{

"swagger": "2.0",

"info": {

"description": "Use the Orders Selling Partner API to programmatically retrieve order information. With this API, you can develop fast, flexible, and custom applications to manage order synchronization, perform order research, and create demand-based decision support tools. \n\n\_Note:\_ For the JP, AU, and SG marketplaces, the Orders API supports orders from 2016 onward. For all other marketplaces, the Orders API supports orders for the last two years (orders older than this don't show up in the response).",

"version": "v0",

"title": "Selling Partner API for Orders",

"contact": {

"name": "Selling Partner API Developer Support",

"url": "https://sellercentral.amazon.com/gp/mws/contactus.html"

},

"license": {

"name": "Apache License 2.0",

"url": "http://www.apache.org/licenses/LICENSE-2.0"

}

},

"host": "sellingpartnerapi-na.amazon.com",

"schemes": [

"https"

],

"consumes": [

"application/json"

],

"produces": [

"application/json"

],

"paths": {

"/orders/v0/orders": {

"get": {

"tags": [

"ordersV0"

],

"description": "Returns orders that are created or updated during the specified time period. If you want to return specific types of orders, you can apply filters to your request. `NextToken` doesn't affect any filters that you include in your request; it only impacts the pagination for the filtered orders response. \n\n\*\*Usage Plan:\*\*\n\n| Rate (requests per second) | Burst |\n| ---- | ---- |\n| 0.0167 | 20 |\n\nThe `x-amzn-RateLimit-Limit` response header contains the usage plan rate limits for the operation, when available. The preceding table contains the default rate and burst values for this operation. Selling partners whose business demands require higher throughput might have higher rate and burst values than those shown here. For more information, refer to [Usage Plans and Rate Limits](https://developer-docs.amazon.com/sp-api/docs/usage-plans-and-rate-limits-in-the-sp-api).",

"operationId": "getOrders",

"parameters": [

{

"name": "CreatedAfter",

"in": "query",

"description": "Use this date to select orders created after (or at) a specified time. Only orders placed after the specified time are returned. The date must be in [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) format.\n\n\*\*Note\*\*: Either the `CreatedAfter` parameter or the `LastUpdatedAfter` parameter is required. Both cannot be empty. `LastUpdatedAfter` and `LastUpdatedBefore` cannot be set when `CreatedAfter` is set.",

"required": false,

"type": "string"

},

{

"name": "CreatedBefore",

"in": "query",

"description": "Use this date to select orders created before (or at) a specified time. Only orders placed before the specified time are returned. The date must be in [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) format.\n\n\*\*Note\*\*: `CreatedBefore` is optional when `CreatedAfter` is set. If specified, `CreatedBefore` must be equal to or after the `CreatedAfter` date and at least two minutes before current time.",

"required": false,

"type": "string"

},

{

"name": "LastUpdatedAfter",

"in": "query",

"description": "Use this date to select orders that were last updated after (or at) a specified time. An update is defined as any change in order status, including the creation of a new order. Includes updates made by Amazon and by the seller. The date must be in [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) format.\n\n\*\*Note\*\*: Either the `CreatedAfter` parameter or the `LastUpdatedAfter` parameter is required. Both cannot be empty. `CreatedAfter` or `CreatedBefore` cannot be set when `LastUpdatedAfter` is set.",

"required": false,

"type": "string"

},

{

"name": "LastUpdatedBefore",

"in": "query",

"description": "Use this date to select orders that were last updated before (or at) a specified time. An update is defined as any change in order status, including the creation of a new order. Includes updates made by Amazon and by the seller. The date must be in [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) format.\n\n\*\*Note\*\*: `LastUpdatedBefore` is optional when `LastUpdatedAfter` is set. But if specified, `LastUpdatedBefore` must be equal to or after the `LastUpdatedAfter` date and at least two minutes before current time.",

"required": false,

"type": "string"

},

{

"name": "OrderStatuses",

"in": "query",

"description": "A list of `OrderStatus` values used to filter the results.\n\n\*\*Possible values:\*\*\n- `PendingAvailability` (This status is available for pre-orders only. The order has been placed, payment has not been authorized, and the release date of the item is in the future.)\n- `Pending` (The order has been placed but payment has not been authorized.)\n- `Unshipped` (Payment has been authorized and the order is ready for shipment, but no items in the order have been shipped.)\n- `PartiallyShipped` (One or more, but not all, items in the order have been shipped.)\n- `Shipped` (All items in the order have been shipped.)\n- `InvoiceUnconfirmed` (All items in the order have been shipped. The seller has not yet given confirmation to Amazon that the invoice has been shipped to the buyer.)\n- `Canceled` (The order has been canceled.)\n- `Unfulfillable` (The order cannot be fulfilled. This state applies only to Multi-Channel Fulfillment orders.)",

"required": false,

"type": "array",

"items": {

"type": "string"

}

},

{

"name": "MarketplaceIds",

"in": "query",

"description": "A list of `MarketplaceId` values. Used to select orders that were placed in the specified marketplaces.\n\nRefer to [Marketplace IDs](https://developer-docs.amazon.com/sp-api/docs/marketplace-ids) for a complete list of `marketplaceId` values.",

"required": true,

"type": "array",

"items": {

"type": "string"

},

"maxItems": 50

},

{

"name": "FulfillmentChannels",

"in": "query",

"description": "A list that indicates how an order was fulfilled. Filters the results by fulfillment channel. \n\n\*\*Possible values\*\*: `AFN` (fulfilled by Amazon), `MFN` (fulfilled by seller).",

"required": false,

"type": "array",

"items": {

"type": "string"

}

},

{

"name": "PaymentMethods",

"in": "query",

"description": "A list of payment method values. Use this field to select orders that were paid with the specified payment methods.\n\n\*\*Possible values\*\*: `COD` (cash on delivery), `CVS` (convenience store), `Other` (Any payment method other than COD or CVS).",

"required": false,

"type": "array",

"items": {

"type": "string"

}

},

{

"name": "BuyerEmail",

"in": "query",

"description": "The email address of a buyer. Used to select orders that contain the specified email address.",

"required": false,

"type": "string"

},

{

"name": "SellerOrderId",

"in": "query",

"description": "An order identifier that is specified by the seller. Used to select only the orders that match the order identifier. If `SellerOrderId` is specified, then `FulfillmentChannels`, `OrderStatuses`, `PaymentMethod`, `LastUpdatedAfter`, LastUpdatedBefore, and `BuyerEmail` cannot be specified.",

"required": false,

"type": "string"

},

{

"name": "MaxResultsPerPage",

"in": "query",

"description": "A number that indicates the maximum number of orders that can be returned per page. Value must be 1 - 100. Default 100.",

"required": false,

"type": "integer"

},

{

"name": "EasyShipShipmentStatuses",

"in": "query",

"description": "A list of `EasyShipShipmentStatus` values. Used to select Easy Ship orders with statuses that match the specified values. If `EasyShipShipmentStatus` is specified, only Amazon Easy Ship orders are returned.\n\n\*\*Possible values:\*\*\n- `PendingSchedule` (The package is awaiting the schedule for pick-up.)\n- `PendingPickUp` (Amazon has not yet picked up the package from the seller.)\n- `PendingDropOff` (The seller will deliver the package to the carrier.)\n- `LabelCanceled` (The seller canceled the pickup.)\n- `PickedUp` (Amazon has picked up the package from the seller.)\n- `DroppedOff` (The package is delivered to the carrier by the seller.)\n- `AtOriginFC` (The packaged is at the origin fulfillment center.)\n- `AtDestinationFC` (The package is at the destination fulfillment center.)\n- `Delivered` (The package has been delivered.)\n- `RejectedByBuyer` (The package has been rejected by the buyer.)\n- `Undeliverable` (The package cannot be delivered.)\n- `ReturningToSeller` (The package was not delivered and is being returned to the seller.)\n- `ReturnedToSeller` (The package was not delivered and was returned to the seller.)\n- `Lost` (The package is lost.)\n- `OutForDelivery` (The package is out for delivery.)\n- `Damaged` (The package was damaged by the carrier.)",

"required": false,

"type": "array",

"items": {

"type": "string"

}

},

{

"name": "ElectronicInvoiceStatuses",

"in": "query",

"description": "A list of `ElectronicInvoiceStatus` values. Used to select orders with electronic invoice statuses that match the specified values.\n\n\*\*Possible values:\*\*\n- `NotRequired` (Electronic invoice submission is not required for this order.)\n- `NotFound` (The electronic invoice was not submitted for this order.)\n- `Processing` (The electronic invoice is being processed for this order.)\n- `Errored` (The last submitted electronic invoice was rejected for this order.)\n- `Accepted` (The last submitted electronic invoice was submitted and accepted.)",

"required": false,

"type": "array",

"items": {

"type": "string"

}

},

{

"name": "NextToken",

"in": "query",

"description": "A string token returned in the response of your previous request.",

"required": false,

"type": "string"

},

{

"name": "AmazonOrderIds",

"in": "query",

"description": "A list of `AmazonOrderId` values. An `AmazonOrderId` is an Amazon-defined order identifier, in 3-7-7 format.",

"required": false,

"type": "array",

"items": {

"type": "string"

},

"maxItems": 50

},

{

"name": "ActualFulfillmentSupplySourceId",

"in": "query",

"description": "The `sourceId` of the location from where you want the order fulfilled.",

"required": false,

"type": "string"

},

{

"name": "IsISPU",

"in": "query",

"description": "When true, this order is marked to be picked up from a store rather than delivered.",

"required": false,

"type": "boolean"

},

{

"name": "StoreChainStoreId",

"in": "query",

"description": "The store chain store identifier. Linked to a specific store in a store chain.",

"required": false,

"type": "string"

},

{

"name": "EarliestDeliveryDateBefore",

"in": "query",

"description": "Use this date to select orders with a earliest delivery date before (or at) a specified time. The date must be in [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) format.",

"required": false,

"type": "string"

},

{

"name": "EarliestDeliveryDateAfter",

"in": "query",

"description": "Use this date to select orders with a earliest delivery date after (or at) a specified time. The date must be in [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) format.",

"required": false,

"type": "string"

},

{

"name": "LatestDeliveryDateBefore",

"in": "query",

"description": "Use this date to select orders with a latest delivery date before (or at) a specified time. The date must be in [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) format.",

"required": false,

"type": "string"

},

{

"name": "LatestDeliveryDateAfter",

"in": "query",

"description": "Use this date to select orders with a latest delivery date after (or at) a specified time. The date must be in [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) format.",

"required": false,

"type": "string"

}

],

"responses": {

"200": {

"description": "Success.",

"schema": {

"$ref": "#/definitions/GetOrdersResponse"

},

"examples": {

"application/json": {

"payload": {

"NextToken": "2YgYW55IGNhcm5hbCBwbGVhc3VyZS4",

"Orders": [

{

"AmazonOrderId": "902-3159896-1390916",

"PurchaseDate": "2017-01-20T19:49:35Z",

"LastUpdateDate": "2017-01-20T19:49:35Z",

"OrderStatus": "Pending",

"FulfillmentChannel": "SellerFulfilled",

"NumberOfItemsShipped": 0,

"NumberOfItemsUnshipped": 0,

"PaymentMethod": "Other",

"PaymentMethodDetails": [

"CreditCard",

"GiftCerificate"

],

"MarketplaceId": "ATVPDKIKX0DER",

"ShipmentServiceLevelCategory": "Standard",

"OrderType": "StandardOrder",

"EarliestShipDate": "2017-01-20T19:51:16Z",

"LatestShipDate": "2017-01-25T19:49:35Z",

"IsBusinessOrder": false,

"IsPrime": false,

"IsAccessPointOrder": false,

"IsGlobalExpressEnabled": false,

"IsPremiumOrder": false,

"IsSoldByAB": false,

"IsIBA": false,

"ShippingAddress": {

"Name": "Michigan address",

"AddressLine1": "1 Cross St.",

"City": "Canton",

"StateOrRegion": "MI",

"PostalCode": "48817",

"CountryCode": "US"

},

"BuyerInfo": {

"BuyerEmail": "user@example.com",

"BuyerName": "John Doe",

"BuyerTaxInfo": {

"CompanyLegalName": "A Company Name"

},

"PurchaseOrderNumber": "1234567890123"

}

}

]

}

}

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"CreatedAfter": {

"value": "TEST\_CASE\_200"

},

"MarketplaceIds": {

"value": [

"ATVPDKIKX0DER"

]

}

}

},

"response": {

"payload": {

"CreatedBefore": "1.569521782042E9",

"Orders": [

{

"AmazonOrderId": "902-1845936-5435065",

"PurchaseDate": "1970-01-19T03:58:30Z",

"LastUpdateDate": "1970-01-19T03:58:32Z",

"OrderStatus": "Unshipped",

"FulfillmentChannel": "MFN",

"SalesChannel": "Amazon.com",

"ShipServiceLevel": "Std US D2D Dom",

"OrderTotal": {

"CurrencyCode": "USD",

"Amount": "11.01"

},

"NumberOfItemsShipped": 0,

"NumberOfItemsUnshipped": 1,

"PaymentMethod": "Other",

"PaymentMethodDetails": [

"Standard"

],

"IsReplacementOrder": false,

"MarketplaceId": "ATVPDKIKX0DER",

"ShipmentServiceLevelCategory": "Standard",

"OrderType": "StandardOrder",

"EarliestShipDate": "1970-01-19T03:59:27Z",

"LatestShipDate": "1970-01-19T04:05:13Z",

"EarliestDeliveryDate": "1970-01-19T04:06:39Z",

"LatestDeliveryDate": "1970-01-19T04:15:17Z",

"IsBusinessOrder": false,

"IsPrime": false,

"IsGlobalExpressEnabled": false,

"IsPremiumOrder": false,

"IsSoldByAB": false,

"IsIBA": false,

"DefaultShipFromLocationAddress": {

"Name": "MFNIntegrationTestMerchant",

"AddressLine1": "2201 WESTLAKE AVE",

"City": "SEATTLE",

"StateOrRegion": "WA",

"PostalCode": "98121-2778",

"CountryCode": "US",

"Phone": "+1 480-386-0930 ext. 73824",

"AddressType": "Commercial"

},

"FulfillmentInstruction": {

"FulfillmentSupplySourceId": "sampleSupplySourceId"

},

"IsISPU": false,

"IsAccessPointOrder": false,

"AutomatedShippingSettings": {

"HasAutomatedShippingSettings": false

},

"EasyShipShipmentStatus": "PendingPickUp",

"ElectronicInvoiceStatus": "NotRequired"

},

{

"AmazonOrderId": "902-8745147-1934268",

"PurchaseDate": "1970-01-19T03:58:30Z",

"LastUpdateDate": "1970-01-19T03:58:32Z",

"OrderStatus": "Unshipped",

"FulfillmentChannel": "MFN",

"SalesChannel": "Amazon.com",

"ShipServiceLevel": "Std US D2D Dom",

"OrderTotal": {

"CurrencyCode": "USD",

"Amount": "11.01"

},

"NumberOfItemsShipped": 0,

"NumberOfItemsUnshipped": 1,

"PaymentMethod": "Other",

"PaymentMethodDetails": [

"Standard"

],

"IsReplacementOrder": false,

"MarketplaceId": "ATVPDKIKX0DER",

"ShipmentServiceLevelCategory": "Standard",

"OrderType": "StandardOrder",

"EarliestShipDate": "1970-01-19T03:59:27Z",

"LatestShipDate": "1970-01-19T04:05:13Z",

"EarliestDeliveryDate": "1970-01-19T04:06:39Z",

"LatestDeliveryDate": "1970-01-19T04:15:17Z",

"IsBusinessOrder": false,

"IsPrime": false,

"IsAccessPointOrder": false,

"IsGlobalExpressEnabled": false,

"IsPremiumOrder": false,

"IsSoldByAB": false,

"IsIBA": false,

"EasyShipShipmentStatus": "PendingPickUp",

"ElectronicInvoiceStatus": "NotRequired"

}

]

}

}

},

{

"request": {

"parameters": {

"CreatedAfter": {

"value": "TEST\_CASE\_200\_NEXT\_TOKEN"

},

"MarketplaceIds": {

"value": [

"ATVPDKIKX0DER"

]

}

}

},

"response": {

"payload": {

"NextToken": "2YgYW55IGNhcm5hbCBwbGVhc3VyZS4",

"Orders": [

{

"AmazonOrderId": "902-3159896-1390916",

"PurchaseDate": "2017-01-20T19:49:35Z",

"LastUpdateDate": "2017-01-20T19:49:35Z",

"OrderStatus": "Pending",

"FulfillmentChannel": "SellerFulfilled",

"NumberOfItemsShipped": 0,

"NumberOfItemsUnshipped": 0,

"PaymentMethod": "Other",

"PaymentMethodDetails": [

"CreditCard",

"GiftCerificate"

