

# Servlet to REST API Migration Guide

March 2019



## CyberSource Contact Information

For general information about our company, products, and services, go to <http://www.cybersource.com>.

For sales questions about any CyberSource Service, email [sales@cybersource.com](mailto:sales@cybersource.com) or call 650-432-7350 or 888-330-2300 (toll free in the United States).

For support information about any CyberSource Service, visit the Support Center: <http://www.cybersource.com/support>

## Copyright

© 2019 CyberSource Corporation. All rights reserved. CyberSource Corporation ("CyberSource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and CyberSource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by CyberSource. CyberSource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of CyberSource.

## Restricted Rights Legends

**For Government or defense agencies.** Use, duplication, or disclosure by the Government or defense agencies is subject to restrictions as set forth the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

**For civilian agencies.** Use, reproduction, or disclosure is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in CyberSource Corporation's standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States.

## Trademarks

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of CyberSource Corporation.

CyberSource, CyberSource Payment Manager, CyberSource Risk Manager, CyberSource Decision Manager, and CyberSource Connect are trademarks and/or service marks of CyberSource Corporation.

All other brands and product names are trademarks or registered trademarks of their respective owners.

# Contents

## **Recent Revisions to This Document** 4

## **About This Guide** 5

Conventions 5

    Note Statements 5

    Text and Command Conventions 5

Related Documents 6

Customer Support 6

---

## **Chapter 1** **Transitioning from Servlet to REST** 8

Servlet and REST Request Examples 9

REST FAQs 12

    What is an API? 12

    What is an API specification? 13

**What are the components of a REST API?** 13

---

## **Appendix 2** **Field Mapping** 15

Field Names in JSON Response 15

# Recent Revisions to This Document

Release	Changes
March 2019	Initial release of document.

# About This Guide

This guide is for merchants or partners who used the programmatic download (servlet) to download reports and files, search for transactions, and retrieve user details, and are transitioning to the use of REST APIs for these functions.

If you access the legacy reports programmatically, you can continue to use the servlets until July 31, 2019. After July 31, 2019, the old servlets will be retired. After this date, you will no longer be able to access old reports via servlets.

REST APIs are available in our [Developer Center](#).

## Conventions

---

### Note Statements



**Note**

A *Note* contains helpful suggestions or references to material not contained in the document.

### Text and Command Conventions

Convention	Usage
<b>Bold</b>	<ul style="list-style-type: none"> <li>Field and service names in text; for example: Include the <b>ics_applications</b> field.</li> <li>Items that you are instructed to act upon; for example: Click <b>Save</b>.</li> </ul>
Screen text	<ul style="list-style-type: none"> <li>XML elements.</li> <li>Code examples and samples.</li> <li>Text that you enter in an API environment; for example: Set the <b>davService_run</b> field to <code>true</code>.</li> </ul>

## Related Documents

---

- *Reporting REST API Developer Guide*
- *Reporting User Guide*
- *New Business Center User Guide*

Refer to the Support Center for complete CyberSource technical documentation:

[http://www.cybersource.com/support\\_center/support\\_documentation](http://www.cybersource.com/support_center/support_documentation)

## Customer Support

---

For support information about any CyberSource service, visit the Support Center at:

<http://www.cybersource.com/support>



# Transitioning from Servlet to REST

In the classic Business Center, users could download reports and files, search for transactions, and retrieve user details via programmatic access, which we refer to as servlets. The servlets were:

- Download servlet
- Acquiring servlet
- Query servlet

Reports and other data that were accessed via servlets are now be available using REST APIs. The new REST APIs include:

- Reporting API
- Transaction Detail API
- Transaction Search API
- User Management API
- Transaction Batch Service API

For detailed information about the servlets, see the [Legacy Business Center Reporting Developer Guide](#).

The table below lists the functions of the old servlets and the REST APIs that will replace them.

