CheckNumber { get; set; }

[DataMember(Name="Routing\_Number", EmitDefaultValue=false)]

public string RoutingNumber { get; set; }

[DataMember(Name="MICR\_Data", EmitDefaultValue=false)]

public string MICRData { get; set; }

[DataMember(Name="First\_Name", EmitDefaultValue=false)]

public string FirstName { get; set; }

[DataMember(Name="Last\_Name", EmitDefaultValue=false)]

public string LastName { get; set; }

[DataMember(Name="Company\_Name", EmitDefaultValue=false)]

public string CompanyName { get; set; }

[DataMember(Name="Address1", EmitDefaultValue=false)]

public string Address1 { get; set; }

[DataMember(Name="Address2", EmitDefaultValue=false)]

public string Address2 { get; set; }

[DataMember(Name="City", EmitDefaultValue=false)]

public string City { get; set; }

[DataMember(Name="State", EmitDefaultValue=false)]

public string State { get; set; }

[DataMember(Name="Zip", EmitDefaultValue=false)]

public string Zip { get; set; }

[DataMember(Name="Phone\_Number", EmitDefaultValue=false)]

public string PhoneNumber { get; set; }

[DataMember(Name="DL\_Number", EmitDefaultValue=false)]

public string DLNumber { get; set; }

[DataMember(Name="DL\_State", EmitDefaultValue=false)]

public string DLState { get; set; }

[DataMember(Name="Courtesy\_Card\_ID", EmitDefaultValue=false)]

public string CourtesyCardID { get; set; }

[DataMember(Name="SSN4", EmitDefaultValue=false)]

public string SSN4 { get; set; }

[DataMember(Name="DOB\_Year", EmitDefaultValue=false)]

public string DOBYear { get; set; }

[DataMember(Name="Check\_Amount", EmitDefaultValue=false)]

public string CheckAmount { get; set; }

[DataMember(Name="Tran\_Type", EmitDefaultValue=false)]

public CheckTransactionType TranType { get; set; }

[DataMember(Name="Proc\_ID", EmitDefaultValue=false)]

public string ProcID { get; set; }

[DataMember(Name="IP\_Address", EmitDefaultValue=false)]

public string IPAddress { get; set; }

[DataMember(Name="CustomerNumber", EmitDefaultValue=false)]

public string CustomerNumber { get; set; }

[DataMember(Name="PONumber", EmitDefaultValue=false)]

public string PONumber { get; set; }

[DataMember(Name="InvoiceNumber", EmitDefaultValue=false)]

public string InvoiceNumber { get; set; }

[DataMember(Name="TranNote", EmitDefaultValue=false)]

public string TranNote { get; set; }

[DataMember(Name="EmailAddress", EmitDefaultValue=false)]

public string EmailAddress { get; set; }

[DataMember(Name="SEC\_Code", EmitDefaultValue=false)]

public string SECCode { get; set; }

[DataMember(Name="Proc\_Resp", EmitDefaultValue=false)]

public string ProcResp { get; set; }

[DataMember(Name="Proc\_Code", EmitDefaultValue=false)]

public string ProcCode { get; set; }

[DataMember(Name="Proc\_Mess", EmitDefaultValue=false)]

public string ProcMess { get; set; }

}

}

# CheckTransactionSettings

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class CheckTransactionSettings : IEquatable<CheckTransactionSettings>, IValidatableObject

{

public CheckTransactionSettings(FieldInfo reader = default(FieldInfo), FieldInfo contactName = default(FieldInfo), FieldInfo contactTitle = default(FieldInfo), FieldInfo emailAddress = default(FieldInfo), string phoneDefault = default(string), FieldInfo fax = default(FieldInfo), string billingAddressDefault = default(string), FieldInfo billingAddress2 = default(FieldInfo), string billingCityDefault = default(string), string billingStateDefault = default(string), string billingZipDefault = default(string), FieldInfo pONumber = default(FieldInfo), FieldInfo invoiceNumber = default(FieldInfo), FieldInfo businessName = default(FieldInfo), FieldInfo customerNumber = default(FieldInfo), FieldInfo referenceNote = default(FieldInfo), string defaultSECCode = default(string))

{

this.Reader = reader;

this.ContactName = contactName;

this.ContactTitle = contactTitle;

this.EmailAddress = emailAddress;

this.PhoneDefault = phoneDefault;

this.Fax = fax;

this.BillingAddressDefault = billingAddressDefault;

this.BillingAddress2 = billingAddress2;

this.BillingCityDefault = billingCityDefault;

this.BillingStateDefault = billingStateDefault;

this.BillingZipDefault = billingZipDefault;

this.PONumber = pONumber;

this.InvoiceNumber = invoiceNumber;

this.BusinessName = businessName;

this.CustomerNumber = customerNumber;

this.ReferenceNote = referenceNote;

this.DefaultSECCode = defaultSECCode;

}

[DataMember(Name="Reader", EmitDefaultValue=false)]

public FieldInfo Reader { get; set; }

[DataMember(Name="ContactName", EmitDefaultValue=false)]

public FieldInfo ContactName { get; set; }

[DataMember(Name="ContactTitle", EmitDefaultValue=false)]

public FieldInfo ContactTitle { get; set; }

[DataMember(Name="EmailAddress", EmitDefaultValue=false)]

public FieldInfo EmailAddress { get; set; }

[DataMember(Name="PhoneDefault", EmitDefaultValue=false)]

public string PhoneDefault { get; set; }

[DataMember(Name="Fax", EmitDefaultValue=false)]

public FieldInfo Fax { get; set; }

[DataMember(Name="BillingAddressDefault", EmitDefaultValue=false)]

public string BillingAddressDefault { get; set; }

[DataMember(Name="BillingAddress2", EmitDefaultValue=false)]

public FieldInfo BillingAddress2 { get; set; }

[DataMember(Name="BillingCityDefault", EmitDefaultValue=false)]

public string BillingCityDefault { get; set; }

[DataMember(Name="BillingStateDefault", EmitDefaultValue=false)]