],

"MarketplaceId": "ATVPDKIKX0DER",

"ShipmentServiceLevelCategory": "Standard",

"OrderType": "StandardOrder",

"EasyShipShipmentStatus": "PendingPickUp",

"ElectronicInvoiceStatus": "NotRequired",

"EarliestShipDate": "2017-01-20T19:51:16Z",

"LatestShipDate": "2017-01-25T19:49:35Z",

"IsBusinessOrder": false,

"IsPrime": false,

"IsAccessPointOrder": false,

"IsGlobalExpressEnabled": false,

"IsPremiumOrder": false,

"IsSoldByAB": false,

"IsIBA": false

}

]

}

}

},

{

"request": {

"parameters": {

"CreatedAfter": {

"value": "TEST\_CASE\_200\_NEXT\_TOKEN"

},

"MarketplaceIds": {

"value": [

"ATVPDKIKX0DER"

]

},

"NextToken": {

"value": "2YgYW55IGNhcm5hbCBwbGVhc3VyZS4"

}

}

},

"response": {

"payload": {

"Orders": [

{

"AmazonOrderId": "902-3159896-1390916",

"PurchaseDate": "2017-01-20T19:49:35Z",

"LastUpdateDate": "2017-01-20T19:49:35Z",

"OrderStatus": "Pending",

"FulfillmentChannel": "SellerFulfilled",

"NumberOfItemsShipped": 0,

"NumberOfItemsUnshipped": 0,

"PaymentMethod": "Other",

"PaymentMethodDetails": [

"CreditCard",

"GiftCerificate"

],

"MarketplaceId": "ATVPDKIKX0DER",

"ShipmentServiceLevelCategory": "Standard",

"OrderType": "StandardOrder",

"EasyShipShipmentStatus": "PendingPickUp",

"ElectronicInvoiceStatus": "NotRequired",

"EarliestShipDate": "2017-01-20T19:51:16Z",

"LatestShipDate": "2017-01-25T19:49:35Z",

"IsBusinessOrder": false,

"IsPrime": false,

"IsAccessPointOrder": false,

"IsGlobalExpressEnabled": false,

"IsPremiumOrder": false,

"IsSoldByAB": false,

"IsIBA": false

}

]

}

}

}

]

}

},

"400": {

"description": "Request has missing or invalid parameters and cannot be parsed.",

"schema": {

"$ref": "#/definitions/GetOrdersResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"CreatedAfter": {

"value": "TEST\_CASE\_400"

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Invalid Input"

}

]

}

}

]

}

},

"403": {

"description": "Indicates access to the resource is forbidden. Possible reasons include Access Denied, Unauthorized, Expired Token, or Invalid Signature.",

"schema": {

"$ref": "#/definitions/GetOrdersResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"404": {

"description": "The resource specified does not exist.",

"schema": {

"$ref": "#/definitions/GetOrdersResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"429": {

"description": "The frequency of requests was greater than allowed.",

"schema": {

"$ref": "#/definitions/GetOrdersResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"500": {

"description": "An unexpected condition occurred that prevented the server from fulfilling the request.",

"schema": {

"$ref": "#/definitions/GetOrdersResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"503": {

"description": "Temporary overloading or maintenance of the server.",

"schema": {

"$ref": "#/definitions/GetOrdersResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

}

}

}

},

"/orders/v0/orders/{orderId}": {

"get": {

"tags": [

"ordersV0"

],

"description": "Returns the order that you specify.\n\n\*\*Usage Plan:\*\*\n\n| Rate (requests per second) | Burst |\n| ---- | ---- |\n| 0.5 | 30 |\n\nThe `x-amzn-RateLimit-Limit` response header contains the usage plan rate limits for the operation, when available. The preceding table contains the default rate and burst values for this operation. Selling partners whose business demands require higher throughput might have higher rate and burst values than those shown here. For more information, refer to [Usage Plans and Rate Limits](https://developer-docs.amazon.com/sp-api/docs/usage-plans-and-rate-limits-in-the-sp-api).",

"operationId": "getOrder",

"parameters": [

{

"name": "orderId",

"in": "path",

"description": "An Amazon-defined order identifier, in 3-7-7 format.",

"required": true,

"type": "string"

}

],

"responses": {

"200": {

"description": "Success.",

"schema": {

"$ref": "#/definitions/GetOrderResponse"

},

"examples": {

"application/json": {

"payload": {

"AmazonOrderId": "902-3159896-1390916",

"PurchaseDate": "2017-01-20T19:49:35Z",

"LastUpdateDate": "2017-01-20T19:49:35Z",

"OrderStatus": "Pending",

"FulfillmentChannel": "SellerFulfilled",

"NumberOfItemsShipped": 0,

"NumberOfItemsUnshipped": 0,

"PaymentMethod": "Other",

"PaymentMethodDetails": [

"CreditCard",

"GiftCerificate"

],

"MarketplaceId": "ATVPDKIKX0DER",

"ShipmentServiceLevelCategory": "Standard",

"OrderType": "StandardOrder",

"EarliestShipDate": "2017-01-20T19:51:16Z",

"LatestShipDate": "2017-01-25T19:49:35Z",

"IsBusinessOrder": false,

"IsPrime": false,

"IsGlobalExpressEnabled": false,

"IsPremiumOrder": false,

"IsSoldByAB": false,

"IsIBA": false,

"DefaultShipFromLocationAddress": {

"Name": "MFNIntegrationTestMerchant",

"AddressLine1": "2201 WESTLAKE AVE",

"City": "SEATTLE",

"StateOrRegion": "WA",

"PostalCode": "98121-2778",

"CountryCode": "US",

"Phone": "+1 480-386-0930 ext. 73824",

"AddressType": "Commercial"

},

"FulfillmentInstruction": {

"FulfillmentSupplySourceId": "sampleSupplySourceId"

},

"IsISPU": false,

"IsAccessPointOrder": false,

"ShippingAddress": {

"Name": "Michigan address",

"AddressLine1": "1 Cross St.",

"City": "Canton",

"StateOrRegion": "MI",

"PostalCode": "48817",

"CountryCode": "US"

},

"BuyerInfo": {

"BuyerEmail": "user@example.com",

"BuyerName": "John Doe",

"BuyerTaxInfo": {

"CompanyLegalName": "A Company Name"

},

"PurchaseOrderNumber": "1234567890123"

},

"AutomatedShippingSettings": {

"HasAutomatedShippingSettings": false

}

}

}

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_200"

}

}

},

"response": {

"payload": {

"AmazonOrderId": "902-1845936-5435065",

"PurchaseDate": "1970-01-19T03:58:30Z",

"LastUpdateDate": "1970-01-19T03:58:32Z",

"OrderStatus": "Unshipped",

"FulfillmentChannel": "MFN",

"SalesChannel": "Amazon.com",

"ShipServiceLevel": "Std US D2D Dom",

"OrderTotal": {

"CurrencyCode": "USD",

"Amount": "11.01"

},

"NumberOfItemsShipped": 0,

"NumberOfItemsUnshipped": 1,

"PaymentMethod": "Other",

"PaymentMethodDetails": [

"Standard"

],

"IsReplacementOrder": false,

"MarketplaceId": "ATVPDKIKX0DER",

"ShipmentServiceLevelCategory": "Standard",

"OrderType": "StandardOrder",

"EarliestShipDate": "1970-01-19T03:59:27Z",

"LatestShipDate": "1970-01-19T04:05:13Z",

"EarliestDeliveryDate": "1970-01-19T04:06:39Z",

"LatestDeliveryDate": "1970-01-19T04:15:17Z",

"IsBusinessOrder": false,

"IsPrime": false,

"IsGlobalExpressEnabled": false,

"IsPremiumOrder": false,

"IsSoldByAB": false,

"IsIBA": false,

"DefaultShipFromLocationAddress": {

"Name": "MFNIntegrationTestMerchant",

"AddressLine1": "2201 WESTLAKE AVE",

"City": "SEATTLE",

"StateOrRegion": "WA",

"PostalCode": "98121-2778",

"CountryCode": "US",

"Phone": "+1 480-386-0930 ext. 73824",

"AddressType": "Commercial"

},

"FulfillmentInstruction": {

"FulfillmentSupplySourceId": "sampleSupplySourceId"

},

"IsISPU": false,

"IsAccessPointOrder": false,

"AutomatedShippingSettings": {

"HasAutomatedShippingSettings": false

},

"EasyShipShipmentStatus": "PendingPickUp",

"ElectronicInvoiceStatus": "NotRequired"

}

}

},

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_IBA\_200"

}

}

},

"response": {

"payload": {

"AmazonOrderId": "921-3175655-0452641",

"PurchaseDate": "2019-05-07T15:42:57.058Z",

"LastUpdateDate": "2019-05-08T21:59:59Z",

"OrderStatus": "Shipped",

"FulfillmentChannel": "AFN",

"SalesChannel": "Amazon.de",

"ShipServiceLevel": "Standard",

"OrderTotal": {

"CurrencyCode": "EUR",

"Amount": "100.00"

},

"NumberOfItemsShipped": 1,

"NumberOfItemsUnshipped": 0,

"PaymentMethod": "Other",

"PaymentMethodDetails": [

"Invoice"

],

"IsReplacementOrder": false,

"MarketplaceId": "A1PA6795UKMFR9",

"ShipmentServiceLevelCategory": "Standard",

"OrderType": "StandardOrder",

"EarliestShipDate": "1970-01-19T03:59:27Z",

"LatestShipDate": "2019-05-08T21:59:59Z",

"EarliestDeliveryDate": "2019-05-10T21:59:59Z",

"LatestDeliveryDate": "2019-05-12T21:59:59Z",

"IsBusinessOrder": true,

"IsPrime": false,

"IsGlobalExpressEnabled": false,

"IsPremiumOrder": false,

"IsSoldByAB": true,

"IsIBA": true,

"DefaultShipFromLocationAddress": {

"Name": "MFNIntegrationTestMerchant",

"AddressLine1": "2201 WESTLAKE AVE",

"City": "SEATTLE",

"StateOrRegion": "WA",

"PostalCode": "98121-2778",

"CountryCode": "US",

"Phone": "+1 480-386-0930 ext. 73824",

"AddressType": "Commercial"

},

"FulfillmentInstruction": {

"FulfillmentSupplySourceId": "sampleSupplySourceId"

},

"IsISPU": false,

"IsAccessPointOrder": false,

"AutomatedShippingSettings": {

"HasAutomatedShippingSettings": false

},

"EasyShipShipmentStatus": "PendingPickUp",

"ElectronicInvoiceStatus": "NotRequired"

}

}

}

]

}

},

"400": {

"description": "Request has missing or invalid parameters and cannot be parsed.",

"schema": {

"$ref": "#/definitions/GetOrderResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_400"

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Invalid Input"

}

]

}

}

]

}

},

"403": {

"description": "Indicates access to the resource is forbidden. Possible reasons include Access Denied, Unauthorized, Expired Token, or Invalid Signature.",

"schema": {

"$ref": "#/definitions/GetOrderResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"404": {

"description": "The resource specified does not exist.",

"schema": {

"$ref": "#/definitions/GetOrderResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"429": {

"description": "The frequency of requests was greater than allowed.",

"schema": {

"$ref": "#/definitions/GetOrderResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"500": {

"description": "An unexpected condition occurred that prevented the server from fulfilling the request.",

"schema": {

"$ref": "#/definitions/GetOrderResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"503": {

"description": "Temporary overloading or maintenance of the server.",

"schema": {

"$ref": "#/definitions/GetOrderResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

}

}

}

},

"/orders/v0/orders/{orderId}/buyerInfo": {

"get": {

"tags": [

"ordersV0"

],

"description": "Returns buyer information for the order that you specify.\n\n\*\*Usage Plan:\*\*\n\n| Rate (requests per second) | Burst |\n| ---- | ---- |\n| 0.5 | 30 |\n\nThe `x-amzn-RateLimit-Limit` response header contains the usage plan rate limits for the operation, when available. The preceding table contains the default rate and burst values for this operation. Selling partners whose business demands require higher throughput might have higher rate and burst values than those shown here. For more information, refer to [Usage Plans and Rate Limits](https://developer-docs.amazon.com/sp-api/docs/usage-plans-and-rate-limits-in-the-sp-api).",

"operationId": "getOrderBuyerInfo",

"parameters": [

{

"name": "orderId",

"in": "path",

"description": "The Amazon order identifier in 3-7-7 format.",

"required": true,

"type": "string"

}

],

"responses": {

"200": {

"description": "Success.",

"schema": {

"$ref": "#/definitions/GetOrderBuyerInfoResponse"

},

"examples": {

"application/json": {

"payload": {

"AmazonOrderId": "902-3159896-1390916",

"BuyerEmail": "user@example.com",

"BuyerName": "John Smith",

"BuyerTaxInfo": {

"CompanyLegalName": "Company Name"

},

"PurchaseOrderNumber": "1234567890123"

}

}

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_200"

}

}

},

"response": {

"payload": {

"AmazonOrderId": "902-1845936-5435065",

"BuyerEmail": "fzyrv6gwkhbb15c@example.com",

"BuyerName": "MFNIntegrationTestMerchant"

}

}

}

]

}

},

"400": {

"description": "Request has missing or invalid parameters and cannot be parsed.",

"schema": {

"$ref": "#/definitions/GetOrderBuyerInfoResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_400"

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Invalid Input"

}

]

}

}

]

}

},

"403": {

"description": "Indicates access to the resource is forbidden. Possible reasons include Access Denied, Unauthorized, Expired Token, or Invalid Signature.",

"schema": {

"$ref": "#/definitions/GetOrderBuyerInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"404": {

"description": "The resource specified does not exist.",

"schema": {

"$ref": "#/definitions/GetOrderBuyerInfoResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"429": {

"description": "The frequency of requests was greater than allowed.",

"schema": {

"$ref": "#/definitions/GetOrderBuyerInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"500": {

"description": "An unexpected condition occurred that prevented the server from fulfilling the request.",

"schema": {

"$ref": "#/definitions/GetOrderBuyerInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"503": {

"description": "Temporary overloading or maintenance of the server.",

"schema": {

"$ref": "#/definitions/GetOrderBuyerInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

}

}

}

},

"/orders/v0/orders/{orderId}/address": {

"get": {

"tags": [

"ordersV0"

],

"description": "Returns the shipping address for the order that you specify.\n\n\*\*Usage Plan:\*\*\n\n| Rate (requests per second) | Burst |\n| ---- | ---- |\n| 0.5 | 30 |\n\nThe `x-amzn-RateLimit-Limit` response header contains the usage plan rate limits for the operation, when available. The preceding table contains the default rate and burst values for this operation. Selling partners whose business demands require higher throughput might have higher rate and burst values than those shown here. For more information, refer to [Usage Plans and Rate Limits](https://developer-docs.amazon.com/sp-api/docs/usage-plans-and-rate-limits-in-the-sp-api).",

"operationId": "getOrderAddress",

"parameters": [

{

"name": "orderId",

"in": "path",

"description": "The Amazon order identifier in 3-7-7 format.",

"required": true,

"type": "string"

}

],

"responses": {

"200": {

"description": "Success.",

"schema": {

"$ref": "#/definitions/GetOrderAddressResponse"

},

"examples": {

"application/json": {

"payload": {

"AmazonOrderId": "902-3159896-1390916",

"ShippingAddress": {

"Name": "Michigan address",

"AddressLine1": "1 cross st",

"City": "Canton",

"StateOrRegion": "MI",

"PostalCode": "48817",

"CountryCode": "US"

}

}

}

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_200"

}

}

},

"response": {

"payload": {

"AmazonOrderId": "902-1845936-5435065",

"ShippingAddress": {

"Name": "MFNIntegrationTestMerchant",

"AddressLine1": "2201 WESTLAKE AVE",

"City": "SEATTLE",

"StateOrRegion": "WA",

"PostalCode": "98121-2778",

"CountryCode": "US",

"Phone": "+1 480-386-0930 ext. 73824",

"AddressType": "Commercial"

}

}

}

}

]

}

},

"400": {

"description": "Request has missing or invalid parameters and cannot be parsed.",

"schema": {

"$ref": "#/definitions/GetOrderAddressResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_400"

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Invalid Input"

}

]

}

}

]

}

},

"403": {

"description": "Indicates access to the resource is forbidden. Possible reasons include Access Denied, Unauthorized, Expired Token, or Invalid Signature.",

"schema": {

"$ref": "#/definitions/GetOrderAddressResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"404": {

"description": "The resource specified does not exist.",

"schema": {

"$ref": "#/definitions/GetOrderAddressResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"429": {

"description": "The frequency of requests was greater than allowed.",

"schema": {

"$ref": "#/definitions/GetOrderAddressResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"500": {

"description": "An unexpected condition occurred that prevented the server from fulfilling the request.",

"schema": {

"$ref": "#/definitions/GetOrderAddressResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"503": {

"description": "Temporary overloading or maintenance of the server.",

"schema": {

"$ref": "#/definitions/GetOrderAddressResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