Servlet Name	Function	REST API
Download Servlet	Downloads report or other files.	Reporting API or Secure File Share API
Acquiring Servlet	Downloads reports.	Reporting API
Query Servlet	Retrieve Transaction/Payment Status.	Transaction Detail API
Query Servlet	Retrieve a single transaction (search by request ID).	Transaction Detail API
Query Servlet	Search for transactions by merchant reference number.	Transaction Search API
Query Servlet	Notification of Change (on demand).	Reporting API
Query Servlet	Conversion Detail Report (on demand).	Reporting API



Servlet Name	Function	REST API
Query Servlet	Order Detail Report (on demand)	Reporting API
Query Servlet	User Listing Servlet	User Management API
Query Servlet	Response File Write (Batch Upload status)	Batch Status API

These are the new REST APIs:

API Name	Purpose
Transaction Detail API	Retrieves a single transaction based on Request ID.
Search API	Retrieves a set of transactions based on several available query parameters.
Reporting API	Downloads and manages downloadable detail reports.
Secure File Share API	Downloads third party files, reports from the old Business Center, Batch Upload Reply, and Account Updater Reply files.
User Management API	Retrieves a list of users.
Batch Status API	Retrieves the status of a batch upload.

For information on using REST APIs, see [CyberSource REST API Getting Started Guide](#). For full documentation on the REST APIs, see our [Developer Center](#).

## Servlet and REST Request Examples

The Download Servlet sends a GET request using the following URL format:

```
https://<server_name>/reports/servlet/DownloadReport/YYYY/MM/DD/<merchant_ID>/<report_name>.<report_format>
```

The Query Servlet sends a POST request with search parameters:

```
https://<server_name>/ebc/Query
```

The search parameters required for each request are dependent on the servlet function.

REST requests vary by function. For detailed information, see the Reporting REST API, Transaction Detail REST API, and Transaction Search REST API in our Developer Center.

The examples below show the differences between servlet and REST requests for different functions.

**Table 1 Servlet vs REST Request Examples**

---

**Function: Report Download [Download]**

**Servlet Request:** `https://<server_name>/reports/servlet/DownloadReport/YYYY/MM/DD/<merchant_ID>/<report_name>.<report_format>`

**REST Request:** `https://api.cybersource.com/reporting/v3/report-downloads`

```
{
  "organizationID": "myorg",
  "reportDate": "2018-05-23",
  "reportName": "MyReport"
}
```

---

**Function: Single Transaction Query (Transaction Detail in New Business Center)**

**Servlet Request:** `https://<server_name>/ebc/Query?merchantID=merchant1&requestID=5187901209120179112634&type=transaction&subtype=transactionDetail&versionNumber=1.7`

**REST Request:** `https://api.cybersource.com/tss/v2/transactions/{id}`

---

**Function: Transaction Search by Merchant Reference Number**

**Servlet Request:** `https://<server_name>/ebc/Query?merchantID=merchant1&merchantReferenceNumber=1234&targetDate=20180523&type=transaction&subtype=transactionDetail&versionNumber=1.7`

**REST Request:** `https://api.cybersource.com/tss/v2/searches`

```
{
  "save": "false",
  "name": "Search By Code",
  "timezone": "America/Chicago",
  "query": "clientReferenceInformation.code:123456",
  "offset": 0,
  "limit": 100,
  "sort": "id:asc, submitTimeUtc:asc"
}
```

---

**Table 1 Servlet vs REST Request Examples (Continued)**

---

**Function: Order Detail Report (Transaction Request Report in New Business Center)**

**Servlet Request:** `https://<server_name>/ebc/Query?merchantID=merchant1&requestID=5187901209120179112634&type=orderPaymentData`

**REST Request:** `https://api.cybersource.com/reporting/v3/report-downloads`

```
{
  "organizationID": "myorg",
  "reportDate": "2018-05-23",
  "reportName": "TransactionRequestReport"
}
```

---

**Function: Notification of Change**

**Servlet Request:** `https://<server_name>/ebc/Query?merchantID=pa_rbsworldpay&startDate=2018-02-15&endDate=2018-02-16&type=NOC&exportType=xml`

**REST Request:** `https://api.cybersource.com/reporting/v3/notification-of-changes`

```
{
  "startTime": "yyyy-MM-dd'T'HH:mm:ssXXX",
  "endTime": "yyyy-MM-dd'T'HH:mm:ssXXX"
}
```

---

**Function: Batch Submission Detail Report**

**Servlet Request:** `https://<server_name>/ebc/batchupload/ResponseFileWrite.do?merchantID=merchant1&batchID=12345&format=csv&password=password&username=username`