public string BillingStateDefault { get; set; }

[DataMember(Name="BillingZipDefault", EmitDefaultValue=false)]

public string BillingZipDefault { get; set; }

[DataMember(Name="PONumber", EmitDefaultValue=false)]

public FieldInfo PONumber { get; set; }

[DataMember(Name="InvoiceNumber", EmitDefaultValue=false)]

public FieldInfo InvoiceNumber { get; set; }

[DataMember(Name="BusinessName", EmitDefaultValue=false)]

public FieldInfo BusinessName { get; set; }

[DataMember(Name="CustomerNumber", EmitDefaultValue=false)]

public FieldInfo CustomerNumber { get; set; }

[DataMember(Name="ReferenceNote", EmitDefaultValue=false)]

public FieldInfo ReferenceNote { get; set; }

[DataMember(Name="DefaultSECCode", EmitDefaultValue=false)]

public string DefaultSECCode { get; set; }

}

}

# CheckTransactionType

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[JsonConverter(typeof(StringEnumConverter))]

public enum CheckTransactionType

{

[EnumMember(Value = "Unknown")]

Unknown = 1,

[EnumMember(Value = "Authorize")]

Authorize = 2,

[EnumMember(Value = "Void")]

Void = 3,

[EnumMember(Value = "Override")]

Override = 4,

[EnumMember(Value = "Payroll")]

Payroll = 5,

[EnumMember(Value = "Recurring")]

Recurring = 6,

[EnumMember(Value = "Refund")]

Refund = 7 }

}

# Contact

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class Contact : IEquatable<Contact>, IValidatableObject

{

public Contact(string firstName = default(string), string lastName = default(string), string name = default(string), string title = default(string), string phone = default(string), string email = default(string))

{

this.FirstName = firstName;

this.LastName = lastName;

this.Name = name;

this.Title = title;

this.Phone = phone;

this.Email = email;

}

[DataMember(Name="FirstName", EmitDefaultValue=false)]

public string FirstName { get; set; }

[DataMember(Name="LastName", EmitDefaultValue=false)]

public string LastName { get; set; }

[DataMember(Name="Name", EmitDefaultValue=false)]

public string Name { get; set; }

[DataMember(Name="Title", EmitDefaultValue=false)]

public string Title { get; set; }

[DataMember(Name="Phone", EmitDefaultValue=false)]

public string Phone { get; set; }

[DataMember(Name="Email", EmitDefaultValue=false)]

public string Email { get; set; }

}

}

# CreditCardTransactionSettings

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class CreditCardTransactionSettings : IEquatable<CreditCardTransactionSettings>, IValidatableObject

{

public CreditCardTransactionSettings(FieldInfo cVV = default(FieldInfo), FieldInfo contactName = default(FieldInfo), FieldInfo contactTitle = default(FieldInfo), FieldInfo emailAddress = default(FieldInfo), FieldInfo phone = default(FieldInfo), FieldInfo fax = default(FieldInfo), FieldInfo billingAddress = default(FieldInfo), FieldInfo billingAddress2 = default(FieldInfo), FieldInfo billingCity = default(FieldInfo), FieldInfo billingState = default(FieldInfo), FieldInfo billingZip = default(FieldInfo), FieldInfo pONumber = default(FieldInfo), FieldInfo invoiceNumber = default(FieldInfo), FieldInfo businessName = default(FieldInfo), FieldInfo customerNumber = default(FieldInfo), FieldInfo referenceNote = default(FieldInfo), FieldInfo salesTax = default(FieldInfo), FieldInfo taxExempt = default(FieldInfo), FieldInfo cardTracks = default(FieldInfo), FieldInfo cardZip = default(FieldInfo), FieldInfo cardAddress = default(FieldInfo), string itemDescription = default(string), TransactionType defaultTransactionType = default(TransactionType))

{

this.CVV = cVV;

this.ContactName = contactName;

this.ContactTitle = contactTitle;

this.EmailAddress = emailAddress;

this.Phone = phone;

this.Fax = fax;

this.BillingAddress = billingAddress;

this.BillingAddress2 = billingAddress2;

this.BillingCity = billingCity;

this.BillingState = billingState;

this.BillingZip = billingZip;

this.PONumber = pONumber;

this.InvoiceNumber = invoiceNumber;

this.BusinessName = businessName;

this.CustomerNumber = customerNumber;

this.ReferenceNote = referenceNote;

this.SalesTax = salesTax;

this.TaxExempt = taxExempt;

this.CardTracks = cardTracks;

this.CardZip = cardZip;

this.CardAddress = cardAddress;

this.ItemDescription = itemDescription;

this.DefaultTransactionType = defaultTransactionType;

}

[DataMember(Name="CVV", EmitDefaultValue=false)]

public FieldInfo CVV { get; set; }

[DataMember(Name="ContactName", EmitDefaultValue=false)]

public FieldInfo ContactName { get; set; }

[DataMember(Name="ContactTitle", EmitDefaultValue=false)]

public FieldInfo ContactTitle { get; set; }

[DataMember(Name="EmailAddress", EmitDefaultValue=false)]

public FieldInfo EmailAddress { get; set; }

[DataMember(Name="Phone", EmitDefaultValue=false)]

public FieldInfo Phone { get; set; }

[DataMember(Name="Fax", EmitDefaultValue=false)]

public FieldInfo Fax { get; set; }

[DataMember(Name="BillingAddress", EmitDefaultValue=false)]

public FieldInfo BillingAddress { get; set; }

[DataMember(Name="BillingAddress2", EmitDefaultValue=false)]

public FieldInfo BillingAddress2 { get; set; }

[DataMember(Name="BillingCity", EmitDefaultValue=false)]

public FieldInfo BillingCity { get; set; }

[DataMember(Name="BillingState", EmitDefaultValue=false)]

public FieldInfo BillingState { get; set; }

[DataMember(Name="BillingZip", EmitDefaultValue=false)]