}

}

}

},

"/orders/v0/orders/{orderId}/orderItems": {

"get": {

"tags": [

"ordersV0"

],

"description": "Returns detailed order item information for the order that you specify. If `NextToken` is provided, it's used to retrieve the next page of order items.\n\n\_\_Note\_\_: When an order is in the Pending state (the order has been placed but payment has not been authorized), the getOrderItems operation does not return information about pricing, taxes, shipping charges, gift status or promotions for the order items in the order. After an order leaves the Pending state (this occurs when payment has been authorized) and enters the Unshipped, Partially Shipped, or Shipped state, the getOrderItems operation returns information about pricing, taxes, shipping charges, gift status and promotions for the order items in the order.\n\n\*\*Usage Plan:\*\*\n\n| Rate (requests per second) | Burst |\n| ---- | ---- |\n| 0.5 | 30 |\n\nThe `x-amzn-RateLimit-Limit` response header contains the usage plan rate limits for the operation, when available. The preceding table contains the default rate and burst values for this operation. Selling partners whose business demands require higher throughput might have higher rate and burst values than those shown here. For more information, refer to [Usage Plans and Rate Limits](https://developer-docs.amazon.com/sp-api/docs/usage-plans-and-rate-limits-in-the-sp-api).",

"operationId": "getOrderItems",

"parameters": [

{

"name": "orderId",

"in": "path",

"description": "An Amazon-defined order identifier, in 3-7-7 format.",

"required": true,

"type": "string"

},

{

"name": "NextToken",

"in": "query",

"description": "A string token returned in the response of your previous request.",

"required": false,

"type": "string"

}

],

"responses": {

"200": {

"description": "Success.",

"schema": {

"$ref": "#/definitions/GetOrderItemsResponse"

},

"examples": {

"application/json": {

"payload": {

"AmazonOrderId": "903-1671087-0812628",

"NextToken": "2YgYW55IGNhcm5hbCBwbGVhc3VyZS4",

"OrderItems": [

{

"ASIN": "BT0093TELA",

"OrderItemId": "68828574383266",

"SellerSKU": "CBA\_OTF\_1",

"Title": "Example item name",

"QuantityOrdered": 1,

"QuantityShipped": 1,

"PointsGranted": {

"PointsNumber": 10,

"PointsMonetaryValue": {

"CurrencyCode": "JPY",

"Amount": "10.00"

}

},

"ItemPrice": {

"CurrencyCode": "JPY",

"Amount": "25.99"

},

"ShippingPrice": {

"CurrencyCode": "JPY",

"Amount": "1.26"

},

"ScheduledDeliveryEndDate": "2013-09-09T01:30:00Z",

"ScheduledDeliveryStartDate": "2013-09-07T02:00:00Z",

"CODFee": {

"CurrencyCode": "JPY",

"Amount": "10.00"

},

"CODFeeDiscount": {

"CurrencyCode": "JPY",

"Amount": "1.00"

},

"PriceDesignation": "BusinessPrice",

"BuyerInfo": {

"BuyerCustomizedInfo": {

"CustomizedURL": "https://zme-caps.amazon.com/t/bR6qHkzSOxuB/J8nbWhze0Bd3DkajkOdY-XQbWkFralegp2sr\_QZiKEE/1"

},

"GiftMessageText": "For you!",

"GiftWrapPrice": {

"CurrencyCode": "GBP",

"Amount": "41.99"

},

"GiftWrapLevel": "Classic"

},

"BuyerRequestedCancel": {

"IsBuyerRequestedCancel": "true",

"BuyerCancelReason": "Found cheaper somewhere else."

},

"SerialNumbers": ["854"]

},

{

"ASIN": "BCTU1104UEFB",

"OrderItemId": "79039765272157",

"SellerSKU": "CBA\_OTF\_5",

"Title": "Example item name",

"QuantityOrdered": 2,

"ItemPrice": {

"CurrencyCode": "JPY",

"Amount": "17.95"

},

"PromotionIds": [

"FREESHIP"

],

"ConditionId": "Used",

"ConditionSubtypeId": "Mint",

"ConditionNote": "Example ConditionNote",

"PriceDesignation": "BusinessPrice",

"BuyerInfo": {

"BuyerCustomizedInfo": {

"CustomizedURL": "https://zme-caps.amazon.com/t/bR6qHkzSOxuB/J8nbWhze0Bd3DkajkOdY-XQbWkFralegp2sr\_QZiKEE/1"

},

"GiftMessageText": "For you!",

"GiftWrapPrice": {

"CurrencyCode": "JPY",

"Amount": "1.99"

},

"GiftWrapLevel": "Classic"

},

"BuyerRequestedCancel": {

"IsBuyerRequestedCancel": "true",

"BuyerCancelReason": "Found cheaper somewhere else."

}

}

]

}

}

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_200"

}

}

},

"response": {

"payload": {

"AmazonOrderId": "902-1845936-5435065",

"OrderItems": [

{

"ASIN": "B00551Q3CS",

"OrderItemId": "05015851154158",

"SellerSKU": "NABetaASINB00551Q3CS",

"Title": "B00551Q3CS [Card Book]",

"QuantityOrdered": 1,

"QuantityShipped": 0,

"ProductInfo": {

"NumberOfItems": "1"

},

"ItemPrice": {

"CurrencyCode": "USD",

"Amount": "10.00"

},

"ItemTax": {

"CurrencyCode": "USD",

"Amount": "1.01"

},

"PromotionDiscount": {

"CurrencyCode": "USD",

"Amount": "0.00"

},

"IsGift": "false",

"ConditionId": "New",

"ConditionSubtypeId": "New",

"IsTransparency": false,

"SerialNumberRequired": false,

"IossNumber": "",

"DeemedResellerCategory": "IOSS",

"StoreChainStoreId": "ISPU\_StoreId",

"BuyerRequestedCancel": {

"IsBuyerRequestedCancel": "true",

"BuyerCancelReason": "Found cheaper somewhere else."

}

}

]

}

}

}

]

}

},

"400": {

"description": "Request has missing or invalid parameters and cannot be parsed.",

"schema": {

"$ref": "#/definitions/GetOrderItemsResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_400"

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Invalid Input"

}

]

}

}

]

}

},

"403": {

"description": "Indicates access to the resource is forbidden. Possible reasons include Access Denied, Unauthorized, Expired Token, or Invalid Signature.",

"schema": {

"$ref": "#/definitions/GetOrderItemsResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"404": {

"description": "The resource specified does not exist.",

"schema": {

"$ref": "#/definitions/GetOrderItemsResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"429": {

"description": "The frequency of requests was greater than allowed.",

"schema": {

"$ref": "#/definitions/GetOrderItemsResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"500": {

"description": "An unexpected condition occurred that prevented the server from fulfilling the request.",

"schema": {

"$ref": "#/definitions/GetOrderItemsResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"503": {

"description": "Temporary overloading or maintenance of the server.",

"schema": {

"$ref": "#/definitions/GetOrderItemsResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

}

}

}

},

"/orders/v0/orders/{orderId}/orderItems/buyerInfo": {

"get": {

"tags": [

"ordersV0"

],

"description": "Returns buyer information for the order items in the order that you specify.\n\n\*\*Usage Plan:\*\*\n\n| Rate (requests per second) | Burst |\n| ---- | ---- |\n| 0.5 | 30 |\n\nThe `x-amzn-RateLimit-Limit` response header contains the usage plan rate limits for the operation, when available. The preceding table contains the default rate and burst values for this operation. Selling partners whose business demands require higher throughput might have higher rate and burst values than those shown here. For more information, refer to [Usage Plans and Rate Limits](https://developer-docs.amazon.com/sp-api/docs/usage-plans-and-rate-limits-in-the-sp-api).",

"operationId": "getOrderItemsBuyerInfo",

"parameters": [

{

"name": "orderId",

"in": "path",

"description": "An Amazon-defined order identifier, in 3-7-7 format.",

"required": true,

"type": "string"

},

{

"name": "NextToken",

"in": "query",

"description": "A string token returned in the response of your previous request.",

"required": false,

"type": "string"

}

],

"responses": {

"200": {

"description": "Success.",

"schema": {

"$ref": "#/definitions/GetOrderItemsBuyerInfoResponse"

},

"examples": {

"application/json": {

"payload": {

"OrderItemId": "903-1671087-0812628",

"BuyerCustomizedInfo": {

"CustomizedURL": "https://zme-caps.amazon.com/t/bR6qHkzSOxuB/J8nbWhze0Bd3DkajkOdY-XQbWkFralegp2sr\_QZiKEE/1"

},

"GiftMessageText": "For you!",

"GiftWrapPrice": {

"CurrencyCode": "JPY",

"Amount": "1.99"

},

"GiftWrapLevel": "Classic"

}

}

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_200"

}

}

},

"response": {

"payload": {

"AmazonOrderId": "902-1845936-5435065",

"OrderItems": [

{

"BuyerCustomizedInfo": {

"CustomizedURL": "https://zme-caps.amazon.com/t/bR6qHkzSOxuB/J8nbWhze0Bd3DkajkOdY-XQbWkFralegp2sr\_QZiKEE/1"

},

"GiftMessageText": "Et toi!",

"GiftWrapPrice": {

"CurrencyCode": "JPY",

"Amount": "1.99"

},

"GiftWrapLevel": "Classic"

}

]

}

}

}

]

}

},

"400": {

"description": "Request has missing or invalid parameters and cannot be parsed.",

"schema": {

"$ref": "#/definitions/GetOrderItemsBuyerInfoResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_400"

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Invalid Input"

}

]

}

}

]

}

},

"403": {

"description": "Indicates access to the resource is forbidden. Possible reasons include Access Denied, Unauthorized, Expired Token, or Invalid Signature.",

"schema": {

"$ref": "#/definitions/GetOrderItemsBuyerInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"404": {

"description": "The resource specified does not exist.",

"schema": {

"$ref": "#/definitions/GetOrderItemsBuyerInfoResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"429": {

"description": "The frequency of requests was greater than allowed.",

"schema": {

"$ref": "#/definitions/GetOrderItemsBuyerInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"500": {

"description": "An unexpected condition occurred that prevented the server from fulfilling the request.",

"schema": {

"$ref": "#/definitions/GetOrderItemsBuyerInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"503": {

"description": "Temporary overloading or maintenance of the server.",

"schema": {

"$ref": "#/definitions/GetOrderItemsBuyerInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

}

}

}

},

"/orders/v0/orders/{orderId}/shipment": {

"post": {

"tags": [

"shipment"

],

"description": "Update the shipment status for an order that you specify.\n\n\*\*Usage Plan:\*\*\n\n| Rate (requests per second) | Burst |\n| ---- | ---- |\n| 5 | 15 |\n\nThe `x-amzn-RateLimit-Limit` response header contains the usage plan rate limits for the operation, when available. The preceding table contains the default rate and burst values for this operation. Selling partners whose business demands require higher throughput might have higher rate and burst values than those shown here. For more information, refer to [Usage Plans and Rate Limits](https://developer-docs.amazon.com/sp-api/docs/usage-plans-and-rate-limits-in-the-sp-api).",

"operationId": "updateShipmentStatus",

"parameters": [

{

"name": "orderId",

"in": "path",

"description": "An Amazon-defined order identifier, in 3-7-7 format.",

"required": true,

"type": "string"

},

{

"name": "payload",

"in": "body",

"description": "The request body for the `updateShipmentStatus` operation.",

"required": true,

"schema": {

"$ref": "#/definitions/UpdateShipmentStatusRequest"

}

}

],

"responses": {

"204": {

"description": "Success.",

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {}

},

"response": {}

}

]

},

"headers": {

"x-amzn-RateLimit-Limit": {

"description": "Your rate limit (requests per second) for this operation.",

"type": "string"

},

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

}

},

"400": {

"description": "Request has missing or invalid parameters and cannot be parsed.",

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"body": {

"value": {

"marketplaceId": "1",

"shipmentStatus": "ReadyForPickup"

}

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Marketplace ID is not defined",

"details": "1001"

}

]

}

}

]

},

"headers": {

"x-amzn-RateLimit-Limit": {

"description": "Your rate limit (requests per second) for this operation.",

"type": "string"

},

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateShipmentStatusErrorResponse"

}

},

"403": {

"description": "Indicates that access to the resource is forbidden. Possible reasons include Access Denied, Unauthorized, Expired Token, or Invalid Signature.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateShipmentStatusErrorResponse"

}

},

"404": {

"description": "The resource specified does not exist.",

"headers": {

"x-amzn-RateLimit-Limit": {

"description": "Your rate limit (requests per second) for this operation.",

"type": "string"

},

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateShipmentStatusErrorResponse"

}

},

"413": {

"description": "The request size exceeded the maximum accepted size.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateShipmentStatusErrorResponse"

}

},

"415": {

"description": "The request payload is in an unsupported format.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateShipmentStatusErrorResponse"

}

},

"429": {

"description": "The frequency of requests was greater than allowed.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateShipmentStatusErrorResponse"

}

},

"500": {

"description": "An unexpected condition occurred that prevented the server from fulfilling the request.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateShipmentStatusErrorResponse"

}

},

"503": {

"description": "Temporary overloading or maintenance of the server.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateShipmentStatusErrorResponse"

}

}

}

}

},

"/orders/v0/orders/{orderId}/regulatedInfo": {

"get": {

"tags": [

"ordersV0"

],

"description": "Returns regulated information for the order that you specify.\n\n\*\*Usage Plan:\*\*\n\n| Rate (requests per second) | Burst |\n| ---- | ---- |\n| 0.5 | 30 |\n\nThe `x-amzn-RateLimit-Limit` response header contains the usage plan rate limits for the operation, when available. The preceding table contains the default rate and burst values for this operation. Selling partners whose business demands require higher throughput might have higher rate and burst values than those shown here. For more information, refer to [Usage Plans and Rate Limits](https://developer-docs.amazon.com/sp-api/docs/usage-plans-and-rate-limits-in-the-sp-api).",

"operationId": "getOrderRegulatedInfo",

"parameters": [

{

"name": "orderId",

"in": "path",

"description": "The Amazon order identifier in 3-7-7 format.",

"required": true,

"type": "string"

}

],

"responses": {

"200": {

"description": "Success.",

"schema": {

"$ref": "#/definitions/GetOrderRegulatedInfoResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "902-3159896-1390916"

}

}

},

"response": {

"payload": {

"AmazonOrderId": "902-3159896-1390916",

"RequiresDosageLabel": false,

"RegulatedInformation": {

"Fields": [

{

"FieldId": "pet\_prescription\_name",

"FieldLabel": "Name",

"FieldType": "Text",

"FieldValue": "Ruffus"

},

{

"FieldId": "pet\_prescription\_species",

"FieldLabel": "Species",

"FieldType": "Text",

"FieldValue": "Dog"

}

]

},

"RegulatedOrderVerificationStatus": {

"Status": "Pending",

"RequiresMerchantAction": true,

"ValidRejectionReasons": [

{

"RejectionReasonId": "shield\_pom\_vps\_reject\_product",

"RejectionReasonDescription": "This medicine is not suitable for your pet."

},

{

"RejectionReasonId": "shield\_pom\_vps\_reject\_age",

"RejectionReasonDescription": "Your pet is too young for this medicine."

},

{

"RejectionReasonId": "shield\_pom\_vps\_reject\_incorrect\_weight",

"RejectionReasonDescription": "Your pet's weight does not match ordered size."