**REST Request:** `https://api.cybersource.com/pts/v1/transaction-batches`

```
{
  "startTime": "yyyy-MM-dd'T'HH:mm:ssXXX",
  "endTime": "yyyy-MM-dd'T'HH:mm:ssXXX"
}
```

---

**Table 1 Servlet vs REST Request Examples (Continued)**

---

**Function: Conversion Detail Report**

**Servlet Request:** `https://<server_name>/ebc/ConversionDetailReportRequest.do?merchantID=testmerchant&startDate=2018-03-23&startTime=00:00:00&endDate=2018-03-23&endTime=23:59:59&username=username&password=password`

**REST Request:** `https://api.cybersource.com/reporting/v3/conversion-details`

```
{
  "startTime": "yyyy-MM-dd'T'HH:mm:ssXXX",
  "endTime": "yyyy-MM-dd'T'HH:mm:ssXXX"
  "organizationId": "organizationID"
}
```

---

**Function: Purchase and Refund Details [Acquiring]**

**Servlet Request:** `https://<server_name>/ebc/DownloadAcquiringReports?hierarchyId=1234&organizationId=orgID&reportId=75&startDay=01&startMonth=12&startYear=2017&endDay=31&endMonth=12&endYear=2017&format=xml`

**REST Request:** `https://api.cybersource.com/reporting/v3/purchase-refund-details`

```
{
  "startTime": "yyyy-MM-dd'T'HH:mm:ssXXX",
  "endTime": "yyyy-MM-dd'T'HH:mm:ssXXX"
  "organizationId": "organizationID"
  "paymentSubtype": "ALL"
  "viewBy": "requestDate"
  "groupName": "group1"
  "offset": "5"
  "limit": "2000"
}
```

---

## REST FAQs

---

### What is an API?

API stands for Application Programming Interface. It's a way for computer programs to talk to each other. The program using the API is known as the client. The client is usually written in a language such as Java or JavaScript.

The conversation between the programs follows the pattern of request and response.

The request:

- What do you want to do?
  - Get information.
  - Perform some task.

The response:

- The status of your request - did it fail or succeed?
- Other information you might have asked for.
- Often this is called the payload.

## What is an API specification?

The conversation between programs happens in a structured way. The way to talk to an API is known as the “interface”, which is specific structure or specification standard.

## What are the components of a REST API?

The components of a REST API are:

- Resource—A resource is the fundamental component of an API. It is considered an object (noun) and can have attributes like a name or a date. A resource can also be a collection of other resources.
- Verb—A verb is the action you want to perform with the API, such as the following:
  - GET - Used to read a resource.
  - PUT - Use to update a resource.
  - POST - Used to make a new resource.
  - PATCH - Similar to put.
  - DELETE - Delete a resource.
- Parameters—Parameters are extra information that tell the application how to handle your request. With the Reporting API, you can:
  - Get a recurring or a one-time report.
  - Create and update report subscriptions.
  - Delete subscriptions.
  - Get a list of all subscriptions.

- Get the status of a report.

For example:

```
GET <url_prefix>/v2/  
reports?orgID=BerniesNeatMerchnt&download=true&reportName=BerniesNeatRepo  
rt&reportDate=20161215
```

In the example above, "GET" is the verb; "reports" is the resource; and the parameters are:

- orgId
- download
- reportName
- reportDate

# Field Mapping

## Field Names in JSON Response

The Transaction Search and Transaction Detail REST APIs return JSON responses. The table below shows the new REST field names and the old servlet field names.

**Table 2 Servlet to REST Field Mapping**

Servlet Field Name	REST Field Name
ApplicationReplies.ApplicationReply.Name	applicationInformation.applications[].name
ApplicationReplies.ApplicationReply.Rcode	applicationInformation.applications[].rCode
ApplicationReplies.ApplicationReply.Rflag	applicationInformation.applications[].rFlag
ApplicationReplies.ApplicationReply.RMsg	HTTP 201: errorInformation.message HTTP 400/502: message
Applications.application	applicationInformation.applications[].name
Applications.reasonCode	applicationInformation.applications[].reasonCode
Applications.status	applicationInformation.applications[].status
BillTo.Address1	orderInformation.billTo.address1
BillTo.Address2	orderInformation.billTo.address2
BillTo.City	orderInformation.billTo.locality
BillTo.CompanyName	orderInformation.billTo.company
BillTo.Country	orderInformation.billTo.country
BillTo.CustomerID	buyerInformation.merchantCustomerId
BillTo.Email	orderInformation.billTo.email
BillTo.FirstName	orderInformation.billTo.firstName
BillTo.Hostname	deviceInformation.hostName
BillTo.IPAddress	deviceInformation.ipAddress
BillTo.LastName	orderInformation.billTo.lastName
BillTo.MiddleName	orderInformation.billTo.middleName
BillTo.NameSuffix	orderInformation.billTo.suffix
BillTo.Phone	orderInformation.billTo.phoneNumber
BillTo.State	orderInformation.billTo.administrativeArea