public FieldInfo BillingZip { get; set; }

[DataMember(Name="PONumber", EmitDefaultValue=false)]

public FieldInfo PONumber { get; set; }

[DataMember(Name="InvoiceNumber", EmitDefaultValue=false)]

public FieldInfo InvoiceNumber { get; set; }

[DataMember(Name="BusinessName", EmitDefaultValue=false)]

public FieldInfo BusinessName { get; set; }

[DataMember(Name="CustomerNumber", EmitDefaultValue=false)]

public FieldInfo CustomerNumber { get; set; }

[DataMember(Name="ReferenceNote", EmitDefaultValue=false)]

public FieldInfo ReferenceNote { get; set; }

[DataMember(Name="SalesTax", EmitDefaultValue=false)]

public FieldInfo SalesTax { get; set; }

[DataMember(Name="TaxExempt", EmitDefaultValue=false)]

public FieldInfo TaxExempt { get; set; }

[DataMember(Name="CardTracks", EmitDefaultValue=false)]

public FieldInfo CardTracks { get; set; }

[DataMember(Name="CardZip", EmitDefaultValue=false)]

public FieldInfo CardZip { get; set; }

[DataMember(Name="CardAddress", EmitDefaultValue=false)]

public FieldInfo CardAddress { get; set; }

[DataMember(Name="ItemDescription", EmitDefaultValue=false)]

public string ItemDescription { get; set; }

[DataMember(Name="DefaultTransactionType", EmitDefaultValue=false)]

public TransactionType DefaultTransactionType { get; set; }

}

}

# CVVPresenceType

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[JsonConverter(typeof(StringEnumConverter))]

public enum CVVPresenceType

{

[EnumMember(Value = "Normal")]

Normal = 1,

[EnumMember(Value = "Bypass")]

Bypass = 2,

[EnumMember(Value = "Illegible")]

Illegible = 3,

[EnumMember(Value = "NotAvailable")]

NotAvailable = 4

}

}

# EmailReceipt

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class EmailReceipt : IEquatable<EmailReceipt>, IValidatableObject

{

public EmailReceipt(string transactionID = default(string), string toAddresses = default(string), string subject = default(string), string header = default(string), string footer = default(string))

{

this.TransactionID = transactionID;

this.ToAddresses = toAddresses;

this.Subject = subject;

this.Header = header;

this.Footer = footer;

}

[DataMember(Name="TransactionID", EmitDefaultValue=false)]

public string TransactionID { get; set; }

[DataMember(Name="ToAddresses", EmitDefaultValue=false)]

public string ToAddresses { get; set; }

[DataMember(Name="Subject", EmitDefaultValue=false)]

public string Subject { get; set; }

[DataMember(Name="Header", EmitDefaultValue=false)]

public string Header { get; set; }

[DataMember(Name="Footer", EmitDefaultValue=false)]

public string Footer { get; set; }

}

}

# FieldAccess

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[JsonConverter(typeof(StringEnumConverter))]

public enum FieldAccess

{

[EnumMember(Value = "Disabled")]

Disabled = 1,

[EnumMember(Value = "Enabled")]

Enabled = 2,

[EnumMember(Value = "Required")]

Required = 3

}

}

# FieldInfo

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class FieldInfo : IEquatable<FieldInfo>, IValidatableObject

{

public FieldInfo(FieldAccess access = default(FieldAccess), string \_default = default(string))

{

this.Access = access;

this.\_Default = \_default;

}

[DataMember(Name="Access", EmitDefaultValue=false)]

public FieldAccess Access { get; set; }

[DataMember(Name="Default", EmitDefaultValue=false)]

public string \_Default { get; set; }

}

}

# QueryTransaction

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class QueryTransaction : IEquatable<QueryTransaction>, IValidatableObject

{

public QueryTransaction(string tranSequence = default(string), string tranBatch = default(string), string tranModified = default(string), string tranProcessed = default(string), string tranSettled = default(string), string submitTime = default(string), string tranType = default(string), double? tranAmt = default(double?), double? tranTax = default(double?), string tranInv = default(string), string tranPO = default(string), string tranCNum = default(string), string tranNote = default(string), string cardNum = default(string), string cardName = default(string), string cardExp = default(string), string aVSStreet = default(string), string aVSZip = default(string), string cVVNum = default(string), string procID = default(string), string procResp = default(string), string procCode = default(string), string procMess = default(string), string aVSCode = default(string), string aVSResp = default(string), string cVVCode = default(string), string cVVResp = default(string), List<TransactionItem> items = default(List<TransactionItem>), string businessName = default(string), string businessAddress = default(string), string businessAddress2 = default(string), string businessCity = default(string), string businessState = default(string), string businessZip = default(string), string businessCountry = default(string), string businessWebsite = default(string), string contactFirstName = default(string), string contactLastName = default(string), string contactPhone = default(string), string contactFax = default(string), string contactEmail = default(string), string contactTitle = default(string), string iP = default(string), bool? taxExempt = default(bool?))

{

this.TranSequence = tranSequence;

this.TranBatch = tranBatch;

this.TranModified = tranModified;

this.TranProcessed = tranProcessed;

this.TranSettled = tranSettled;

this.SubmitTime = submitTime;

this.TranType = tranType;

this.TranAmt = tranAmt;

this.TranTax = tranTax;

this.TranInv = tranInv;

this.TranPO = tranPO;

this.TranCNum = tranCNum;

this.TranNote = tranNote;

this.CardNum = cardNum;

this.CardName = cardName;

this.CardExp = cardExp;

this.AVSStreet = aVSStreet;

this.AVSZip = aVSZip;

this.CVVNum = cVVNum;

this.ProcID = procID;

this.ProcResp = procResp;

this.ProcCode = procCode;

this.ProcMess = procMess;

this.AVSCode = aVSCode;

this.AVSResp = aVSResp;

this.CVVCode = cVVCode;

this.CVVResp = cVVResp;

this.Items = items;

this.BusinessName = businessName;

this.BusinessAddress = businessAddress;

this.BusinessAddress2 = businessAddress2;

this.BusinessCity = businessCity;

this.BusinessState = businessState;

this.BusinessZip = businessZip;

this.BusinessCountry = businessCountry;

this.BusinessWebsite = businessWebsite;

this.ContactFirstName = contactFirstName;

this.ContactLastName = contactLastName;

this.ContactPhone = contactPhone;

this.ContactFax = contactFax;

this.ContactEmail = contactEmail;

this.ContactTitle = contactTitle;

this.IP = iP;

this.TaxExempt = taxExempt;