}

]

}

}

}

},

{

"request": {

"parameters": {

"orderId": {

"value": "902-2592119-3531015"

}

}

},

"response": {

"payload": {

"AmazonOrderId": "902-2592119-3531015",

"RequiresDosageLabel": false,

"RegulatedInformation": {

"Fields": [

{

"FieldId": "pets\_rx\_scName",

"FieldLabel": "Pet name",

"FieldType": "Text",

"FieldValue": "Woofer"

},

{

"FieldId": "pets\_rx\_scType",

"FieldLabel": "Pet type",

"FieldType": "Text",

"FieldValue": "Dog"

},

{

"FieldId": "pets\_rx\_scBreed",

"FieldLabel": "Pet breed",

"FieldType": "Text",

"FieldValue": "Husky"

},

{

"FieldId": "pets\_rx\_scGender",

"FieldLabel": "Pet gender",

"FieldType": "Text",

"FieldValue": "Female"

},

{

"FieldId": "pets\_rx\_scDateOfBirth",

"FieldLabel": "Pet Birth Date",

"FieldType": "Text",

"FieldValue": "2016-05-01"

},

{

"FieldId": "pets\_rx\_scWeight",

"FieldLabel": "Weight",

"FieldType": "Text",

"FieldValue": "12"

},

{

"FieldId": "pets\_rx\_scWeightUnit",

"FieldLabel": "Weight Unit",

"FieldType": "Text",

"FieldValue": "Pound"

},

{

"FieldId": "pets\_rx\_scHasAllergies",

"FieldLabel": "Does your pet have allergies?",

"FieldType": "Text",

"FieldValue": "False"

},

{

"FieldId": "pets\_rx\_scTakesAdditionalMedications",

"FieldLabel": "Is your pet on any other medication?",

"FieldType": "Text",

"FieldValue": "False"

},

{

"FieldId": "pets\_rx\_scHasOtherProblems",

"FieldLabel": "Any pet health problems?",

"FieldType": "Text",

"FieldValue": "False"

},

{

"FieldId": "pets\_rx\_scSourceClinicId",

"FieldLabel": "Source Clinic ID",

"FieldType": "Text",

"FieldValue": "Clinic-1234"

},

{

"FieldId": "pets\_rx\_scVetClinicName",

"FieldLabel": "Vet Clinic Name",

"FieldType": "Text",

"FieldValue": "Test Clinic"

},

{

"FieldId": "pets\_rx\_scVetClinicCity",

"FieldLabel": "Vet Clinic City",

"FieldType": "Text",

"FieldValue": "Seattle"

},

{

"FieldId": "pets\_rx\_scVetClinicState",

"FieldLabel": "Vet Clinic State",

"FieldType": "Text",

"FieldValue": "WA"

},

{

"FieldId": "pets\_rx\_scVetClinicZipCode",

"FieldLabel": "Vet Clinic Zip Code",

"FieldType": "Text",

"FieldValue": "98000"

},

{

"FieldId": "pets\_rx\_scVetClinicPhoneNumber",

"FieldLabel": "Vet Clinic Phone Number",

"FieldType": "Text",

"FieldValue": "2060000000"

}

]

},

"RegulatedOrderVerificationStatus": {

"Status": "Pending",

"RequiresMerchantAction": true,

"ValidRejectionReasons": [

{

"RejectionReasonId": "pets\_rx\_sc\_incorrect\_product",

"RejectionReasonDescription": "Canceled order due to veterinarian indicating wrong product ordered"

}, {

"RejectionReasonId": "pets\_rx\_sc\_no\_vcpr",

"RejectionReasonDescription": "Canceled order due to veterinarian indicating they do not have you as their client"

}, {

"RejectionReasonId": "pets\_rx\_sc\_visit\_required",

"RejectionReasonDescription": "Canceled order due to veterinarian indicating they need to see your pet for an appointment"

},{

"RejectionReasonId": "pets\_rx\_sc\_wrx\_required",

"RejectionReasonDescription": "Canceled order due to veterinarian policy requiring you pick up a written prescription and mail to pharmacy"

}, {

"RejectionReasonId": "pets\_rx\_sc\_other",

"RejectionReasonDescription": "Canceled order due to prescription denied - contact your vetinarian"

}, {

"RejectionReasonId": "pets\_rx\_sc\_therapy\_change",

"RejectionReasonDescription": "Canceled order due to a change in therapy, follow up with your veterinarian"

}, {

"RejectionReasonId": "pets\_rx\_sc\_wrong\_weight",

"RejectionReasonDescription": "Canceled order due to incorrect pet weight on file, update weight and replace order or order correct product"

}, {

"RejectionReasonId": "pets\_rx\_sc\_early\_refill",

"RejectionReasonDescription": "Canceled due to refilling prescription order too soon"

}, {

"RejectionReasonId": "pets\_rx\_sc\_wrong\_species",

"RejectionReasonDescription": "Canceled due to ordering for wrong pet species, replace order for correct product"

}, {

"RejectionReasonId": "pets\_rx\_sc\_duplicate",

"RejectionReasonDescription": "Canceled due to duplicate order identified"

}, {

"RejectionReasonId": "pets\_rx\_sc\_invalid\_rx",

"RejectionReasonDescription": "Canceled due to not receiving a valid prescription"

}, {

"RejectionReasonId": "pets\_rx\_sc\_address\_validation\_error",

"RejectionReasonDescription": "Canceled due to a non-verified address, correct address and replace order"

}, {

"RejectionReasonId": "pets\_rx\_sc\_no\_clinic\_match",

"RejectionReasonDescription": "Canceled due to no valid clinic match, provide complete and accurate details for vet clinic and replace order"

}, {

"RejectionReasonId": "pets\_rx\_sc\_pharmacist\_canceled",

"RejectionReasonDescription": "Order canceled by pharmacy"

}

],

"ValidVerificationDetails": [

{

"VerificationDetailType": "prescriptionDetail",

"ValidVerificationStatuses": [

"Approved"

]

}

]

}

}

}

}

]

},

"examples": {

"PendingOrder": {

"payload": {

"AmazonOrderId": "902-3159896-1390916",

"RequiresDosageLabel": false,

"RegulatedInformation": {

"Fields": [

{

"FieldId": "pet\_prescription\_name",

"FieldLabel": "Name",

"FieldType": "Text",

"FieldValue": "Ruffus"

},

{

"FieldId": "pet\_prescription\_species",

"FieldLabel": "Species",

"FieldType": "Text",

"FieldValue": "Dog"

}

]

},

"RegulatedOrderVerificationStatus": {

"Status": "Pending",

"RequiresMerchantAction": true,

"ValidRejectionReasons": [

{

"RejectionReasonId": "shield\_pom\_vps\_reject\_product",

"RejectionReasonDescription": "This medicine is not suitable for your pet."

},

{

"RejectionReasonId": "shield\_pom\_vps\_reject\_age",

"RejectionReasonDescription": "Your pet is too young for this medicine."

},

{

"RejectionReasonId": "shield\_pom\_vps\_reject\_incorrect\_weight",

"RejectionReasonDescription": "Your pet's weight does not match ordered size."

}

]

}

}

},

"ApprovedOrder": {

"payload": {

"AmazonOrderId": "902-3159896-1390916",

"RequiresDosageLabel": false,

"RegulatedInformation": {

"Fields": [

{

"FieldId": "pet\_prescription\_name",

"FieldLabel": "Name",

"FieldType": "Text",

"FieldValue": "Ruffus"

},

{

"FieldId": "pet\_prescription\_species",

"FieldLabel": "Species",

"FieldType": "Text",

"FieldValue": "Dog"

}

]

},

"RegulatedOrderVerificationStatus": {

"Status": "Approved",

"RequiresMerchantAction": false,

"ValidRejectionReasons": [],

"ExternalReviewerId": "externalId",

"ReviewDate": "1970-01-19T03:59:27Z"

}

}

},

"RejectedOrder": {

"payload": {

"AmazonOrderId": "902-3159896-1390916",

"RequiresDosageLabel": false,

"RegulatedInformation": {

"Fields": [

{

"FieldId": "pet\_prescription\_name",

"FieldLabel": "Name",

"FieldType": "Text",

"FieldValue": "Ruffus"

},

{

"FieldId": "pet\_prescription\_species",

"FieldLabel": "Species",

"FieldType": "Text",

"FieldValue": "Dog"

}

]

},

"RegulatedOrderVerificationStatus": {

"Status": "Rejected",

"RequiresMerchantAction": false,

"RejectionReason": {

"RejectionReasonId": "shield\_pom\_vps\_reject\_species",

"RejectionReasonDescription": "This medicine is not suitable for this type of pet."

},

"ValidRejectionReasons": [],

"ExternalReviewerId": "externalId",

"ReviewDate": "1970-01-19T03:59:27Z"

}

}

}

}

},

"400": {

"description": "Request has missing or invalid parameters and cannot be parsed.",

"schema": {

"$ref": "#/definitions/GetOrderRegulatedInfoResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "TEST\_CASE\_400"

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Invalid Input"

}

]

}

}

]

}

},

"403": {

"description": "Indicates access to the resource is forbidden. Possible reasons include Access Denied, Unauthorized, Expired Token, or Invalid Signature.",

"schema": {

"$ref": "#/definitions/GetOrderRegulatedInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"404": {

"description": "The resource specified does not exist.",

"schema": {

"$ref": "#/definitions/GetOrderRegulatedInfoResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"429": {

"description": "The frequency of requests was greater than allowed.",

"schema": {

"$ref": "#/definitions/GetOrderRegulatedInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"500": {

"description": "An unexpected condition occurred that prevented the server from fulfilling the request.",

"schema": {

"$ref": "#/definitions/GetOrderRegulatedInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"503": {

"description": "Temporary overloading or maintenance of the server.",

"schema": {

"$ref": "#/definitions/GetOrderRegulatedInfoResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

}

}

},

"patch": {

"tags": [

"ordersV0"

],

"description": "Updates (approves or rejects) the verification status of an order containing regulated products.\n\n\*\*Usage Plan:\*\*\n\n| Rate (requests per second) | Burst |\n| ---- | ---- |\n| 0.5 | 30 |\n\nThe `x-amzn-RateLimit-Limit` response header contains the usage plan rate limits for the operation, when available. The preceding table contains the default rate and burst values for this operation. Selling partners whose business demands require higher throughput might have higher rate and burst values than those shown here. For more information, refer to [Usage Plans and Rate Limits](https://developer-docs.amazon.com/sp-api/docs/usage-plans-and-rate-limits-in-the-sp-api).",

"operationId": "updateVerificationStatus",

"parameters": [

{

"name": "orderId",

"in": "path",

"description": "The Amazon order identifier in 3-7-7 format.",

"required": true,

"type": "string"

},

{

"name": "payload",

"in": "body",

"description": "The request body for the `updateVerificationStatus` operation.",

"required": true,

"schema": {

"$ref": "#/definitions/UpdateVerificationStatusRequest"

}

}

],

"responses": {

"204": {

"description": "Success.",

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "902-3159896-1390916"

},

"body": {

"value": {

"regulatedOrderVerificationStatus": {

"status": "Rejected",

"externalReviewerId": "reviewer1234",

"rejectionReasonId": "shield\_pom\_vps\_reject\_incorrect\_weight"

}

}

}

}

},

"response": {}

},

{

"request": {

"parameters": {

"orderId": {

"value": "902-3159896-1390916"

},

"body": {

"value": {

"regulatedOrderVerificationStatus": {

"externalReviewerId": "reviewer1234",

"verificationDetails": {

"prescriptionDetail":

{

"prescriptionId": "Rx-1234",

"expirationDate": "2024-01-01T00:00:00Z",

"writtenQuantity": 3,

"totalRefillsAuthorized": 10,

"usageInstructions": "Take one per day by mouth with food",

"refillsRemaining": 10,

"clinicId": "ABC-1234"

}

}

}

}

}

}

},

"response": {}

}

]

},

"headers": {

"x-amzn-RateLimit-Limit": {

"description": "Your rate limit (requests per second) for this operation.",

"type": "string"

},

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

}

},

"400": {

"description": "Request has missing or invalid parameters and cannot be parsed.",

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "902-3159896-1390916"

},

"body": {

"value": {

"regulatedOrderVerificationStatus": {

"status": "Rejected"

}

}

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Missing request parameter: rejectionReasonId."

},

{

"code": "InvalidInput",

"message": "Missing request parameter: externalReviewerId."

}

]

}

},

{

"request": {

"parameters": {

"orderId": {

"value": "902-3159896-1390916"

},

"body": {

"value": {

"regulatedOrderVerificationStatus": {

"status": "Cancelled",

"externalReviewerId": "reviewer1234"

}

}

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Invalid request parameter `status`. Must be one of [Approved, Rejected]."

}

]

}

},

{

"request": {

"parameters": {

"orderId": {

"value": "902-3159896-1390916"

},

"body": {

"value": {

"regulatedOrderVerificationStatus": {

"status": "Rejected",

"rejectionReasonId": "shield\_pom\_vps\_reject\_incorrect\_weight",

"verificationDetails": {

"prescriptionDetail":

{

"prescriptionId": "Rx-1234",

"expirationDate": "2024-01-01T00:00:00Z",

"writtenQuantity": 3,

"totalRefillsAuthorized": 10,

"usageInstructions": "Take one per day by mouth with food",

"refillsRemaining": 10,

"clinicId": "ABC-1234"

}

}

}

}

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Verification Detail `prescriptionDetail` is not supported when order is in Rejected status."

},

{

"code": "InvalidInput",

"message": "Missing request parameter: externalReviewerId."

}

]

}

},

{

"request": {

"parameters": {

"orderId": {

"value": "902-3159896-1390916"

},

"body": {

"value": {

"regulatedOrderVerificationStatus": {

"externalReviewerId": "reviewer1234",

"verificationDetails": {

"prescriptionDetail":

{

"prescriptionId": "Rx-1234",

"expirationDate": "2024-01-01T00:00:00Z",

"writtenQuantity": 3,

"totalRefillsAuthorized": 10,

"usageInstructions": "Take one per day by mouth with food",

"refillsRemaining": 10

}

}

}

}

}

}

},

"response": {

"errors": [

{

"code": "InvalidInput",

"message": "Missing required parameter(s) from prescriptionDetail value: clinicId"

}

]

}

}

]

},

"headers": {

"x-amzn-RateLimit-Limit": {

"description": "Your rate limit (requests per second) for this operation.",

"type": "string"

},

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateVerificationStatusErrorResponse"

}

},

"403": {

"description": "Indicates that access to the resource is forbidden. Possible reasons include Access Denied, Unauthorized, Expired Token, or Invalid Signature.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateVerificationStatusErrorResponse"

}

},

"404": {

"description": "The resource specified does not exist.",

"headers": {

"x-amzn-RateLimit-Limit": {

"description": "Your rate limit (requests per second) for this operation.",

"type": "string"

},

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateVerificationStatusErrorResponse"

}

},

"413": {

"description": "The request size exceeded the maximum accepted size.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateVerificationStatusErrorResponse"

}

},

"415": {

"description": "The request payload is in an unsupported format.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateVerificationStatusErrorResponse"

}

},

"429": {

"description": "The frequency of requests was greater than allowed.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateVerificationStatusErrorResponse"

}

},

"500": {

"description": "An unexpected condition occurred that prevented the server from fulfilling the request.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateVerificationStatusErrorResponse"

}

},

"503": {

"description": "Temporary overloading or maintenance of the server.",

"headers": {

"x-amzn-RequestId": {

"description": "Unique request reference identifier.",

"type": "string"

}

},

"schema": {

"$ref": "#/definitions/UpdateVerificationStatusErrorResponse"

}

}

}

}

},

"/orders/v0/orders/{orderId}/shipmentConfirmation": {

"post": {

"tags": [

"ordersV0"

],

"description": "Updates the shipment confirmation status for a specified order.\n\n\*\*Usage Plan:\*\*\n\n| Rate (requests per second) | Burst |\n| ---- | ---- |\n| 2 | 10 |\n\nThe `x-amzn-RateLimit-Limit` response header contains the usage plan rate limits for the operation, when available. The preceding table contains the default rate and burst values for this operation. Selling partners whose business demands require higher throughput might have higher rate and burst values than those shown here. For more information, refer to [Usage Plans and Rate Limits](https://developer-docs.amazon.com/sp-api/docs/usage-plans-and-rate-limits-in-the-sp-api).",

"operationId": "confirmShipment",

"parameters": [

{

"name": "orderId",

"in": "path",

"description": "An Amazon-defined order identifier, in 3-7-7 format.",

"required": true,

"type": "string"

},

{

"name": "payload",

"in": "body",

"description": "Request body of `confirmShipment`.",

"required": true,

"schema": {

"$ref": "#/definitions/ConfirmShipmentRequest"

}

}

],

"responses": {

"204": {

"description": "Success.",

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "902-1106328-1059050"

},

"body": {

"value": {

"marketplaceId": "ATVPDKIKX0DER",

"packageDetail": {

"packageReferenceId": "1",

"carrierCode": "FedEx",

"carrierName": "FedEx",

"shippingMethod": "FedEx Ground",

"trackingNumber": "112345678",

"shipDate": "2022-02-11T01:00:00.000Z",

"shipFromSupplySourceId": "057d3fcc-b750-419f-bbcd-4d340c60c430",

"orderItems":[

{

"orderItemId": "79039765272157",

"quantity": 1,

"transparencyCodes": [

"09876543211234567890"

]

}

]

}

}

}

}

},

"response": {

}

}

]

}

},

"400": {

"description": "Request has missing or invalid parameters and cannot be parsed.",

"schema": {

"$ref": "#/definitions/ConfirmShipmentErrorResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

},

"x-amzn-api-sandbox": {

"static": [

{

"request": {

"parameters": {

"orderId": {

"value": "902-1106328-1059050"

},

"body": {

"value": {

"marketplaceId": "ATVPDKIKX0DER",

"packageDetail": {

"packageReferenceId": "1",

"carrierCode": "FedEx",

"carrierName": "FedEx",

"shippingMethod": "FedEx Ground",

"trackingNumber": "112345678",

"shipDate": "02/21/2022",

"shipFromSupplySourceId": "057d3fcc-b750-419f-bbcd-4d340c60c430",

"orderItems":[

{

"orderItemId": "79039765272157",

"quantity": 1,

"transparencyCodes": [

"09876543211234567890"

]

}

]

}

}

}

}

},

"response": {

"errors": [

{

"code": "Invalid Input",

"message": "Invalid Input"

}

]

}

}

]

}

},

"401": {

"description": "The request's Authorization header is not formatted correctly or does not contain a valid token.",

"schema": {

"$ref": "#/definitions/ConfirmShipmentErrorResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"403": {

"description": "Indicates that access to the resource is forbidden. Possible reasons include Access Denied, Unauthorized, Expired Token, or Invalid Signature.",

"schema": {

"$ref": "#/definitions/ConfirmShipmentErrorResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"404": {

"description": "The specified resource does not exist.",

"schema": {

"$ref": "#/definitions/ConfirmShipmentErrorResponse"

},

"headers": {

"x-amzn-RateLimit-Limit": {

"type": "string",

"description": "Your rate limit (requests per second) for this operation."