**Table 2 Servlet to REST Field Mapping**

<b>Servlet Field Name</b>	<b>REST Field Name</b>
BillTo.Title	orderInformation.billTo.title
BillTo.UserName	buyerInformation.merchantCustomerId
BillTo.Zip	orderInformation.billTo.postalCode
Comments	clientReferenceInformation.applicationUser
LineItems.lineItem.FulfillmentType	orderInformation.lineItems[].fulfillmentType
LineItems.lineItem.MerchantProductSKU	orderInformation.lineItems[].productSku
LineItems.lineItem.Number	Not required to return as API response but it should be used to reconstruct the line item array.
LineItems.lineItem.ProductCode	orderInformation.lineItems[].productCode
LineItems.lineItem.ProductName	orderInformation.lineItems[].productName
LineItems.lineItem.Quantity	orderInformation.lineItems[].quantity
LineItems.lineItem.TaxAmount	orderInformation.lineItems[].taxAmount
LineItems.lineItem.UnitPrice	orderInformation.lineItems[].unitPrice
MerchantDefinedData.field1	merchantDefinedInformation[].key = "field1" merchantDefinedInformation[].value
MerchantDefinedData.field2	merchantDefinedInformation[].key = "field2" merchantDefinedInformation[].value
MerchantDefinedData.field3	merchantDefinedInformation[].key = "field3" merchantDefinedInformation[].value
MerchantDefinedData.field4	merchantDefinedInformation[].key = "field4" merchantDefinedInformation[].value
MerchantDefinedData.field5	merchantDefinedInformation[].key = "field5" merchantDefinedInformation[].value
MerchantDefinedData.field6	merchantDefinedInformation[].key = "field6" merchantDefinedInformation[].value
MerchantDefinedData.field7	merchantDefinedInformation[].key = "field7" merchantDefinedInformation[].value
MerchantDefinedData.field8	merchantDefinedInformation[].key = "field8" merchantDefinedInformation[].value
MerchantDefinedData.field9	merchantDefinedInformation[].key = "field9" merchantDefinedInformation[].value
MerchantDefinedData.field10	merchantDefinedInformation[].key = "field10" merchantDefinedInformation[].value
MerchantDefinedData.field11	merchantDefinedInformation[].key = "field11" merchantDefinedInformation[].value
MerchantDefinedData.field12	merchantDefinedInformation[].key = "field12" merchantDefinedInformation[].value
MerchantDefinedData.field13	merchantDefinedInformation[].key = "field13" merchantDefinedInformation[].value