}

[DataMember(Name="Tran\_Sequence", EmitDefaultValue=false)]

public string TranSequence { get; set; }

[DataMember(Name="Tran\_Batch", EmitDefaultValue=false)]

public string TranBatch { get; set; }

[DataMember(Name="Tran\_Modified", EmitDefaultValue=false)]

public string TranModified { get; set; }

[DataMember(Name="Tran\_Processed", EmitDefaultValue=false)]

public string TranProcessed { get; set; }

[DataMember(Name="Tran\_Settled", EmitDefaultValue=false)]

public string TranSettled { get; set; }

[DataMember(Name="Submit\_Time", EmitDefaultValue=false)]

public string SubmitTime { get; set; }

[DataMember(Name="Tran\_Type", EmitDefaultValue=false)]

public string TranType { get; set; }

[DataMember(Name="Tran\_Amt", EmitDefaultValue=false)]

public double? TranAmt { get; set; }

[DataMember(Name="Tran\_Tax", EmitDefaultValue=false)]

public double? TranTax { get; set; }

[DataMember(Name="Tran\_Inv", EmitDefaultValue=false)]

public string TranInv { get; set; }

[DataMember(Name="Tran\_PO", EmitDefaultValue=false)]

public string TranPO { get; set; }

[DataMember(Name="Tran\_CNum", EmitDefaultValue=false)]

public string TranCNum { get; set; }

[DataMember(Name="Tran\_Note", EmitDefaultValue=false)]

public string TranNote { get; set; }

[DataMember(Name="Card\_Num", EmitDefaultValue=false)]

public string CardNum { get; set; }

[DataMember(Name="Card\_Name", EmitDefaultValue=false)]

public string CardName { get; set; }

[DataMember(Name="Card\_Exp", EmitDefaultValue=false)]

public string CardExp { get; set; }

[DataMember(Name="AVS\_Street", EmitDefaultValue=false)]

public string AVSStreet { get; set; }

[DataMember(Name="AVS\_Zip", EmitDefaultValue=false)]

public string AVSZip { get; set; }

[DataMember(Name="CVV\_Num", EmitDefaultValue=false)]

public string CVVNum { get; set; }

[DataMember(Name="Proc\_ID", EmitDefaultValue=false)]

public string ProcID { get; set; }

[DataMember(Name="Proc\_Resp", EmitDefaultValue=false)]

public string ProcResp { get; set; }

[DataMember(Name="Proc\_Code", EmitDefaultValue=false)]

public string ProcCode { get; set; }

[DataMember(Name="Proc\_Mess", EmitDefaultValue=false)]

public string ProcMess { get; set; }

[DataMember(Name="AVS\_Code", EmitDefaultValue=false)]

public string AVSCode { get; set; }

[DataMember(Name="AVS\_Resp", EmitDefaultValue=false)]

public string AVSResp { get; set; }

[DataMember(Name="CVV\_Code", EmitDefaultValue=false)]

public string CVVCode { get; set; }

[DataMember(Name="CVV\_Resp", EmitDefaultValue=false)]

public string CVVResp { get; set; }

[DataMember(Name="Items", EmitDefaultValue=false)]

public List<TransactionItem> Items { get; set; }

[DataMember(Name="Business\_Name", EmitDefaultValue=false)]

public string BusinessName { get; set; }

[DataMember(Name="Business\_Address", EmitDefaultValue=false)]

public string BusinessAddress { get; set; }

[DataMember(Name="Business\_Address2", EmitDefaultValue=false)]

public string BusinessAddress2 { get; set; }

[DataMember(Name="Business\_City", EmitDefaultValue=false)]

public string BusinessCity { get; set; }

[DataMember(Name="Business\_State", EmitDefaultValue=false)]

public string BusinessState { get; set; }

[DataMember(Name="Business\_Zip", EmitDefaultValue=false)]

public string BusinessZip { get; set; }

[DataMember(Name="Business\_Country", EmitDefaultValue=false)]

public string BusinessCountry { get; set; }

[DataMember(Name="Business\_Website", EmitDefaultValue=false)]

public string BusinessWebsite { get; set; }

[DataMember(Name="Contact\_FirstName", EmitDefaultValue=false)]

public string ContactFirstName { get; set; }

[DataMember(Name="Contact\_LastName", EmitDefaultValue=false)]

public string ContactLastName { get; set; }

[DataMember(Name="Contact\_Phone", EmitDefaultValue=false)]

public string ContactPhone { get; set; }

[DataMember(Name="Contact\_Fax", EmitDefaultValue=false)]

public string ContactFax { get; set; }

[DataMember(Name="Contact\_Email", EmitDefaultValue=false)]

public string ContactEmail { get; set; }

[DataMember(Name="Contact\_Title", EmitDefaultValue=false)]

public string ContactTitle { get; set; }

[DataMember(Name="IP", EmitDefaultValue=false)]

public string IP { get; set; }

[DataMember(Name="TaxExempt", EmitDefaultValue=false)]

public bool? TaxExempt { get; set; }

}

}

# Receipt

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class Receipt : IEquatable<Receipt>, IValidatableObject

{

public Receipt(ReceiptCustomLines header = default(ReceiptCustomLines), ReceiptCustomLines footer = default(ReceiptCustomLines))

{

this.Header = header;

this.Footer = footer;