},

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"429": {

"description": "The frequency of requests was greater than allowed.",

"schema": {

"$ref": "#/definitions/ConfirmShipmentErrorResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"500": {

"description": "An unexpected condition occurred that prevented the server from fulfilling the request.",

"schema": {

"$ref": "#/definitions/ConfirmShipmentErrorResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

},

"503": {

"description": "Temporary overloading or maintenance of the server.",

"schema": {

"$ref": "#/definitions/ConfirmShipmentErrorResponse"

},

"headers": {

"x-amzn-RequestId": {

"type": "string",

"description": "Unique request reference identifier."

}

}

}

}

}

}

},

"definitions": {

"UpdateShipmentStatusRequest": {

"description": "The request body for the `updateShipmentStatus` operation.",

"type": "object",

"properties": {

"marketplaceId": {

"$ref": "#/definitions/MarketplaceId"

},

"shipmentStatus": {

"$ref": "#/definitions/ShipmentStatus"

},

"orderItems": {

"$ref": "#/definitions/OrderItems"

}

},

"required": [

"marketplaceId",

"shipmentStatus"

]

},

"UpdateVerificationStatusRequest": {

"description": "The request body for the `updateVerificationStatus` operation.",

"type": "object",

"properties": {

"regulatedOrderVerificationStatus": {

"description": "The updated values of the `VerificationStatus` field.",

"$ref": "#/definitions/UpdateVerificationStatusRequestBody"

}

},

"required": [

"regulatedOrderVerificationStatus"

]

},

"UpdateVerificationStatusRequestBody": {

"description": "The updated values of the `VerificationStatus` field.",

"type": "object",

"properties": {

"status": {

"description": "The new verification status of the order.",

"$ref": "#/definitions/VerificationStatus"

},

"externalReviewerId": {

"description": "The identifier of the order's regulated information reviewer.",

"type": "string"

},

"rejectionReasonId": {

"description": "The unique identifier of the rejection reason used for rejecting the order's regulated information. Only required if the new status is rejected.",

"type": "string"

},

"verificationDetails": {

"description": "Additional information regarding the verification of the order.",

"$ref": "#/definitions/VerificationDetails"

}

},

"required": [

"externalReviewerId"

]

},

"MarketplaceId": {

"description": "The unobfuscated marketplace identifier.",

"type": "string"

},

"ShipmentStatus": {

"description": "The shipment status to apply.",

"type": "string",

"enum": [

"ReadyForPickup",

"PickedUp",

"RefusedPickup"

],

"x-docgen-enum-table-extension": [

{

"value": "ReadyForPickup",

"description": "Ready for pickup."

},

{

"value": "PickedUp",

"description": "Picked up."

},

{

"value": "RefusedPickup",

"description": "Refused pickup."

}

]

},

"OrderItems": {

"description": "For partial shipment status updates, the list of order items and quantities to be updated.",

"type": "array",

"items": {

"type": "object",

"properties": {

"orderItemId": {

"description": "The order item's unique identifier.",

"type": "string"

},

"quantity": {

"type": "integer",

"description": "The quantity for which to update the shipment status."

}

}

}

},

"UpdateShipmentStatusErrorResponse": {

"type": "object",

"properties": {

"errors": {

"description": "One or more unexpected errors occurred during the `UpdateShipmentStatus` operation.",

"$ref": "#/definitions/ErrorList"

}

},

"description": "The error response schema for the `UpdateShipmentStatus` operation."

},

"UpdateVerificationStatusErrorResponse": {

"type": "object",

"properties": {

"errors": {

"description": "One or more unexpected errors occurred during the `UpdateVerificationStatus` operation.",

"$ref": "#/definitions/ErrorList"

}

},

"description": "The error response schema for the `UpdateVerificationStatus` operation."

},

"GetOrdersResponse": {

"type": "object",

"properties": {

"payload": {

"description": "The payload for the `getOrders` operation.",

"$ref": "#/definitions/OrdersList"

},

"errors": {

"description": "One or more unexpected errors occurred during the `getOrders` operation.",

"$ref": "#/definitions/ErrorList"

}

},

"description": "The response schema for the `getOrders` operation."

},

"GetOrderResponse": {

"type": "object",

"properties": {

"payload": {

"description": "The payload for the `getOrder` operation.",

"$ref": "#/definitions/Order"

},

"errors": {

"description": "One or more unexpected errors occurred during the `getOrder` operation.",

"$ref": "#/definitions/ErrorList"

}

},

"description": "The response schema for the `getOrder` operation."

},

"GetOrderBuyerInfoResponse": {

"type": "object",

"properties": {

"payload": {

"description": "The payload for the `getOrderBuyerInfo` operation.",

"$ref": "#/definitions/OrderBuyerInfo"

},

"errors": {

"description": "One or more unexpected errors occurred during the `getOrderBuyerInfo` operation.",

"$ref": "#/definitions/ErrorList"

}

},

"description": "The response schema for the `getOrderBuyerInfo` operation."

},

"GetOrderRegulatedInfoResponse": {

"type": "object",

"properties": {

"payload": {

"description": "The payload for the `getOrderRegulatedInfo` operation.",

"$ref": "#/definitions/OrderRegulatedInfo"

},

"errors": {

"description": "One or more unexpected errors occurred during the `getOrderRegulatedInfo` operation.",

"$ref": "#/definitions/ErrorList"

}

},

"description": "The response schema for the `getOrderRegulatedInfo` operation."

},

"GetOrderAddressResponse": {

"type": "object",

"properties": {

"payload": {

"description": "The payload for the `getOrderAddress` operations.",

"$ref": "#/definitions/OrderAddress"

},

"errors": {

"description": "One or more unexpected errors occurred during the `getOrderAddress` operation.",

"$ref": "#/definitions/ErrorList"

}

},

"description": "The response schema for the `getOrderAddress` operation."

},

"GetOrderItemsResponse": {

"type": "object",

"properties": {

"payload": {

"description": "The payload for the `getOrderItems` operation.",

"$ref": "#/definitions/OrderItemsList"

},

"errors": {

"description": "One or more unexpected errors occurred during the `getOrderItems` operation.",

"$ref": "#/definitions/ErrorList"

}

},

"description": "The response schema for the `getOrderItems` operation."

},

"GetOrderItemsBuyerInfoResponse": {

"type": "object",

"properties": {

"payload": {

"description": "The payload for the `getOrderItemsBuyerInfo` operation.",

"$ref": "#/definitions/OrderItemsBuyerInfoList"

},

"errors": {

"description": "One or more unexpected errors occurred during the `getOrderItemsBuyerInfo` operation.",

"$ref": "#/definitions/ErrorList"

}

},

"description": "The response schema for the `getOrderItemsBuyerInfo` operation."

},

"OrdersList": {

"type": "object",

"required": [

"Orders"

],

"properties": {

"Orders": {

"$ref": "#/definitions/OrderList"

},

"NextToken": {

"type": "string",

"description": "When present and not empty, pass this string token in the next request to return the next response page."

},

"LastUpdatedBefore": {

"type": "string",

"description": "Use this date to select orders that were last updated before (or at) a specified time. An update is defined as any change in order status, including the creation of a new order. Includes updates made by Amazon and by the seller. Use [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) format for all dates."

},

"CreatedBefore": {

"type": "string",

"description": "Use this date to select orders created before (or at) a specified time. Only orders placed before the specified time are returned. The date must be in [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) format."

}

},

"description": "A list of orders along with additional information to make subsequent API calls."

},

"OrderList": {

"type": "array",

"description": "A list of orders.",

"items": {

"$ref": "#/definitions/Order"

}

},

"Order": {

"type": "object",

"required": [

"AmazonOrderId",

"LastUpdateDate",

"OrderStatus",

"PurchaseDate"

],

"properties": {

"AmazonOrderId": {

"type": "string",

"description": "An Amazon-defined order identifier, in 3-7-7 format."

},

"SellerOrderId": {

"type": "string",

"description": "A seller-defined order identifier."

},

"PurchaseDate": {

"type": "string",

"description": "The date when the order was created."

},

"LastUpdateDate": {

"type": "string",

"description": "The date when the order was last updated.\n\n\_\_Note\_\_: `LastUpdateDate` is returned with an incorrect date for orders that were last updated before 2009-04-01."

},

"OrderStatus": {

"type": "string",

"description": "The current order status.",

"enum": [

"Pending",

"Unshipped",

"PartiallyShipped",

"Shipped",

"Canceled",

"Unfulfillable",

"InvoiceUnconfirmed",

"PendingAvailability"

],

"x-docgen-enum-table-extension": [

{

"value": "Pending",

"description": "The order has been placed but payment has not been authorized. The order is not ready for shipment. Note that for orders with `OrderType = Standard`, the initial order status is Pending. For orders with `OrderType = Preorder`, the initial order status is `PendingAvailability`, and the order passes into the Pending status when the payment authorization process begins."

},

{

"value": "Unshipped",

"description": "Payment has been authorized and order is ready for shipment, but no items in the order have been shipped."

},

{

"value": "PartiallyShipped",

"description": "One or more (but not all) items in the order have been shipped."

},

{

"value": "Shipped",

"description": "All items in the order have been shipped."

},

{

"value": "Canceled",

"description": "The order was canceled."

},

{

"value": "Unfulfillable",

"description": "The order cannot be fulfilled. This state applies only to Amazon-fulfilled orders that were not placed on Amazon's retail web site."

},

{

"value": "InvoiceUnconfirmed",

"description": "All items in the order have been shipped. The seller has not yet given confirmation to Amazon that the invoice has been shipped to the buyer."

},

{

"value": "PendingAvailability",

"description": "This status is available for pre-orders only. The order has been placed, payment has not been authorized, and the release date for the item is in the future. The order is not ready for shipment."

}

]

},

"FulfillmentChannel": {

"type": "string",

"description": "Whether the order was fulfilled by Amazon (`AFN`) or by the seller (`MFN`).",

"enum": [

"MFN",

"AFN"

],

"x-docgen-enum-table-extension": [

{

"value": "MFN",

"description": "Fulfilled by the seller."

},

{

"value": "AFN",

"description": "Fulfilled by Amazon."

}

]

},

"SalesChannel": {

"type": "string",

"description": "The sales channel for the first item in the order."

},

"OrderChannel": {

"type": "string",

"description": "The order channel for the first item in the order."

},

"ShipServiceLevel": {

"type": "string",

"description": "The order's shipment service level."

},

"OrderTotal": {

"description": "The total charge for this order.",

"$ref": "#/definitions/Money"

},

"NumberOfItemsShipped": {

"type": "integer",

"description": "The number of items shipped."

},

"NumberOfItemsUnshipped": {

"type": "integer",

"description": "The number of items unshipped."

},

"PaymentExecutionDetail": {

"description": "Information about sub-payment methods for a cash-on-delivery (COD) order.\n\n\_\_Note\_\_: For a COD order that is paid for using one sub-payment method, one `PaymentExecutionDetailItem` object is returned, with `PaymentExecutionDetailItem`/`PaymentMethod = COD`. For a COD order that is paid for using multiple sub-payment methods, two or more `PaymentExecutionDetailItem` objects are returned.",

"$ref": "#/definitions/PaymentExecutionDetailItemList"

},

"PaymentMethod": {

"type": "string",

"description": "The payment method for the order. This property is limited to COD and CVS payment methods. Unless you need the specific COD payment information provided by the `PaymentExecutionDetailItem` object, we recommend using the `PaymentMethodDetails` property to get payment method information.",

"enum": [

"COD",

"CVS",

"Other"

],

"x-docgen-enum-table-extension": [

{

"value": "COD",

"description": "Cash on delivery."

},

{

"value": "CVS",

"description": "Convenience store."

},

{

"value": "Other",

"description": "A payment method other than COD and CVS."

}

]

},

"PaymentMethodDetails": {

"description": "A list of payment methods for the order.",

"$ref": "#/definitions/PaymentMethodDetailItemList"

},

"MarketplaceId": {

"type": "string",

"description": "The identifier for the marketplace where the order was placed."

},

"ShipmentServiceLevelCategory": {

"type": "string",

"description": "The shipment service level category for the order.\n\n\*\*Possible values\*\*: `Expedited`, `FreeEconomy`, `NextDay`, `Priority`, `SameDay`, `SecondDay`, `Scheduled`, and `Standard`."

},

"EasyShipShipmentStatus": {

"description": "The status of the Amazon Easy Ship order. This property is only included for Amazon Easy Ship orders.",

"$ref": "#/definitions/EasyShipShipmentStatus"

},

"CbaDisplayableShippingLabel": {

"type": "string",

"description": "Custom ship label for Checkout by Amazon (CBA)."

},

"OrderType": {

"type": "string",

"description": "The order's type.",

"enum": [

"StandardOrder",

"LongLeadTimeOrder",

"Preorder",

"BackOrder",

"SourcingOnDemandOrder"

],

"x-docgen-enum-table-extension": [

{

"value": "StandardOrder",

"description": "An order that contains items for which the selling partner currently has inventory in stock."

},

{

"value": "LongLeadTimeOrder",

"description": "An order that contains items that have a long lead time to ship."

},

{

"value": "Preorder",

"description": "An order that contains items with a release date that is in the future."

},

{

"value": "BackOrder",

"description": "An order that contains items that already have been released in the market but are currently out of stock and will be available in the future."

},

{

"value": "SourcingOnDemandOrder",

"description": "A Sourcing On Demand order."

}

]

},

"EarliestShipDate": {

"type": "string",

"description": "The start of the time period within which you have committed to ship the order. In [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) date time format. Only returned for seller-fulfilled orders.\n\n\_\_Note\_\_: `EarliestShipDate` might not be returned for orders placed before February 1, 2013."

},

"LatestShipDate": {

"type": "string",

"description": "The end of the time period within which you have committed to ship the order. In [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) date time format. Only returned for seller-fulfilled orders.\n\n\_\_Note\_\_: `LatestShipDate` might not be returned for orders placed before February 1, 2013."

},

"EarliestDeliveryDate": {

"type": "string",

"description": "The start of the time period within which you have committed to fulfill the order. In [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) date time format. Only returned for seller-fulfilled orders."

},

"LatestDeliveryDate": {

"type": "string",

"description": "The end of the time period within which you have committed to fulfill the order. In [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) date time format. Only returned for seller-fulfilled orders that do not have a `PendingAvailability`, `Pending`, or `Canceled` status."

},

"IsBusinessOrder": {

"type": "boolean",

"description": "When true, the order is an Amazon Business order. An Amazon Business order is an order where the buyer is a Verified Business Buyer."

},

"IsPrime": {

"type": "boolean",

"description": "When true, the order is a seller-fulfilled Amazon Prime order."

},

"IsPremiumOrder": {

"type": "boolean",

"description": "When true, the order has a Premium Shipping Service Level Agreement. For more information about Premium Shipping orders, refer to \"Premium Shipping Options\" in the Seller Central Help for your marketplace."

},

"IsGlobalExpressEnabled": {

"type": "boolean",

"description": "When true, the order is a `GlobalExpress` order."

},

"ReplacedOrderId": {

"type": "string",

"description": "The order ID value for the order that is being replaced. Returned only if IsReplacementOrder = true."

},

"IsReplacementOrder": {

"type": "boolean",

"description": "When true, this is a replacement order."

},

"PromiseResponseDueDate": {

"type": "string",

"description": "Indicates the date by which the seller must respond to the buyer with an estimated ship date. Only returned for Sourcing on Demand orders."

},

"IsEstimatedShipDateSet": {

"type": "boolean",

"description": "When true, the estimated ship date is set for the order. Only returned for Sourcing on Demand orders."

},

"IsSoldByAB": {

"type": "boolean",

"description": "When true, the item within this order was bought and re-sold by Amazon Business EU SARL (ABEU). By buying and instantly re-selling your items, ABEU becomes the seller of record, making your inventory available for sale to customers who would not otherwise purchase from a third-party seller."

},

"IsIBA": {

"type": "boolean",

"description": "When true, the item within this order was bought and re-sold by Amazon Business EU SARL (ABEU). By buying and instantly re-selling your items, ABEU becomes the seller of record, making your inventory available for sale to customers who would not otherwise purchase from a third-party seller."