**Table 2 Servlet to REST Field Mapping**

<b>Servlet Field Name</b>	<b>REST Field Name</b>
MerchantDefinedData.field14	merchantDefinedInformation[].key = "field14" merchantDefinedInformation[].value
MerchantDefinedData.field15	merchantDefinedInformation[].key = "field15" merchantDefinedInformation[].value
MerchantDefinedData.field16	merchantDefinedInformation[].key = "field16" merchantDefinedInformation[].value
MerchantDefinedData.field17	merchantDefinedInformation[].key = "field17" merchantDefinedInformation[].value
MerchantDefinedData.field18	merchantDefinedInformation[].key = "field18" merchantDefinedInformation[].value
MerchantDefinedData.field19	merchantDefinedInformation[].key = "field19" merchantDefinedInformation[].value
MerchantDefinedData.field20	merchantDefinedInformation[].key = "field20" merchantDefinedInformation[].value
MerchantReferenceNumber	clientReferenceInformation.code
PaymentData.ACHVerificationResult	processorInformation.achVerification.resultCode Raw
PaymentData.ACHVerificationResult Mapped	processorInformation.achVerification.resultCode
PaymentData.Amount	orderInformation.amountDetails.totalAmount
PaymentData.AuthorizationCode	processorInformation.approvalCode
PaymentData.AVSResult	processorInformation.avs.codeRaw
PaymentData.AVSResultMapped	processorInformation.avs.code
PaymentData.BalanceAmount	paymentInformation.accountFeatures.balanceA mount
PaymentData.BalanceCurrencyCode	paymentInformation.accountFeatures.currency
PaymentData.CurrencyCode	orderInformation.amountDetails.currency
PaymentData.CVResult	processorInformation.cardVerification.resultCode
PaymentData.EVEmail	processorInformation.electronicVerification Results.email
PaymentData.EVEmailRaw	processorInformation.electronicVerification Results.emailRaw
PaymentData.EventType	NA
PaymentData.EVName	processorInformation.electronicVerification Results.name
PaymentData.EVNameRaw	processorInformation.electronicVerification Results.nameRaw
PaymentData.EVPhoneNumber	processorInformation.electronicVerification Results.phoneNumber

**Table 2 Servlet to REST Field Mapping**

<b>Servlet Field Name</b>	<b>REST Field Name</b>
PaymentData.EVPhoneNumberRaw	processorInformation.electronicVerificationResults.phoneNumberRaw
PaymentData.EVPostalCode	processorInformation.electronicVerificationResults.postalCode
PaymentData.EVPostalCodeRaw	processorInformation.electronicVerificationResults.postalCodeRaw
PaymentData.EVStreet	processorInformation.electronicVerificationResults.street
PaymentData.EVStreetRaw	processorInformation.electronicVerificationResults.streetRaw
PaymentData.NumberOfInstallments	installmentInformation.numberofInstallments
PaymentData.payerAuthenticationInfo.AAV_CAVV	consumerAuthenticationInformation.cavv
PaymentData.payerAuthenticationInfo.ECI	consumerAuthenticationInformation.eciRaw
PaymentData.payerAuthenticationInfo.XID	consumerAuthenticationInformation.xid
PaymentData.PaymentProcessor	processorInformation.processor.name
PaymentData.PaymentRequestID	Not supported
PaymentData.ProcessorResponseCode	processorInformation.responseCode
PaymentData.RequestedAmount	orderInformation.amountDetails.totalAmount
PaymentData.RequestedAmountCurrencyCode	orderInformation.amountDetails.currencyCode
PaymentData.TotalTaxAmount	orderInformation.amountDetails.taxAmount
PaymentMethod.Card.AccountSuffix	paymentInformation.card.suffix
PaymentMethod.Card.BoletoNumber	paymentInformation.invoice.number
PaymentMethod.Card.CardType	paymentInformation.card.type
PaymentMethod.Card.ExpirationMonth	paymentInformation.card.expirationMonth
PaymentMethod.Card.ExpirationYear	paymentInformation.card.expirationYear
PaymentMethod.Card.IssueNumber	paymentInformation.card.issueNumber
PaymentMethod.Card.StartMonth	paymentInformation.card.startMonth
PaymentMethod.Card.StartYear	paymentInformation.card.startYear
PaymentMethod.Check.AccountSuffix	paymentInformation.bank.account.suffix
PaymentMethod.Check.CheckNumber	paymentInformation.bank.account.checkNumber
PaymentMethod.WalletType	processingInformation.paymentSolution
PredecessorRequestID	rootId
Profilelist.Profile.Name	riskInformation.profile.name
Profilelist.Profile.Name	riskInformation.passiveProfile.name
Profilelist.Profile.ProfileDecision	riskInformation.profile.decision