}

[DataMember(Name="Header", EmitDefaultValue=false)]

public ReceiptCustomLines Header { get; set; }

[DataMember(Name="Footer", EmitDefaultValue=false)]

public ReceiptCustomLines Footer { get; set; }

}

}

# ReceiptCustomLines

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class ReceiptCustomLines : IEquatable<ReceiptCustomLines>, IValidatableObject

{

public ReceiptCustomLines(string customLine1 = default(string), string customLine2 = default(string), string customLine3 = default(string), string customLine4 = default(string))

{

this.CustomLine1 = customLine1;

this.CustomLine2 = customLine2;

this.CustomLine3 = customLine3;

this.CustomLine4 = customLine4;

}

[DataMember(Name="CustomLine1", EmitDefaultValue=false)]

public string CustomLine1 { get; set; }

[DataMember(Name="CustomLine2", EmitDefaultValue=false)]

public string CustomLine2 { get; set; }

[DataMember(Name="CustomLine3", EmitDefaultValue=false)]

public string CustomLine3 { get; set; }

[DataMember(Name="CustomLine4", EmitDefaultValue=false)]

public string CustomLine4 { get; set; }

}

}

# Record

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class Record : IEquatable<Record>, IValidatableObject

{

public Record(Response response = default(Response), int? iD = default(int?), string reference = default(string), Business business = default(Business), Contact contact = default(Contact), string pO = default(string), string invoice = default(string), string customer = default(string), bool? taxExempt = default(bool?), string note = default(string), List<VaultCreditCard> creditCards = default(List<VaultCreditCard>), List<VaultCheck> checks = default(List<VaultCheck>))

{

this.Response = response;

this.ID = iD;

this.Reference = reference;

this.Business = business;

this.Contact = contact;

this.PO = pO;

this.Invoice = invoice;

this.Customer = customer;

this.TaxExempt = taxExempt;

this.Note = note;

this.CreditCards = creditCards;

this.Checks = checks;

}

[DataMember(Name="Response", EmitDefaultValue=false)]

public Response Response { get; set; }

[DataMember(Name="ID", EmitDefaultValue=false)]

public int? ID { get; set; }

[DataMember(Name="Reference", EmitDefaultValue=false)]

public string Reference { get; set; }

[DataMember(Name="Business", EmitDefaultValue=false)]

public Business Business { get; set; }

[DataMember(Name="Contact", EmitDefaultValue=false)]

public Contact Contact { get; set; }

[DataMember(Name="PO", EmitDefaultValue=false)]

public string PO { get; set; }

[DataMember(Name="Invoice", EmitDefaultValue=false)]

public string Invoice { get; set; }

[DataMember(Name="Customer", EmitDefaultValue=false)]

public string Customer { get; set; }

[DataMember(Name="TaxExempt", EmitDefaultValue=false)]

public bool? TaxExempt { get; set; }

[DataMember(Name="Note", EmitDefaultValue=false)]

public string Note { get; set; }

[DataMember(Name="CreditCards", EmitDefaultValue=false)]

public List<VaultCreditCard> CreditCards { get; set; }

[DataMember(Name="Checks", EmitDefaultValue=false)]

public List<VaultCheck> Checks { get; set; }

}

}

# Response

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class Response : IEquatable<Response>, IValidatableObject

{

public Response(bool? error = default(bool?), string message = default(string))

{

this.Error = error;

this.Message = message;

}

[DataMember(Name="Error", EmitDefaultValue=false)]

public bool? Error { get; set; }

[DataMember(Name="Message", EmitDefaultValue=false)]

public string Message { get; set; }

}

}

# SearchVault

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class SearchVault : IEquatable<SearchVault>, IValidatableObject

{

public SearchVault(string reference = default(string), string businessName = default(string), string firstName = default(string), string lastName = default(string))

{

this.Reference = reference;

this.BusinessName = businessName;

this.FirstName = firstName;

this.LastName = lastName;

}

[DataMember(Name="Reference", EmitDefaultValue=false)]

public string Reference { get; set; }

[DataMember(Name="BusinessName", EmitDefaultValue=false)]

public string BusinessName { get; set; }

[DataMember(Name="FirstName", EmitDefaultValue=false)]

public string FirstName { get; set; }

[DataMember(Name="LastName", EmitDefaultValue=false)]

public string LastName { get; set; }

}

}

# Settings

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class Settings : IEquatable<Settings>, IValidatableObject

{

public Settings(CreditCardTransactionSettings creditCardTransactionSettings = default(CreditCardTransactionSettings), CheckTransactionSettings checkTransactionSettings = default(CheckTransactionSettings), Business business = default(Business), Contact mainContact = default(Contact), Receipt receipt40 = default(Receipt), Receipt receipt80 = default(Receipt), bool? printSignatureLine = default(bool?), Response response = default(Response), string accountID = default(string), FieldInfo copyEmailReceipt = default(FieldInfo))

{

this.CreditCardTransactionSettings = creditCardTransactionSettings;

this.CheckTransactionSettings = checkTransactionSettings;

this.Business = business;

this.MainContact = mainContact;

this.Receipt40 = receipt40;

this.Receipt80 = receipt80;

this.PrintSignatureLine = printSignatureLine;

this.Response = response;

this.AccountID = accountID;

this.CopyEmailReceipt = copyEmailReceipt;

}

[DataMember(Name="CreditCardTransactionSettings", EmitDefaultValue=false)]

public CreditCardTransactionSettings CreditCardTransactionSettings { get; set; }

[DataMember(Name="CheckTransactionSettings", EmitDefaultValue=false)]

public CheckTransactionSettings CheckTransactionSettings { get; set; }

[DataMember(Name="Business", EmitDefaultValue=false)]

public Business Business { get; set; }

[DataMember(Name="MainContact", EmitDefaultValue=false)]

public Contact MainContact { get; set; }

[DataMember(Name="Receipt40", EmitDefaultValue=false)]

public Receipt Receipt40 { get; set; }

[DataMember(Name="Receipt80", EmitDefaultValue=false)]

public Receipt Receipt80 { get; set; }

[DataMember(Name="PrintSignatureLine", EmitDefaultValue=false)]

public bool? PrintSignatureLine { get; set; }

[DataMember(Name="Response", EmitDefaultValue=false)]

public Response Response { get; set; }

[DataMember(Name="AccountID", EmitDefaultValue=false)]

public string AccountID { get; set; }

[DataMember(Name="CopyEmailReceipt", EmitDefaultValue=false)]

public FieldInfo CopyEmailReceipt { get; set; }

}

}

# TerminalSettings

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class TerminalSettings : IEquatable<TerminalSettings>, IValidatableObject

{

public TerminalSettings(int? iD = default(int?), int? terminalID = default(int?), string sECCode = default(string), bool? gatewayTerminal = default(bool?), bool? consumerCredits = default(bool?), bool? dLRequired = default(bool?), bool? checkVerify = default(bool?), bool? identityVerify = default(bool?))