},

"DefaultShipFromLocationAddress": {

"description": "The recommended location for the seller to ship the items from. It is calculated at checkout. The seller may or may not choose to ship from this location.",

"$ref": "#/definitions/Address"

},

"BuyerInvoicePreference": {

"type": "string",

"enum": [

"INDIVIDUAL",

"BUSINESS"

],

"x-docgen-enum-table-extension": [

{

"value": "INDIVIDUAL",

"description": "Issues an individual invoice to the buyer."

},

{

"value": "BUSINESS",

"description": "Issues a business invoice to the buyer. Tax information is available in `BuyerTaxInformation`."

}

],

"description": "The buyer's invoicing preference. Sellers can use this data to issue electronic invoices for orders in Turkey.\n\n\*\*Note\*\*: This attribute is only available in the Turkey marketplace."

},

"BuyerTaxInformation": {

"description": "Contains the business invoice tax information. Sellers could use this data to issue electronic invoices for business orders in Turkey.\n\n\*\*Note\*\*:\n1. This attribute is only available in the Turkey marketplace for the orders that `BuyerInvoicePreference` is BUSINESS.\n2. The `BuyerTaxInformation` is a restricted data. Use the Restricted Data Token (RDT) and restricted SPDS roles to access this restricted data.",

"$ref": "#/definitions/BuyerTaxInformation"

},

"FulfillmentInstruction": {

"description": "Contains the instructions about the fulfillment, such as the location from where you want the order filled.",

"$ref": "#/definitions/FulfillmentInstruction"

},

"IsISPU": {

"type": "boolean",

"description": "When true, this order is marked to be picked up from a store rather than delivered."

},

"IsAccessPointOrder": {

"type": "boolean",

"description": "When true, this order is marked to be delivered to an Access Point. The access location is chosen by the customer. Access Points include Amazon Hub Lockers, Amazon Hub Counters, and pickup points operated by carriers."

},

"MarketplaceTaxInfo": {

"description": "Tax information about the marketplace where the sale took place. Sellers can use this data to issue electronic invoices for orders in Brazil.\n\n\*\*Note\*\*: This attribute is only available in the Brazil marketplace for the orders with `Pending` or `Unshipped` status.",

"$ref": "#/definitions/MarketplaceTaxInfo"

},

"SellerDisplayName": {

"type": "string",

"description": "The seller’s friendly name registered in the marketplace where the sale took place. Sellers can use this data to issue electronic invoices for orders in Brazil.\n\n\*\*Note\*\*: This attribute is only available in the Brazil marketplace for the orders with `Pending` or `Unshipped` status."

},

"ShippingAddress": {

"description": "The shipping address for the order.\n\n\*\*Note\*\*:\n1. `ShippingAddress` is only available for orders with the following status values: Unshipped, `PartiallyShipped`, Shipped and `InvoiceUnconfirmed`.\n2. The `ShippingAddress` contains restricted data. Use the Restricted Data Token (RDT) and restricted SPDS roles to access the restricted data in `ShippingAddress`. For example, `Name`, `AddressLine1`, `AddressLine2`, `AddressLine3`, `Phone`, `AddressType`, and `ExtendedFields`.",

"$ref": "#/definitions/Address"

},

"BuyerInfo": {

"description": "Buyer information.\n\n\*\*Note\*\*: The `BuyerInfo` contains restricted data. Use the Restricted Data Token (RDT) and restricted SPDS roles to access the restricted data in `BuyerInfo`. For example, `BuyerName`, `BuyerTaxInfo`, and `PurchaseOrderNumber`.",

"$ref": "#/definitions/BuyerInfo"

},

"AutomatedShippingSettings": {

"description": "Contains information regarding the Shipping Settings Automaton program, such as whether the order's shipping settings were generated automatically, and what those settings are.",

"$ref": "#/definitions/AutomatedShippingSettings"

},

"HasRegulatedItems": {

"type": "boolean",

"description": "Whether the order contains regulated items which may require additional approval steps before being fulfilled."

},

"ElectronicInvoiceStatus": {

"$ref": "#/definitions/ElectronicInvoiceStatus",

"description": "The status of the electronic invoice."

}

},

"description": "Order information."

},

"OrderBuyerInfo": {

"type": "object",

"required": [

"AmazonOrderId"

],

"properties": {

"AmazonOrderId": {

"type": "string",

"description": "An Amazon-defined order identifier, in 3-7-7 format."

},

"BuyerEmail": {

"type": "string",

"description": "The anonymized email address of the buyer."

},

"BuyerName": {

"type": "string",

"description": "The buyer name or the recipient name."

},

"BuyerCounty": {

"type": "string",

"description": "The county of the buyer.\n\n\*\*Note\*\*: This attribute is only available in the Brazil marketplace."

},

"BuyerTaxInfo": {

"description": "Tax information about the buyer. Sellers can use this data to issue electronic invoices for business orders.\n\n\*\*Note\*\*: This attribute is only available for business orders in the Brazil, Mexico and India marketplaces.",

"$ref": "#/definitions/BuyerTaxInfo"

},

"PurchaseOrderNumber": {

"type": "string",

"description": "The purchase order (PO) number entered by the buyer at checkout. Only returned for orders where the buyer entered a PO number at checkout."

}

},

"description": "Buyer information for an order."

},

"OrderRegulatedInfo": {

"description": "The order's regulated information along with its verification status.",

"type": "object",

"required": [

"AmazonOrderId",

"RegulatedInformation",

"RegulatedOrderVerificationStatus",

"RequiresDosageLabel"

],

"properties": {

"AmazonOrderId": {

"type": "string",

"description": "An Amazon-defined order identifier, in 3-7-7 format."

},

"RegulatedInformation": {

"$ref": "#/definitions/RegulatedInformation",

"description": "The regulated information collected during purchase and used to verify the order."

},

"RequiresDosageLabel": {

"type": "boolean",

"description": "When true, the order requires attaching a dosage information label when shipped."

},

"RegulatedOrderVerificationStatus": {

"$ref": "#/definitions/RegulatedOrderVerificationStatus",

"description": "The order's verification status."

}

}

},

"RegulatedOrderVerificationStatus": {

"type": "object",

"description": "The verification status of the order, along with associated approval or rejection metadata.",

"required": [

"Status",

"RequiresMerchantAction",

"ValidRejectionReasons"

],

"properties": {

"Status": {

"description": "The verification status of the order.",

"$ref": "#/definitions/VerificationStatus"

},

"RequiresMerchantAction": {

"type": "boolean",

"description": "When true, the regulated information provided in the order requires a review by the merchant."

},

"ValidRejectionReasons": {

"type": "array",

"description": "A list of valid rejection reasons that may be used to reject the order's regulated information.",

"items": {

"$ref": "#/definitions/RejectionReason"

}

},

"RejectionReason": {

"$ref": "#/definitions/RejectionReason",

"description": "The reason for rejecting the order's regulated information. Not present if the order isn't rejected."

},

"ReviewDate": {

"type": "string",

"description": "The date the order was reviewed. In [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) date time format."

},

"ExternalReviewerId": {

"type": "string",

"description": "The identifier for the order's regulated information reviewer."

},

"ValidVerificationDetails": {

"type": "array",

"description": "A list of valid verification details that may be provided and the criteria required for when the verification detail can be provided.",

"items": {

"$ref": "#/definitions/ValidVerificationDetail"

}

}

}

},

"RejectionReason": {

"type": "object",

"description": "The reason for rejecting the order's regulated information. This is only present if the order is rejected.",

"required": [

"RejectionReasonId",

"RejectionReasonDescription"

],

"properties": {

"RejectionReasonId": {

"type": "string",

"description": "The unique identifier for the rejection reason."

},

"RejectionReasonDescription": {

"type": "string",

"description": "The description of this rejection reason."

}

}

},

"PrescriptionDetail": {

"type": "object",

"required": [

"prescriptionId",

"expirationDate",

"writtenQuantity",

"totalRefillsAuthorized",

"refillsRemaining",

"clinicId",

"usageInstructions"

],

"properties": {

"prescriptionId": {

"type": "string",

"description": "The identifier for the prescription used to verify the regulated product."

},

"expirationDate": {

"type": "string",

"description": "The expiration date of the prescription used to verify the regulated product, in [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) date time format.",

"format": "date-time"

},

"writtenQuantity": {

"type": "integer",

"description": "The number of units in each fill as provided in the prescription.",

"minimum": 1

},

"totalRefillsAuthorized": {

"type": "integer",

"description": "The total number of refills written in the original prescription used to verify the regulated product. If a prescription originally had no refills, this value must be 0.",

"minimum": 0

},

"refillsRemaining": {

"type": "integer",

"description": "The number of refills remaining for the prescription used to verify the regulated product. If a prescription originally had 10 total refills, this value must be `10` for the first order, `9` for the second order, and `0` for the eleventh order. If a prescription originally had no refills, this value must be 0.",

"minimum": 0

},

"clinicId": {

"type": "string",

"description": "The identifier for the clinic which provided the prescription used to verify the regulated product."

},

"usageInstructions": {

"type": "string",

"description": "The instructions for the prescription as provided by the approver of the regulated product."

}

},

"description": "Information about the prescription that is used to verify a regulated product. This must be provided once per order and reflect the seller’s own records. Only approved orders can have prescriptions."

},

"ValidVerificationDetail": {

"type": "object",

"required": [

"VerificationDetailType",

"ValidVerificationStatuses"

],

"properties": {

"VerificationDetailType": {

"type": "string",

"description": "A supported type of verification detail. The type indicates which verification detail could be shared while updating the regulated order. Valid value: `prescriptionDetail`."

},

"ValidVerificationStatuses": {

"type": "array",

"description": "A list of valid verification statuses where the associated verification detail type may be provided. For example, if the value of this field is [\"Approved\"], calls to provide the associated verification detail will fail for orders with a `VerificationStatus` of `Pending`, `Rejected`, `Expired`, or `Cancelled`.",

"items": {

"$ref": "#/definitions/VerificationStatus"

}

}

},

"description": "The types of verification details that may be provided for the order and the criteria required for when the type of verification detail can be provided. The types of verification details allowed depend on the type of regulated product and will not change order to order."

},

"VerificationDetails": {

"type": "object",

"properties": {

"prescriptionDetail": {

"$ref": "#/definitions/PrescriptionDetail",

"description": "Information regarding the prescription tied to the order."

}

},

"description": "Additional information related to the verification of a regulated order."

},

"VerificationStatus": {

"type": "string",

"description": "The verification status of the order.",

"enum": [

"Pending",

"Approved",

"Rejected",

"Expired",

"Cancelled"

],

"x-docgen-enum-table-extension": [

{

"value": "Pending",

"description": "The order is pending approval. Note that the approval might be needed from someone other than the merchant as determined by the `RequiresMerchantAction` property."

},

{

"value": "Approved",

"description": "The order's regulated information has been reviewed and approved."

},

{

"value": "Rejected",

"description": "The order's regulated information has been reviewed and rejected."

},

{

"value": "Expired",

"description": "The time to review the order's regulated information has expired."

},

{

"value": "Cancelled",

"description": "The order was cancelled by the purchaser."

}

]

},

"RegulatedInformation": {

"type": "object",

"description": "The regulated information collected during purchase and used to verify the order.",

"required": [

"Fields"

],

"properties": {

"Fields": {

"type": "array",

"description": "A list of regulated information fields as collected from the regulatory form.",

"items": {

"$ref": "#/definitions/RegulatedInformationField"

}

}

}

},

"RegulatedInformationField": {

"type": "object",

"required": [

"FieldId",

"FieldLabel",

"FieldType",

"FieldValue"

],

"description": "A field collected from the regulatory form.",

"properties": {

"FieldId": {

"type": "string",

"description": "The unique identifier of the field."

},

"FieldLabel": {

"type": "string",

"description": "The name of the field."

},

"FieldType": {

"type": "string",

"description": "The type of field.",

"enum": [

"Text",

"FileAttachment"

],

"x-docgen-enum-table-extension": [

{

"value": "Text",

"description": "This field is a text representation of the response collected from the regulatory form."

},

{

"value": "FileAttachment",

"description": "This field contains a link to an attachment collected from the regulatory form."

}

]

},

"FieldValue": {

"type": "string",

"description": "The content of the field as collected in regulatory form. Note that `FileAttachment` type fields contain a URL where you can download the attachment."

}

}

},

"OrderAddress": {

"type": "object",

"required": [

"AmazonOrderId"

],

"properties": {

"AmazonOrderId": {

"type": "string",

"description": "An Amazon-defined order identifier, in 3-7-7 format."

},

"BuyerCompanyName": {

"type": "string",

"description": "The company name of the contact buyer. For IBA orders, the buyer company must be Amazon entities."

},

"ShippingAddress": {

"description": "The shipping address for the order.\n\n\*\*Note\*\*: `ShippingAddress` is only available for orders with the following status values: `Unshipped`, `PartiallyShipped`, `Shipped`, and `InvoiceUnconfirmed`.",

"$ref": "#/definitions/Address"

},

"DeliveryPreferences": {

"$ref": "#/definitions/DeliveryPreferences"

}

},

"description": "The shipping address for the order."

},

"Address": {

"type": "object",

"required": [

"Name"

],

"properties": {

"Name": {

"type": "string",

"description": "The name."

},

"CompanyName": {

"type": "string",

"description": "The company name of the recipient.\n\n\*\*Note\*\*: This attribute is only available for shipping address."

},

"AddressLine1": {

"type": "string",

"description": "The street address."

},

"AddressLine2": {

"type": "string",

"description": "Additional street address information, if required."

},

"AddressLine3": {

"type": "string",

"description": "Additional street address information, if required."

},

"City": {

"type": "string",

"description": "The city."

},

"County": {

"type": "string",

"description": "The county."

},

"District": {

"type": "string",

"description": "The district."

},

"StateOrRegion": {

"type": "string",

"description": "The state or region."

},

"Municipality": {

"type": "string",

"description": "The municipality."

},

"PostalCode": {

"type": "string",

"description": "The postal code."

},

"CountryCode": {

"type": "string",

"description": "The country code. A two-character country code, in ISO 3166-1 alpha-2 format."

},

"Phone": {

"type": "string",

"description": "The phone number of the buyer.\n\n\*\*Note\*\*: \n1. This attribute is only available for shipping address.\n2. In some cases, the buyer phone number is suppressed: \na. Phone is suppressed for all `AFN` (fulfilled by Amazon) orders.\nb. Phone is suppressed for the shipped `MFN` (fulfilled by seller) order when the current date is past the Latest Delivery Date."

},

"ExtendedFields": {

"description": "The container for address extended fields. For example, street name or street number. \n\n\*\*Note\*\*: This attribute is currently only available with Brazil shipping addresses.",

"$ref": "#/definitions/AddressExtendedFields"

},

"AddressType": {

"type": "string",

"description": "The address type of the shipping address.",

"enum": [

"Residential",

"Commercial"

],

"x-docgen-enum-table-extension": [

{

"value": "Residential",

"description": "The shipping address is a residential address."

},

{

"value": "Commercial",

"description": "The shipping address is a commercial address."

}

]

}

},

"description": "The shipping address for the order."

},

"AddressExtendedFields": {

"type": "object",

"properties": {

"StreetName": {

"type": "string",

"description": "The street name."

},

"StreetNumber": {

"type": "string",

"description": "The house, building, or property number associated with the location's street address."

},

"Complement": {

"type": "string",

"description": "The floor number/unit number in the building/private house number."

},

"Neighborhood": {

"type": "string",

"description": "The neighborhood. This value is only used in some countries (such as Brazil)."

}

},

"description": "The container for address extended fields (such as `street name` and `street number`). Currently only available with Brazil shipping addresses."

},

"DeliveryPreferences": {

"type": "object",

"properties": {

"DropOffLocation": {

"type": "string",

"description": "Drop-off location selected by the customer."

},

"PreferredDeliveryTime": {

"$ref": "#/definitions/PreferredDeliveryTime",

"description": "Business hours and days when the delivery is preferred."

},

"OtherAttributes": {

"type": "array",

"items": {

"$ref": "#/definitions/OtherDeliveryAttributes"

},

"description": "Enumerated list of miscellaneous delivery attributes associated with the shipping address."

},

"AddressInstructions": {

"type": "string",

"description": "Building instructions, nearby landmark or navigation instructions."

}

},

"description": "Contains all of the delivery instructions provided by the customer for the shipping address."

},

"PreferredDeliveryTime": {

"type": "object",

"description": "The time window when the delivery is preferred.",

"properties": {

"BusinessHours": {

"type": "array",

"items": {

"$ref": "#/definitions/BusinessHours"

},

"description": "Business hours when the business is open for deliveries."

},

"ExceptionDates": {

"type": "array",

"items": {

"$ref": "#/definitions/ExceptionDates"

},

"description": "Dates when the business is closed during the next 30 days."