**Table 2 Servlet to REST Field Mapping**

<b>Servlet Field Name</b>	<b>REST Field Name</b>
Profilelist.Profile.ProfileDecision	riskInformation.passiveProfile.decision
Profilelist.Profile.ProfileMode	riskInformation.profiles.mode
Profilelist.Profile.RuleList.Rule.ruleDecision	riskInformation.rules[].decision
Profilelist.Profile.RuleList.Rule.ruleDecision	riskInformation.passiveRules[].decision
Profilelist.Profile.RuleList.Rule.ruleName	riskInformation.rules[].name
Profilelist.Profile.RuleList.Rule.ruleName	riskInformation.passiveRules[].name
reasonCode	applicationInformation.reasonCode
RequestDate	submitTimeUTC
RequestID	id
RiskData.AppliedAVS	processorInformation.avs.code
RiskData.AppliedCategoryGift	Not supported
RiskData.AppliedCategoryTime	Not supported
RiskData.AppliedCV	processorInformation.cardVerification.resultCode
RiskData.AppliedHostHedge	Not supported
RiskData.AppliedThreshold	Not supported
RiskData.AppliedTimeHedge	Not supported
RiskData.AppliedVelocityHedge	Not supported
RiskData.ConsumerLoyalty	Not supported
RiskData.ConsumerPasswordProvided	buyerInformation.hashedException
RiskData.ConsumerPromotions	Not supported
RiskData.CookiesAccepted	deviceInformation.cookiesAccepted
RiskData.Factors	riskInformation.score.factorCodes[]
RiskData.GiftWrap	orderInformation.shippingDetails.giftWrap
RiskData.HostSeverity	Not supported
RiskData.LostPassword	Not supported
RiskData.ProductRisk	Not supported
RiskData.RepeatCustomer	Not supported
RiskData>ReturnsAccepted	Not supported
RiskData.Score	riskInformation.score.result
RiskData.TimeLocal	riskInformation.localTime
shipping.Carrier	Not supported
shipping.Method	orderInformation.shippingDetails.shippingMethod
ShipTo.Address1	orderInformation.shipTo.address1
ShipTo.Address2	orderInformation.shipTo.address2
ShipTo.City	orderInformation.shipTo.locality

**Table 2 Servlet to REST Field Mapping**

<b>Servlet Field Name</b>	<b>REST Field Name</b>
ShipTo.CompanyName	orderInformation.shipTo.company
ShipTo.Country	orderInformation.shipTo.country
ShipTo.FirstName	orderInformation.shipTo.firstName
ShipTo.LastName	orderInformation.shipTo.lastName
ShipTo.Phone	orderInformation.shipTo.phoneNumber
ShipTo.State	orderInformation.shipTo.administrativeArea
ShipTo.ZipShipTo	orderInformation.shipTo.postalCode
Source	clientReferenceInformation.applicationName
status	applicationInformation.status
SubscriptionID	paymentInformation.customer.customerId
TransactionReferenceNumber	reconciliationID
User	clientReferenceInformation.applicationUser
NA	paymentInformation.card.prefix
NA	paymentInformation.invoice.barcodeNumber
NA	paymentInformation.invoice.expirationDate
NA	paymentInformation.accountFeatures.previousBalanceAmount
NA	consumerAuthenticationInformation.transactionId
pay_req_id	Not supported
order_number	Not supported
request_key	Not supported
auth_type	processingInformation.authorizationOptions.authType
auth_amount	OrderInformation.amountDetails.authorizedAmount
payment_service_data	Not supported
payment_service_data_2	Not supported
response_code	processorInformation.responseCode
type_code	Not supported
acct_enc	Not supported
acct_key	Not supported
NA	paymentInformation.paymentType.name
type	paymentInformation.paymentType.type
subType	paymentInformation.paymentType.method
NA	paymentInformation.paymentType.fundingSource

**Table 2 Servlet to REST Field Mapping**

<b>Servlet Field Name</b>	<b>REST Field Name</b>
NA	paymentInformation.paymentType.fundingSourceAffiliation
NA	paymentInformation.paymentType.credential
merchant_name	merchantId
order_total_amount	orderInformation.amountDetails.totalAmount
eci	processingInformation.commerceIndicator
objFxQuoteld	Not supported
accountEncoderID	paymentInformation.card.accountEncoderId paymentInformation.bank.account.encoderId
posEntryMode	pointOfSaleInformation.entryMode
terminalCapability	pointOfSaleInformation.terminalCapability
processor_trans_id	processorInformation.transactionId
override_payment_method	paymentInformation.card.useAs
payment_network_transaction_id	processorInformation.networkTransactionId
sender_reference_number	senderInformation.referenceNumber
aft_indicator	Not supported
processor_response_id	processorInformation.responseld
provider_transaction_id	processorInformation.providerTransactionId
NA	fraudMarkingInformation.reason