{

this.ID = iD;

this.TerminalID = terminalID;

this.SECCode = sECCode;

this.GatewayTerminal = gatewayTerminal;

this.ConsumerCredits = consumerCredits;

this.DLRequired = dLRequired;

this.CheckVerify = checkVerify;

this.IdentityVerify = identityVerify;

}

[DataMember(Name="ID", EmitDefaultValue=false)]

public int? ID { get; set; }

[DataMember(Name="Terminal\_ID", EmitDefaultValue=false)]

public int? TerminalID { get; set; }

[DataMember(Name="SEC\_Code", EmitDefaultValue=false)]

public string SECCode { get; set; }

[DataMember(Name="Gateway\_Terminal", EmitDefaultValue=false)]

public bool? GatewayTerminal { get; set; }

[DataMember(Name="Consumer\_Credits", EmitDefaultValue=false)]

public bool? ConsumerCredits { get; set; }

[DataMember(Name="DL\_Required", EmitDefaultValue=false)]

public bool? DLRequired { get; set; }

[DataMember(Name="Check\_Verify", EmitDefaultValue=false)]

public bool? CheckVerify { get; set; }

[DataMember(Name="Identity\_Verify", EmitDefaultValue=false)]

public bool? IdentityVerify { get; set; }

}

}

# Transaction

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class Transaction : IEquatable<Transaction>, IValidatableObject

{

public Transaction(DateTime? submitTime = default(DateTime?), string tranType = default(string), double? tranAmt = default(double?), double? tranTax = default(double?), string tranInv = default(string), string tranPO = default(string), string tranCNum = default(string), string tranNote = default(string), string cardNum = default(string), string cardName = default(string), string cardExp = default(string), string cardType = default(string), string aVSStreet = default(string), string aVSZip = default(string), string cVVNum = default(string), string procID = default(string), string procResp = default(string), string procCode = default(string), string procMess = default(string), string aVSCode = default(string), string aVSResp = default(string), string cVVCode = default(string), string cVVResp = default(string), List<TransactionItem> items = default(List<TransactionItem>), string businessName = default(string), string businessAddress = default(string), string businessAddress2 = default(string), string businessCity = default(string), string businessState = default(string), string businessZip = default(string), string businessCountry = default(string), string businessWebsite = default(string), string contactFirstName = default(string), string contactLastName = default(string), string contactPhone = default(string), string contactFax = default(string), string contactEmail = default(string), string contactTitle = default(string), string iP = default(string), bool? taxExempt = default(bool?))

{

this.SubmitTime = submitTime;

this.TranType = tranType;

this.TranAmt = tranAmt;

this.TranTax = tranTax;

this.TranInv = tranInv;

this.TranPO = tranPO;

this.TranCNum = tranCNum;

this.TranNote = tranNote;

this.CardNum = cardNum;

this.CardName = cardName;

this.CardExp = cardExp;

this.CardType = cardType;

this.AVSStreet = aVSStreet;

this.AVSZip = aVSZip;

this.CVVNum = cVVNum;

this.ProcID = procID;

this.ProcResp = procResp;

this.ProcCode = procCode;

this.ProcMess = procMess;

this.AVSCode = aVSCode;

this.AVSResp = aVSResp;

this.CVVCode = cVVCode;

this.CVVResp = cVVResp;

this.Items = items;

this.BusinessName = businessName;

this.BusinessAddress = businessAddress;

this.BusinessAddress2 = businessAddress2;

this.BusinessCity = businessCity;

this.BusinessState = businessState;

this.BusinessZip = businessZip;

this.BusinessCountry = businessCountry;

this.BusinessWebsite = businessWebsite;

this.ContactFirstName = contactFirstName;

this.ContactLastName = contactLastName;

this.ContactPhone = contactPhone;

this.ContactFax = contactFax;

this.ContactEmail = contactEmail;

this.ContactTitle = contactTitle;

this.IP = iP;

this.TaxExempt = taxExempt;

}

[DataMember(Name="Submit\_Time", EmitDefaultValue=false)]

public DateTime? SubmitTime { get; set; }

[DataMember(Name="Tran\_Type", EmitDefaultValue=false)]

public string TranType { get; set; }

[DataMember(Name="Tran\_Amt", EmitDefaultValue=false)]

public double? TranAmt { get; set; }

[DataMember(Name="Tran\_Tax", EmitDefaultValue=false)]

public double? TranTax { get; set; }

[DataMember(Name="Tran\_Inv", EmitDefaultValue=false)]

public string TranInv { get; set; }

[DataMember(Name="Tran\_PO", EmitDefaultValue=false)]

public string TranPO { get; set; }

[DataMember(Name="Tran\_CNum", EmitDefaultValue=false)]

public string TranCNum { get; set; }

[DataMember(Name="Tran\_Note", EmitDefaultValue=false)]

public string TranNote { get; set; }

[DataMember(Name="Card\_Num", EmitDefaultValue=false)]

public string CardNum { get; set; }

[DataMember(Name="Card\_Name", EmitDefaultValue=false)]

public string CardName { get; set; }

[DataMember(Name="Card\_Exp", EmitDefaultValue=false)]