}

}

},

"BusinessHours": {

"type": "object",

"description": "Business days and hours when the destination is open for deliveries.",

"properties": {

"DayOfWeek": {

"type": "string",

"description": "Day of the week.",

"enum": [

"SUN",

"MON",

"TUE",

"WED",

"THU",

"FRI",

"SAT"

],

"x-docgen-enum-table-extension": [

{

"value": "SUN",

"description": "Sunday - Day of the week."

},

{

"value": "MON",

"description": "Monday - Day of the week."

},

{

"value": "TUE",

"description": "Tuesday - Day of the week."

},

{

"value": "WED",

"description": "Wednesday - Day of the week."

},

{

"value": "THU",

"description": "Thursday - Day of the week."

},

{

"value": "FRI",

"description": "Friday - Day of the week."

},

{

"value": "SAT",

"description": "Saturday - Day of the week."

}

]

},

"OpenIntervals": {

"type": "array",

"description": "Time window during the day when the business is open.",

"items": {

"$ref": "#/definitions/OpenInterval"

}

}

}

},

"ExceptionDates": {

"type": "object",

"description": "Dates when the business is closed or open with a different time window.",

"properties": {

"ExceptionDate": {

"type": "string",

"description": "Date when the business is closed, in <a href='https://developer-docs.amazon.com/sp-api/docs/iso-8601'>ISO 8601</a> date format."

},

"IsOpen": {

"type": "boolean",

"description": "Boolean indicating if the business is closed or open on that date."

},

"OpenIntervals": {

"type": "array",

"description": "Time window during the day when the business is open.",

"items": {

"$ref": "#/definitions/OpenInterval"

}

}

}

},

"OpenInterval": {

"type": "object",

"description": "The time interval for which the business is open.",

"properties": {

"StartTime": {

"$ref": "#/definitions/OpenTimeInterval",

"description": "The time when the business opens."

},

"EndTime": {

"$ref": "#/definitions/OpenTimeInterval",

"description": "The time when the business closes."

}

}

},

"OpenTimeInterval": {

"type": "object",

"description": "The time when the business opens or closes.",

"properties": {

"Hour": {

"type": "integer",

"description": "The hour when the business opens or closes."

},

"Minute": {

"type": "integer",

"description": "The minute when the business opens or closes."

}

}

},

"OtherDeliveryAttributes": {

"type": "string",

"description": "Miscellaneous delivery attributes associated with the shipping address.",

"enum": [

"HAS\_ACCESS\_POINT",

"PALLET\_ENABLED",

"PALLET\_DISABLED"

],

"x-docgen-enum-table-extension": [

{

"value": "HAS\_ACCESS\_POINT",

"description": "Indicates whether the delivery has an access point pickup or drop-off location."

},

{

"value": "PALLET\_ENABLED",

"description": "Indicates whether pallet delivery is enabled for the address."

},

{

"value": "PALLET\_DISABLED",

"description": "Indicates whether pallet delivery is disabled for the address."

}

]

},

"Money": {

"type": "object",

"properties": {

"CurrencyCode": {

"type": "string",

"description": "The three-digit currency code. In ISO 4217 format."

},

"Amount": {

"type": "string",

"description": "The currency amount."

}

},

"description": "The monetary value of the order."

},

"PaymentMethodDetailItemList": {

"type": "array",

"description": "A list of payment method detail items.",

"items": {

"type": "string"

}

},

"PaymentExecutionDetailItemList": {

"type": "array",

"description": "A list of payment execution detail items.",

"items": {

"$ref": "#/definitions/PaymentExecutionDetailItem"

}

},

"PaymentExecutionDetailItem": {

"type": "object",

"required": [

"Payment",

"PaymentMethod"

],

"properties": {

"Payment": {

"$ref": "#/definitions/Money"

},

"PaymentMethod": {

"type": "string",

"description": "A sub-payment method for a COD order.\n\n\*\*Possible values\*\*:\n\* `COD`: Cash on delivery \n\* `GC`: Gift card \n\* `PointsAccount`: Amazon Points\n\* `Invoice`: Invoice"

}

},

"description": "Information about a sub-payment method used to pay for a COD order."

},

"BuyerTaxInfo": {

"type": "object",

"properties": {

"CompanyLegalName": {

"type": "string",

"description": "The legal name of the company."

},

"TaxingRegion": {

"type": "string",

"description": "The country or region imposing the tax."

},

"TaxClassifications": {

"type": "array",

"description": "A list of tax classifications that apply to the order.",

"items": {

"$ref": "#/definitions/TaxClassification"

}

}

},

"description": "Tax information about the buyer."

},

"MarketplaceTaxInfo": {

"type": "object",

"properties": {

"TaxClassifications": {

"type": "array",

"description": "A list of tax classifications that apply to the order.",

"items": {

"$ref": "#/definitions/TaxClassification"

}

}

},

"description": "Tax information about the marketplace."

},

"TaxClassification": {

"type": "object",

"properties": {

"Name": {

"type": "string",

"description": "The type of tax."

},

"Value": {

"type": "string",

"description": "The buyer's tax identifier."

}

},

"description": "The tax classification of the order."

},

"OrderItemsList": {

"type": "object",

"required": [

"AmazonOrderId",

"OrderItems"

],

"properties": {

"OrderItems": {

"$ref": "#/definitions/OrderItemList"

},

"NextToken": {

"type": "string",

"description": "When present and not empty, pass this string token in the next request to return the next response page."

},

"AmazonOrderId": {

"type": "string",

"description": "An Amazon-defined order identifier, in 3-7-7 format."

}

},

"description": "The order items list along with the order ID."

},

"OrderItemList": {

"type": "array",

"description": "A list of order items.",

"items": {

"$ref": "#/definitions/OrderItem"

}

},

"OrderItem": {

"type": "object",

"required": [

"ASIN",

"OrderItemId",

"QuantityOrdered"

],

"properties": {

"ASIN": {

"type": "string",

"description": "The item's Amazon Standard Identification Number (ASIN)."

},

"SellerSKU": {

"type": "string",

"description": "The item's seller stock keeping unit (SKU)."

},

"OrderItemId": {

"type": "string",

"description": "An Amazon-defined order item identifier."

},

"AssociatedItems": {

"type": "array",

"description": "A list of associated items that a customer has purchased with a product. For example, a tire installation service purchased with tires.",

"items": {

"$ref": "#/definitions/AssociatedItem"

}

},

"Title": {

"type": "string",

"description": "The item's name."

},

"QuantityOrdered": {

"type": "integer",

"description": "The number of items in the order. "

},

"QuantityShipped": {

"type": "integer",

"description": "The number of items shipped."

},

"ProductInfo": {

"description": "The item's product information.",

"$ref": "#/definitions/ProductInfoDetail"

},

"PointsGranted": {

"description": "The number and value of Amazon Points granted with the purchase of an item.",

"$ref": "#/definitions/PointsGrantedDetail"

},

"ItemPrice": {

"description": "The selling price of the order item. Note that an order item is an item and a quantity. This means that the value of `ItemPrice` is equal to the selling price of the item multiplied by the quantity ordered. `ItemPrice` excludes `ShippingPrice` and GiftWrapPrice.",

"$ref": "#/definitions/Money"

},

"ShippingPrice": {

"description": "The item's shipping price.",

"$ref": "#/definitions/Money"

},

"ItemTax": {

"description": "The tax on the item price.",

"$ref": "#/definitions/Money"

},

"ShippingTax": {

"description": "The tax on the shipping price.",

"$ref": "#/definitions/Money"

},

"ShippingDiscount": {

"description": "The discount on the shipping price.",

"$ref": "#/definitions/Money"

},

"ShippingDiscountTax": {

"description": "The tax on the discount on the shipping price.",

"$ref": "#/definitions/Money"

},

"PromotionDiscount": {

"description": "The total of all promotional discounts in the offer.",

"$ref": "#/definitions/Money"

},

"PromotionDiscountTax": {

"description": "The tax on the total of all promotional discounts in the offer.",

"$ref": "#/definitions/Money"

},

"PromotionIds": {

"$ref": "#/definitions/PromotionIdList"

},

"CODFee": {

"description": "The fee charged for COD service.",

"$ref": "#/definitions/Money"

},

"CODFeeDiscount": {

"description": "The discount on the COD fee.",

"$ref": "#/definitions/Money"

},

"IsGift": {

"type": "string",

"description": "Indicates whether the item is a gift.\n\n\*\*Possible values\*\*: `true` and `false`."

},

"ConditionNote": {

"type": "string",

"description": "The condition of the item, as described by the seller."

},

"ConditionId": {

"type": "string",

"description": "The condition of the item.\n\n\*\*Possible values\*\*: `New`, `Used`, `Collectible`, `Refurbished`, `Preorder`, and `Club`."

},

"ConditionSubtypeId": {

"type": "string",

"description": "The subcondition of the item.\n\n\*\*Possible values\*\*: `New`, `Mint`, `Very Good`, `Good`, `Acceptable`, `Poor`, `Club`, `OEM`, `Warranty`, `Refurbished Warranty`, `Refurbished`, `Open Box`, `Any`, and `Other`."

},

"ScheduledDeliveryStartDate": {

"type": "string",

"description": "The start date of the scheduled delivery window in the time zone for the order destination. In [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) date time format."

},

"ScheduledDeliveryEndDate": {

"type": "string",

"description": "The end date of the scheduled delivery window in the time zone for the order destination. In [ISO 8601](https://developer-docs.amazon.com/sp-api/docs/iso-8601) date time format."

},

"PriceDesignation": {

"type": "string",

"description": "Indicates that the selling price is a special price that is only available for Amazon Business orders. For more information about the Amazon Business Seller Program, refer to the [Amazon Business website](https://www.amazon.com/b2b/info/amazon-business). \n\n\*\*Possible values\*\*: `BusinessPrice`"

},

"TaxCollection": {

"description": "Information about withheld taxes.",

"$ref": "#/definitions/TaxCollection"

},

"SerialNumberRequired": {

"type": "boolean",

"description": "When true, the product type for this item has a serial number.\n\n Only returned for Amazon Easy Ship orders."

},

"IsTransparency": {

"type": "boolean",

"description": "When true, the ASIN is enrolled in Transparency. The Transparency serial number that you must submit is determined by:\n\n\*\*1D or 2D Barcode:\*\* This has a \*\*T\*\* logo. Submit either the 29-character alpha-numeric identifier beginning with \*\*AZ\*\* or \*\*ZA\*\*, or the 38-character Serialized Global Trade Item Number (SGTIN).\n\*\*2D Barcode SN:\*\* Submit the 7- to 20-character serial number barcode, which likely has the prefix \*\*SN\*\*. The serial number is applied to the same side of the packaging as the GTIN (UPC/EAN/ISBN) barcode.\n\*\*QR code SN:\*\* Submit the URL that the QR code generates."

},

"IossNumber": {

"type": "string",

"description": "The IOSS number of the marketplace. Sellers shipping to the EU from outside the EU must provide this IOSS number to their carrier when Amazon has collected the VAT on the sale."

},

"StoreChainStoreId": {

"type": "string",

"description": "The store chain store identifier. Linked to a specific store in a store chain."

},

"DeemedResellerCategory": {

"type": "string",

"description": "The category of deemed reseller. This applies to selling partners that are not based in the EU and is used to help them meet the VAT Deemed Reseller tax laws in the EU and UK.",

"enum": [

"IOSS",

"UOSS"

],

"x-docgen-enum-table-extension": [

{

"value": "IOSS",

"description": "Import one stop shop. The item being purchased is not held in the EU for shipment."

},

{

"value": "UOSS",

"description": "Union one stop shop. The item being purchased is held in the EU for shipment."

}

]

},

"BuyerInfo": {

"description": "A single item's buyer information.\n\n\*\*Note\*\*: The `BuyerInfo` contains restricted data. Use the Restricted Data Token (RDT) and restricted SPDS roles to access the restricted data in `BuyerInfo`. For example, `BuyerCustomizedInfo` and `GiftMessageText`.",

"$ref": "#/definitions/ItemBuyerInfo"

},

"BuyerRequestedCancel": {

"description": "Information about whether or not a buyer requested cancellation.",

"$ref": "#/definitions/BuyerRequestedCancel"

},

"SerialNumbers": {

"type": "array",

"description": "A list of serial numbers for electronic products that are shipped to customers. Returned for FBA orders only.",

"items": {

"type": "string"

}

},

"SubstitutionPreferences": {

"description": "Substitution preferences for the order item. This is an optional field that is only present if a seller supports substitutions, as is the case with some grocery sellers.",

"$ref": "#/definitions/SubstitutionPreferences"

},

"Measurement": {

"description": "Measurement information for the order item.",

"$ref": "#/definitions/Measurement"

},

"ShippingConstraints" : {

"description" : "Shipping constraints applicable to this order.",

"$ref" : "#/definitions/ShippingConstraints"

},

"AmazonPrograms": {

"description": "Contains the list of programs that are associated with an item.",

"$ref" : "#/definitions/AmazonPrograms"

}

},

"description": "A single order item."

},

"AmazonPrograms": {

"type": "object",

"description": "Contains the list of programs that Amazon associates with an item.\n\nPossible programs are:\n - \*\*Subscribe and Save\*\*: Offers recurring, scheduled deliveries to Amazon customers and Amazon Business customers for their frequently ordered products. - \*\*FBM Ship+\*\*: Unlocks expedited shipping without the extra cost. Helps you to provide accurate and fast delivery dates to Amazon customers. You also receive protection from late deliveries, a discount on expedited shipping rates, and cash back when you ship.",

"required": [

"Programs"

],

"properties": {

"Programs": {

"type": "array",

"description": "A list of the programs that Amazon associates with the order item.\n\n\*\*Possible values\*\*: `SUBSCRIBE\_AND\_SAVE`, `FBM\_SHIP\_PLUS`",

"items": {

"type": "string"

}

}

}

},

"SubstitutionPreferences": {

"type": "object",

"description": "Substitution preferences for an order item.",

"required": [

"SubstitutionType"

],

"properties": {

"SubstitutionType": {

"type": "string",

"description": "The type of substitution that these preferences represent.",

"enum": [

"CUSTOMER\_PREFERENCE",

"AMAZON\_RECOMMENDED",

"DO\_NOT\_SUBSTITUTE"

],

"x-docgen-enum-table-extension": [

{

"value": "CUSTOMER\_PREFERENCE",

"description": "Customer has provided the substitution preferences."

},

{

"value": "AMAZON\_RECOMMENDED",

"description": "Amazon has provided the substitution preferences."

},

{

"value": "DO\_NOT\_SUBSTITUTE",

"description": "Do not provide a substitute if item is not found."

}

]

},

"SubstitutionOptions": {

"description": "Substitution options for the order item.",

"$ref": "#/definitions/SubstitutionOptionList"

}

}

},

"SubstitutionOptionList": {

"type": "array",

"description": "A collection of substitution options.",

"items": {

"$ref": "#/definitions/SubstitutionOption"

}

},

"SubstitutionOption": {

"type": "object",

"description": "Substitution options for an order item.",

"properties": {

"ASIN": {

"type": "string",

"description": "The item's Amazon Standard Identification Number (ASIN)."

},

"QuantityOrdered": {

"type": "integer",

"description": "The number of items to be picked for this substitution option. "

},

"SellerSKU": {

"type": "string",

"description": "The item's seller stock keeping unit (SKU)."

},

"Title": {

"type": "string",

"description": "The item's title."

},

"Measurement": {

"description": "Measurement information for the substitution option.",

"$ref": "#/definitions/Measurement"

}

}

},

"Measurement": {

"type": "object",

"description": "Measurement information for an order item.",

"required": [

"Unit",

"Value"

],

"properties": {

"Unit": {

"type": "string",

"description": "The unit of measure.",

"enum": [

"OUNCES",

"POUNDS",

"KILOGRAMS",

"GRAMS",

"MILLIGRAMS",

"INCHES",

"FEET",

"METERS",

"CENTIMETERS",

"MILLIMETERS",

"SQUARE\_METERS",

"SQUARE\_CENTIMETERS",

"SQUARE\_FEET",

"SQUARE\_INCHES",

"GALLONS",

"PINTS",

"QUARTS",

"FLUID\_OUNCES",

"LITERS",

"CUBIC\_METERS",

"CUBIC\_FEET",

"CUBIC\_INCHES",

"CUBIC\_CENTIMETERS",

"COUNT"

],

"x-docgen-enum-table-extension": [

{

"value": "OUNCES",

"description": "The item is measured in ounces."

},

{

"value": "POUNDS",

"description": "The item is measured in pounds."

},

{

"value": "KILOGRAMS",

"description": "The item is measured in kilograms."

},

{

"value": "GRAMS",

"description": "The item is measured in grams."

},

{

"value": "MILLIGRAMS",

"description": "The item is measured in milligrams."

},

{

"value": "INCHES",

"description": "The item is measured in inches."

},

{

"value": "FEET",

"description": "The item is measured in feet."

},

{

"value": "METERS",

"description": "The item is measured in meters."

},

{

"value": "CENTIMETERS",

"description": "The item is measured in centimeters."

},

{

"value": "MILLIMETERS",

"description": "The item is measured in millimeters."

},

{

"value": "SQUARE\_METERS",

"description": "The item is measured in square meters."

},

{

"value": "SQUARE\_CENTIMETERS",

"description": "The item is measured in square centimeters."

},

{

"value": "SQUARE\_FEET",

"description": "The item is measured in square feet."

},

{

"value": "SQUARE\_INCHES",

"description": "The item is measured in square inches."

},

{

"value": "GALLONS",

"description": "The item is measured in gallons."

},

{

"value": "PINTS",

"description": "The item is measured in pints."

},

{

"value": "QUARTS",

"description": "The item is measured in quarts."

},

{

"value": "FLUID\_OUNCES",

"description": "The item is measured in fluid ounces."

},

{

"value": "LITERS",

"description": "The item is measured in liters."

},

{

"value": "CUBIC\_METERS",

"description": "The item is measured in cubic meters."

},

{

"value": "CUBIC\_FEET",

"description": "The item is measured in cubic feet."

},

{

"value": "CUBIC\_INCHES",

"description": "The item is measured in cubic inches."

},

{

"value": "CUBIC\_CENTIMETERS",

"description": "The item is measured in cubic centimeters."

},

{

"value": "COUNT",

"description": "The item is measured by count."

}

]

},

"Value": {

"type": "number",

"description": "The measurement value."

}

}

},

"AssociatedItem": {

"description": "An item that is associated with an order item. For example, a tire installation service that is purchased with tires.",

"type": "object",

"properties": {

"OrderId": {

"type": "string",

"description": "The order item's order identifier, in 3-7-7 format."

},

"OrderItemId": {

"type": "string",

"description": "An Amazon-defined item identifier for the associated item."

},

"AssociationType": {

"$ref": "#/definitions/AssociationType"

}

}

},

"AssociationType": {

"type": "string",

"description": "The type of association an item has with an order item.",

"enum": [

"VALUE\_ADD\_SERVICE"

],

"x-docgen-enum-table-extension": [

{

"value": "VALUE\_ADD\_SERVICE",

"description": "The associated item is a service order."

}

]

},

"OrderItemsBuyerInfoList": {

"type": "object",

"required": [

"AmazonOrderId",

"OrderItems"

],

"properties": {

"OrderItems": {

"$ref": "#/definitions/OrderItemBuyerInfoList"

},

"NextToken": {

"type": "string",

"description": "When present and not empty, pass this string token in the next request to return the next response page."

},

"AmazonOrderId": {

"type": "string",

"description": "An Amazon-defined order identifier, in 3-7-7 format."

}

},

"description": "A single order item's buyer information list with the order ID."

},

"OrderItemBuyerInfoList": {

"type": "array",

"description": "A single order item's buyer information list.",

"items": {

"$ref": "#/definitions/OrderItemBuyerInfo"

}

},

"OrderItemBuyerInfo": {

"type": "object",

"required": [

"OrderItemId"

],

"properties": {

"OrderItemId": {

"type": "string",

"description": "An Amazon-defined order item identifier."

},

"BuyerCustomizedInfo": {

"description": "Buyer information for custom orders from the Amazon Custom program.\n\n\*\*Note\*\*: This attribute is only available for MFN (fulfilled by seller) orders.",

"$ref": "#/definitions/BuyerCustomizedInfoDetail"

},

"GiftWrapPrice": {

"description": "The gift wrap price of the item.",

"$ref": "#/definitions/Money"

},

"GiftWrapTax": {

"description": "The tax on the gift wrap price.",

"$ref": "#/definitions/Money"

},

"GiftMessageText": {

"type": "string",

"description": "A gift message provided by the buyer.\n\n\*\*Note\*\*: This attribute is only available for MFN (fulfilled by seller) orders."

},

"GiftWrapLevel": {

"type": "string",

"description": "The gift wrap level specified by the buyer."

}

},

"description": "A single order item's buyer information."

},

"PointsGrantedDetail": {

"type": "object",

"properties": {

"PointsNumber": {

"type": "integer",

"description": "The number of Amazon Points granted with the purchase of an item."

},

"PointsMonetaryValue": {

"description": "The monetary value of the Amazon Points granted.",

"$ref": "#/definitions/Money"

}

},

"description": "The number of Amazon Points offered with the purchase of an item, and their monetary value."

},

"ProductInfoDetail": {

"type": "object",

"properties": {

"NumberOfItems": {

"type": "string",

"description": "The total number of items that are included in the ASIN."

}

},

"description": "Product information on the number of items."

},

"PromotionIdList": {

"type": "array",

"description": "A list of promotion identifiers provided by the seller when the promotions were created.",

"items": {

"type": "string"

}

},

"BuyerCustomizedInfoDetail": {

"type": "object",

"properties": {

"CustomizedURL": {

"type": "string",

"description": "The location of a ZIP file containing Amazon Custom data."

}

},

"description": "Buyer information for custom orders from the Amazon Custom program."

},

"TaxCollection": {

"type": "object",

"properties": {

"Model": {

"type": "string",

"description": "The tax collection model applied to the item.",

"enum": [

"MarketplaceFacilitator"

],

"x-docgen-enum-table-extension": [

{

"value": "MarketplaceFacilitator",

"description": "Tax is withheld and remitted to the taxing authority by Amazon on behalf of the seller."

}

]

},

"ResponsibleParty": {

"type": "string",

"description": "The party responsible for withholding the taxes and remitting them to the taxing authority.",

"enum": [

"Amazon Services, Inc."

],

"x-docgen-enum-table-extension": [

{

"value": "Amazon Services, Inc.",

"description": "The `MarketplaceFacilitator` entity for the US marketplace."

}

]

}

},

"description": "Information about withheld taxes."

},

"BuyerTaxInformation": {

"type": "object",

"properties": {

"BuyerLegalCompanyName": {

"type": "string",

"description": "Business buyer's company legal name."

},

"BuyerBusinessAddress": {

"type": "string",

"description": "Business buyer's address."

},

"BuyerTaxRegistrationId": {

"type": "string",

"description": "Business buyer's tax registration ID."

},

"BuyerTaxOffice": {

"type": "string",

"description": "Business buyer's tax office."

}

},

"description": "Contains the business invoice tax information. Available only in the TR marketplace."

},

"FulfillmentInstruction": {

"type": "object",

"properties": {

"FulfillmentSupplySourceId": {

"description": "The `sourceId` of the location from where you want the order fulfilled.",

"type": "string"

}

},

"description": "Contains the instructions about the fulfillment, such as the location from where you want the order filled."

},

"ShippingConstraints": {

"type": "object",

"description": "Delivery constraints applicable to this order.",

"properties": {

"PalletDelivery": {

"description": "Indicates if the line item needs to be delivered by pallet.",

"$ref": "#/definitions/ConstraintType"

},

"SignatureConfirmation": {

"description": "Indicates that the recipient of the line item must sign to confirm its delivery.",

"$ref": "#/definitions/ConstraintType"

},

"RecipientIdentityVerification": {

"description": "Indicates that the person receiving the line item must be the same as the intended recipient of the order.",

"$ref": "#/definitions/ConstraintType"

},

"RecipientAgeVerification": {

"description": "Indicates that the carrier must confirm the recipient is of the legal age to receive the line item upon delivery.",

"$ref": "#/definitions/ConstraintType"

}

}

},

"ConstraintType": {

"type": "string",

"description": "Details the importance of the constraint present on the item",

"enum": [

"MANDATORY"

],

"x-docgen-enum-table-extension": [

{

"value": "MANDATORY",

"description": "Item must follow the constraint to ensure order is met with buyer's delivery requirements."

}

]

},

"BuyerInfo": {

"type": "object",

"properties": {

"BuyerEmail": {

"type": "string",

"description": "The anonymized email address of the buyer."

},

"BuyerName": {

"type": "string",

"description": "The buyer name or the recipient name."

},

"BuyerCounty": {

"type": "string",

"description": "The county of the buyer.\n\n\*\*Note\*\*: This attribute is only available in the Brazil marketplace."

},

"BuyerTaxInfo": {

"description": "Tax information about the buyer. Sellers could use this data to issue electronic invoices for business orders.\n\n\*\*Note\*\*: This attribute is only available for business orders in the Brazil, Mexico and India marketplaces.",

"$ref": "#/definitions/BuyerTaxInfo"

},

"PurchaseOrderNumber": {

"type": "string",

"description": "The purchase order (PO) number entered by the buyer at checkout. Only returned for orders where the buyer entered a PO number at checkout."

}

},

"description": "Buyer information."

},

"ItemBuyerInfo": {

"type": "object",

"properties": {

"BuyerCustomizedInfo": {

"description": "Buyer information for custom orders from the Amazon Custom program.\n\n\*\*Note\*\*: This attribute is only available for MFN (fulfilled by seller) orders.",

"$ref": "#/definitions/BuyerCustomizedInfoDetail"

},

"GiftWrapPrice": {

"description": "The gift wrap price of the item.",

"$ref": "#/definitions/Money"

},

"GiftWrapTax": {

"description": "The tax on the gift wrap price.",

"$ref": "#/definitions/Money"

},

"GiftMessageText": {

"type": "string",

"description": "A gift message provided by the buyer.\n\n\*\*Note\*\*: This attribute is only available for MFN (fulfilled by seller) orders."

},

"GiftWrapLevel": {

"type": "string",

"description": "The gift wrap level specified by the buyer."

}

},

"description": "A single item's buyer information."

},

"AutomatedShippingSettings": {

"description": "Contains information regarding the Shipping Settings Automation program, such as whether the order's shipping settings were generated automatically, and what those settings are.",

"type": "object",

"properties": {

"HasAutomatedShippingSettings": {

"description": "When true, this order has automated shipping settings generated by Amazon. This order could be identified as an SSA order.",

"type": "boolean"

},

"AutomatedCarrier": {

"description": "Auto-generated carrier for SSA orders.",

"type": "string"

},

"AutomatedShipMethod": {

"description": "Auto-generated ship method for SSA orders.",

"type": "string"

}

}

},

"BuyerRequestedCancel": {

"type": "object",

"properties": {

"IsBuyerRequestedCancel": {

"type": "string",

"description": "Indicate whether the buyer has requested cancellation.\n\n\*\*Possible Values\*\*: `true`, `false`."

},

"BuyerCancelReason": {

"type": "string",

"description": "The reason that the buyer requested cancellation."

}

},

"description": "Information about whether or not a buyer requested cancellation."

},

"EasyShipShipmentStatus": {

"description": "The status of the Amazon Easy Ship order. This property is only included for Amazon Easy Ship orders.",

"type": "string",

"enum": [

"PendingSchedule",

"PendingPickUp",

"PendingDropOff",

"LabelCanceled",

"PickedUp",

"DroppedOff",

"AtOriginFC",

"AtDestinationFC",

"Delivered",

"RejectedByBuyer",

"Undeliverable",

"ReturningToSeller",

"ReturnedToSeller",

"Lost",

"OutForDelivery",

"Damaged"

],

"x-docgen-enum-table-extension": [

{

"value": "PendingSchedule",

"description": "The package is awaiting the schedule for pick-up."

},

{

"value": "PendingPickUp",

"description": "Amazon has not yet picked up the package from the seller."

},

{

"value": "PendingDropOff",

"description": "The seller will deliver the package to the carrier."

},

{

"value": "LabelCanceled",

"description": "The seller canceled the pickup."

},

{

"value": "PickedUp",

"description": "Amazon has picked up the package from the seller."

},

{

"value": "DroppedOff",

"description": "The package was delivered to the carrier by the seller."

},

{

"value": "AtOriginFC",

"description": "The package is at the origin fulfillment center."

},

{

"value": "AtDestinationFC",

"description": "The package is at the destination fulfillment center."

},

{

"value": "Delivered",

"description": "The package has been delivered."

},

{

"value": "RejectedByBuyer",

"description": "The package has been rejected by the buyer."

},

{

"value": "Undeliverable",

"description": "The package cannot be delivered."

},

{

"value": "ReturningToSeller",

"description": "The package was not delivered and is being returned to the seller."

},

{

"value": "ReturnedToSeller",

"description": "The package was not delivered and was returned to the seller."

},

{

"value": "Lost",

"description": "The package is lost."

},

{

"value": "OutForDelivery",

"description": "The package is out for delivery."

},

{

"value": "Damaged",

"description": "The package was damaged by the carrier."

}

]

},

"ElectronicInvoiceStatus": {

"description": "The status of the electronic invoice. Only available for Easy Ship orders and orders in the BR marketplace.",

"type": "string",

"enum": [

"NotRequired",

"NotFound",

"Processing",

"Errored",

"Accepted"

],

"x-docgen-enum-table-extension": [

{

"value": "NotRequired",

"description": "The order does not require an electronic invoice to be uploaded."

},

{

"value": "NotFound",

"description": "The order requires an electronic invoice but it is not uploaded."

},

{

"value": "Processing",

"description": "The required electronic invoice was uploaded and is processing."

},

{

"value": "Errored",

"description": "The uploaded electronic invoice was not accepted."

},

{

"value": "Accepted",

"description": "The uploaded electronic invoice was accepted."

}

]

},

"ConfirmShipmentRequest": {

"type": "object",

"required": [

"marketplaceId",

"packageDetail"

],

"properties": {

"packageDetail": {

"$ref": "#/definitions/PackageDetail"

},

"codCollectionMethod": {

"type": "string",

"description": "The COD collection method (only supported in the JP marketplace).",

"enum":["DirectPayment"]

},

"marketplaceId": {

"$ref": "#/definitions/MarketplaceId"

}

},

"description": "The request schema for an shipment confirmation."

},

"ConfirmShipmentErrorResponse": {

"type": "object",

"description": "The error response schema for the `confirmShipment` operation.",

"properties": {

"errors": {

"description": "One or more unexpected errors occurred during the `confirmShipment` operation.",

"$ref": "#/definitions/ErrorList"

}

}

},

"PackageDetail": {

"type": "object",

"description": "Properties of packages",

"required": [

"packageReferenceId",

"carrierCode",

"trackingNumber",

"shipDate",

"orderItems"

],

"properties": {

"packageReferenceId": {

"$ref": "#/definitions/PackageReferenceId"

},

"carrierCode": {

"type": "string",

"description": "Identifies the carrier that will deliver the package. This field is required for all marketplaces. For more information, refer to the [`CarrierCode` announcement](https://developer-docs.amazon.com/sp-api/changelog/carriercode-value-required-in-shipment-confirmations-for-br-mx-ca-sg-au-in-jp-marketplaces)."

},

"carrierName": {

"type": "string",

"description": "Carrier name that will deliver the package. Required when `carrierCode` is \"Other\" "

},

"shippingMethod": {

"type": "string",

"description": "Ship method to be used for shipping the order."

},

"trackingNumber": {

"type": "string",

"description": "The tracking number used to obtain tracking and delivery information."

},

"shipDate": {

"type": "string",

"description": "The shipping date for the package. Must be in <a href='https://developer-docs.amazon.com/sp-api/docs/iso-8601'>ISO 8601</a> date/time format.",

"format": "date-time"

},

"shipFromSupplySourceId": {

"type": "string",

"description": "The unique identifier for the supply source."

},

"orderItems": {

"description": "The list of order items and quantities to be updated.",

"$ref": "#/definitions/ConfirmShipmentOrderItemsList"

}

}

},

"ConfirmShipmentOrderItemsList": {

"type": "array",

"description": "A list of order items.",

"items": {

"$ref": "#/definitions/ConfirmShipmentOrderItem"

}

},

"ConfirmShipmentOrderItem": {

"type": "object",

"description": "A single order item.",

"required": [

"orderItemId",

"quantity"

],

"properties": {

"orderItemId": {

"type": "string",

"description": "The order item's unique identifier."

},

"quantity": {

"type": "integer",

"description": "The item's quantity."

},

"transparencyCodes": {

"description": "The list of transparency codes.",

"$ref": "#/definitions/TransparencyCodeList"

}

}

},

"TransparencyCodeList": {

"type": "array",

"description": "A list of order items.",

"items": {

"$ref": "#/definitions/TransparencyCode"

}

},

"TransparencyCode": {

"type": "string",

"description": "The transparency code associated with the item."

},

"PackageReferenceId": {

"type": "string",

"description": "A seller-supplied identifier that uniquely identifies a package within the scope of an order. Only positive numeric values are supported."

},

"ErrorList": {

"type": "array",

"description": "A list of error responses returned when a request is unsuccessful.",

"items": {

"$ref": "#/definitions/Error"

}

},

"Error": {

"type": "object",

"required": [

"code",

"message"

],

"properties": {

"code": {

"type": "string",

"description": "An error code that identifies the type of error that occurred."

},

"message": {

"type": "string",

"description": "A message that describes the error condition."

},

"details": {

"type": "string",

"description": "Additional details that can help the caller understand or fix the issue."

}

},

"description": "Error response returned when the request is unsuccessful."

}

}

}