public string CardExp { get; set; }

[DataMember(Name="Card\_Type", EmitDefaultValue=false)]

public string CardType { get; set; }

[DataMember(Name="AVS\_Street", EmitDefaultValue=false)]

public string AVSStreet { get; set; }

[DataMember(Name="AVS\_Zip", EmitDefaultValue=false)]

public string AVSZip { get; set; }

[DataMember(Name="CVV\_Num", EmitDefaultValue=false)]

public string CVVNum { get; set; }

[DataMember(Name="Proc\_ID", EmitDefaultValue=false)]

public string ProcID { get; set; }

[DataMember(Name="Proc\_Resp", EmitDefaultValue=false)]

public string ProcResp { get; set; }

[DataMember(Name="Proc\_Code", EmitDefaultValue=false)]

public string ProcCode { get; set; }

[DataMember(Name="Proc\_Mess", EmitDefaultValue=false)]

public string ProcMess { get; set; }

[DataMember(Name="AVS\_Code", EmitDefaultValue=false)]

public string AVSCode { get; set; }

[DataMember(Name="AVS\_Resp", EmitDefaultValue=false)]

public string AVSResp { get; set; }

[DataMember(Name="CVV\_Code", EmitDefaultValue=false)]

public string CVVCode { get; set; }

[DataMember(Name="CVV\_Resp", EmitDefaultValue=false)]

public string CVVResp { get; set; }

[DataMember(Name="Items", EmitDefaultValue=false)]

public List<TransactionItem> Items { get; set; }

[DataMember(Name="Business\_Name", EmitDefaultValue=false)]

public string BusinessName { get; set; }

[DataMember(Name="Business\_Address", EmitDefaultValue=false)]

public string BusinessAddress { get; set; }

[DataMember(Name="Business\_Address2", EmitDefaultValue=false)]

public string BusinessAddress2 { get; set; }

[DataMember(Name="Business\_City", EmitDefaultValue=false)]

public string BusinessCity { get; set; }

[DataMember(Name="Business\_State", EmitDefaultValue=false)]

public string BusinessState { get; set; }

[DataMember(Name="Business\_Zip", EmitDefaultValue=false)]

public string BusinessZip { get; set; }

[DataMember(Name="Business\_Country", EmitDefaultValue=false)]

public string BusinessCountry { get; set; }

[DataMember(Name="Business\_Website", EmitDefaultValue=false)]

public string BusinessWebsite { get; set; }

[DataMember(Name="Contact\_FirstName", EmitDefaultValue=false)]

public string ContactFirstName { get; set; }

[DataMember(Name="Contact\_LastName", EmitDefaultValue=false)]

public string ContactLastName { get; set; }

[DataMember(Name="Contact\_Phone", EmitDefaultValue=false)]

public string ContactPhone { get; set; }

[DataMember(Name="Contact\_Fax", EmitDefaultValue=false)]

public string ContactFax { get; set; }

[DataMember(Name="Contact\_Email", EmitDefaultValue=false)]

public string ContactEmail { get; set; }

[DataMember(Name="Contact\_Title", EmitDefaultValue=false)]

public string ContactTitle { get; set; }

[DataMember(Name="IP", EmitDefaultValue=false)]

public string IP { get; set; }

[DataMember(Name="TaxExempt", EmitDefaultValue=false)]

public bool? TaxExempt { get; set; }

}

}

# TransactionItem

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class TransactionItem : IEquatable<TransactionItem>, IValidatableObject

{

public TransactionItem(bool? taxIncluded = default(bool?), bool? credit = default(bool?), double? extendedAmount = default(double?), string itemNumber = default(string), double? unitCost = default(double?), int? quantity = default(int?), string description = default(string), double? discount = default(double?), double? tax = default(double?), string commodityCode = default(string), string unitOfMeasure = default(string))

{

this.TaxIncluded = taxIncluded;

this.Credit = credit;

this.ExtendedAmount = extendedAmount;

this.ItemNumber = itemNumber;

this.UnitCost = unitCost;

this.Quantity = quantity;

this.Description = description;

this.Discount = discount;

this.Tax = tax;

this.CommodityCode = commodityCode;

this.UnitOfMeasure = unitOfMeasure;

}

[DataMember(Name="TaxIncluded", EmitDefaultValue=false)]

public bool? TaxIncluded { get; set; }

[DataMember(Name="Credit", EmitDefaultValue=false)]

public bool? Credit { get; set; }

[DataMember(Name="ExtendedAmount", EmitDefaultValue=false)]

public double? ExtendedAmount { get; set; }

[DataMember(Name="ItemNumber", EmitDefaultValue=false)]

public string ItemNumber { get; set; }

[DataMember(Name="UnitCost", EmitDefaultValue=false)]

public double? UnitCost { get; set; }

[DataMember(Name="Quantity", EmitDefaultValue=false)]

public int? Quantity { get; set; }

[DataMember(Name="Description", EmitDefaultValue=false)]

public string Description { get; set; }

[DataMember(Name="Discount", EmitDefaultValue=false)]

public double? Discount { get; set; }

[DataMember(Name="Tax", EmitDefaultValue=false)]

public double? Tax { get; set; }

[DataMember(Name="CommodityCode", EmitDefaultValue=false)]

public string CommodityCode { get; set; }

[DataMember(Name="UnitOfMeasure", EmitDefaultValue=false)]

public string UnitOfMeasure { get; set; }

}

}

# UpdateTransactionInfo

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class UpdateTransactionInfo : IEquatable<UpdateTransactionInfo>, IValidatableObject

{

public UpdateTransactionInfo(string invoiceNumber = default(string), string customerNumber = default(string), string pONumber = default(string), string note = default(string))

{

this.InvoiceNumber = invoiceNumber;

this.CustomerNumber = customerNumber;

this.PONumber = pONumber;

this.Note = note;

}

[DataMember(Name="InvoiceNumber", EmitDefaultValue=false)]

public string InvoiceNumber { get; set; }

[DataMember(Name="CustomerNumber", EmitDefaultValue=false)]

public string CustomerNumber { get; set; }

[DataMember(Name="PONumber", EmitDefaultValue=false)]

public string PONumber { get; set; }

[DataMember(Name="Note", EmitDefaultValue=false)]

public string Note { get; set; }

}

}

# VaultCheck

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class VaultCheck : IEquatable<VaultCheck>, IValidatableObject

{

public VaultCheck(Response response = default(Response), int? checkID = default(int?), bool? main = default(bool?), string checkRouting = default(string), string checkAccount = default(string), string checkNumber = default(string), string checkFirstName = default(string), string checkLastName = default(string), string note = default(string), string maskedRouting = default(string), string maskedAccount = default(string), string accountType = default(string))

{

this.Response = response;

this.CheckID = checkID;

this.Main = main;

this.CheckRouting = checkRouting;

this.CheckAccount = checkAccount;

this.CheckNumber = checkNumber;

this.CheckFirstName = checkFirstName;

this.CheckLastName = checkLastName;

this.Note = note;

this.MaskedRouting = maskedRouting;

this.MaskedAccount = maskedAccount;

this.AccountType = accountType;

}

[DataMember(Name="Response", EmitDefaultValue=false)]

public Response Response { get; set; }

[DataMember(Name="CheckID", EmitDefaultValue=false)]

public int? CheckID { get; set; }

[DataMember(Name="Main", EmitDefaultValue=false)]

public bool? Main { get; set; }

[DataMember(Name="CheckRouting", EmitDefaultValue=false)]

public string CheckRouting { get; set; }

[DataMember(Name="CheckAccount", EmitDefaultValue=false)]

public string CheckAccount { get; set; }

[DataMember(Name="CheckNumber", EmitDefaultValue=false)]

public string CheckNumber { get; set; }

[DataMember(Name="CheckFirstName", EmitDefaultValue=false)]

public string CheckFirstName { get; set; }

[DataMember(Name="CheckLastName", EmitDefaultValue=false)]

public string CheckLastName { get; set; }

[DataMember(Name="Note", EmitDefaultValue=false)]

public string Note { get; set; }

[DataMember(Name="MaskedRouting", EmitDefaultValue=false)]

public string MaskedRouting { get; set; }

[DataMember(Name="MaskedAccount", EmitDefaultValue=false)]

public string MaskedAccount { get; set; }

[DataMember(Name="AccountType", EmitDefaultValue=false)]

public string AccountType { get; set; }

}

}

# VaultCreditCard

using System;

using System.Linq;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Runtime.Serialization;

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System.ComponentModel.DataAnnotations;

using SwaggerDateConverter = IO.Swagger.Client.SwaggerDateConverter;

namespace IO.Swagger.Model

{

[DataContract]

public partial class VaultCreditCard : IEquatable<VaultCreditCard>, IValidatableObject

{

public VaultCreditCard(string magneticTracks = default(string), string track1 = default(string), string track2 = default(string), string number = default(string), string maskedNumber = default(string), string holder = default(string), string expDate = default(string), string cVV = default(string), string address = default(string), string city = default(string), string state = default(string), string zIP = default(string), bool? swiped = default(bool?), string typeString = default(string), CVVPresenceType cVVType = default(CVVPresenceType), bool? notPresent = default(bool?), Response response = default(Response), int? cardID = default(int?), string note = default(string), bool? main = default(bool?))

{

this.MagneticTracks = magneticTracks;

this.Track1 = track1;

this.Track2 = track2;

this.Number = number;

this.MaskedNumber = maskedNumber;

this.Holder = holder;

this.ExpDate = expDate;

this.CVV = cVV;

this.Address = address;

this.City = city;

this.State = state;

this.ZIP = zIP;

this.Swiped = swiped;

this.TypeString = typeString;

this.CVVType = cVVType;

this.NotPresent = notPresent;

this.Response = response;

this.CardID = cardID;

this.Note = note;

this.Main = main;

}

[DataMember(Name="MagneticTracks", EmitDefaultValue=false)]

public string MagneticTracks { get; set; }

[DataMember(Name="Track1", EmitDefaultValue=false)]

public string Track1 { get; set; }

[DataMember(Name="Track2", EmitDefaultValue=false)]

public string Track2 { get; set; }

[DataMember(Name="Number", EmitDefaultValue=false)]

public string Number { get; set; }

[DataMember(Name="MaskedNumber", EmitDefaultValue=false)]

public string MaskedNumber { get; set; }

[DataMember(Name="Holder", EmitDefaultValue=false)]

public string Holder { get; set; }

[DataMember(Name="ExpDate", EmitDefaultValue=false)]

public string ExpDate { get; set; }

[DataMember(Name="CVV", EmitDefaultValue=false)]

public string CVV { get; set; }

[DataMember(Name="Address", EmitDefaultValue=false)]

public string Address { get; set; }

[DataMember(Name="City", EmitDefaultValue=false)]

public string City { get; set; }

[DataMember(Name="State", EmitDefaultValue=false)]

public string State { get; set; }

[DataMember(Name="ZIP", EmitDefaultValue=false)]

public string ZIP { get; set; }

[DataMember(Name="Swiped", EmitDefaultValue=false)]

public bool? Swiped { get; set; }

[DataMember(Name="TypeString", EmitDefaultValue=false)]

public string TypeString { get; set; }

[DataMember(Name="CVVType", EmitDefaultValue=false)]

public CVVPresenceType CVVType { get; set; }

[DataMember(Name="NotPresent", EmitDefaultValue=false)]

public bool? NotPresent { get; set; }

[DataMember(Name="Response", EmitDefaultValue=false)]

public Response Response { get; set; }

[DataMember(Name="CardID", EmitDefaultValue=false)]

public int? CardID { get; set; }

[DataMember(Name="Note", EmitDefaultValue=false)]

public string Note { get; set; }

[DataMember(Name="Main", EmitDefaultValue=false)]

public bool? Main { get; set; }

}

}