## **Calculate Tax Request**

**Upon request, the calculate tax integration will retrieve the order details from OrderCloud.io and check which products in the order have applyTax set to true from the assigned price schedule for that user. Once it has built up the lines/products for the order that need tax calculated it will then send the object to Avalara to be calculated with the clients APIkey and customerCode that was saved in the integration proxy.**

**(For now, this information can be saved upon request to the OrderCloud.io integration team).**

**POST https://api.ordercloud.io/v1/integrationproxy/avalara HTTP/1.1  
Authorization: bearer insert\_access\_token\_here  
Content-Type: application/json; charset=UTF-8  
  
{  
 "taxRequestType": "...",  
 "buyerID": "...",  
 "orderID": "...",  
 "returnDetails": [  
 {  
 "lineNumber": "1001",  
 "quantity": "8"  
 },  
 {  
 "lineNumber": "1002",  
 "quantity": "1"  
 }  
 ]  
}**

|  |  |
| --- | --- |
| **Tax Request Type** | **Description** |
| **estimateTax** | **Returns tax cost saving a temporary tax document to the tax history in Avalara.** |
| **collectTax** | **Returns tax cost and saves a tax document to the tax history in Avalara with docType Sales Invoice.** |
| **estimatePO** | **Returns tax cost and saves a tax document to the tax history in Avalara with docType Purchase Invoice.** |
| **collectPO** | **Returns tax cost saving a temporary tax document to the tax history in Avalara** |
| **estimateRefund** | **Returns tax cost saving a temporary tax document to the tax history in Avalara** |
| **collectRefund** | **Returns tax cost and saves a tax document to the tax history in Avalara with docType Return Invoice** |

## **Calculate Tax Response**

**After calculating, OrderCloud.io will return a JSON object with the tax costs of the whole order and also broken down by line and any special taxes (state, city, county, transit, etc). The integration also updates the TaxCost on the order object with the total cost from Avalara.**

**On a return type calculation, the integration also takes in an array of lineNumbers and quantity to calculate the correct amount to refund. Upon submission it will locate the line item and check to see if that product has applyTax set to true on the assigned priceSchedule for that user. On the call to Avalara it will use the original order submission date to calculate the correct tax amount to refund. It will then return a JSON object with the tax costs of the whole order and also broken down by line and any special taxes (state, city, county, transit, etc). It will then update the order taxCost with the new tax total after subtracting the refunded tax amount.**

**HTTP/1.1 201 Created  
Content-Type: application/json; charset=UTF-8  
  
{  
 "DocCode": "8888",  
 "DocDate": "2016-07-15",  
 "TimeStamp": null,  
 "TotalAmount": 192.5,  
 "TotalDiscount": 0,  
 "TotalExemption": 0,  
 "TotalTaxable": 192.5,  
 "TotalTax": 14.97,  
 "TotalTaxCalculated": 14.97,  
 "TaxDate": "2016-07-15",  
 "TaxLines": [  
 {  
 "LineNo": "1001",  
 "TaxCode": "P0000000",  
 "Taxability": true,  
 "Taxable": 125,  
 "Rate": 0.07775,  
 "Tax": 9.72,  
 "Discount": 0,  
 "TaxCalculated": 9.72,  
 "Exemption": 0,  
 "BoundaryLevel": "Address",  
 "TaxDetails": [  
 {  
 "Country": "US",  
 "JurisName": "MINNESOTA",  
 "JurisCode": 27,  
 "JurisType": "State",  
 "Rate": 0.06875,  
 "Region": "MN",  
 "Tax": 8.59,  
 "TaxName": "MN STATE TAX"  
 },  
 {  
 "Country": "US",  
 "JurisName": "HENNEPIN",  
 "JurisCode": 53,  
 "JurisType": "County",  
 "Rate": 0.0015,  
 "Region": "MN",  
 "Tax": 0.19,  
 "TaxName": "MN COUNTY TAX"  
 },  
 {  
 "Country": "US",  
 "JurisName": "MINNEAPOLIS",  
 "JurisCode": 43000,  
 "JurisType": "City",  
 "Rate": 0.005,  
 "Region": "MN",  
 "Tax": 0.63,  
 "TaxName": "MN CITY TAX"  
 },  
 {  
 "Country": "US",  
 "JurisName": "TRANSIT IMPROVEMENT TAX",  
 "JurisCode": 80002,  
 "JurisType": "Special",  
 "Rate": 0.0025,  
 "Region": "MN",  
 "Tax": 0.31,  
 "TaxName": "MN SPECIAL TAX"  
 }  
 ]  
 },  
 {  
 "LineNo": "1002",  
 "TaxCode": "P0000000",  
 "Taxability": true,  
 "Taxable": 67.5,  
 "Rate": 0.07775,  
 "Tax": 5.25,  
 "Discount": 0,  
 "TaxCalculated": 5.25,  
 "Exemption": 0,  
 "BoundaryLevel": "Address",  
 "TaxDetails": [  
 {  
 "Country": "US",  
 "JurisName": "MINNESOTA",  
 "JurisCode": 27,  
 "JurisType": "State",  
 "Rate": 0.06875,  
 "Region": "MN",  
 "Tax": 4.64,  
 "TaxName": "MN STATE TAX"  
 },  
 {  
 "Country": "US",  
 "JurisName": "HENNEPIN",  
 "JurisCode": 53,  
 "JurisType": "County",  
 "Rate": 0.0015,  
 "Region": "MN",  
 "Tax": 0.1,  
 "TaxName": "MN COUNTY TAX"  
 },  
 {  
 "Country": "US",  
 "JurisName": "MINNEAPOLIS",  
 "JurisCode": 43000,  
 "JurisType": "City",  
 "Rate": 0.005,  
 "Region": "MN",  
 "Tax": 0.34,  
 "TaxName": "MN CITY TAX"  
 },  
 {  
 "Country": "US",  
 "JurisName": "TRANSIT IMPROVEMENT TAX",  
 "JurisCode": 80002,  
 "JurisType": "Special",  
 "Rate": 0.0025,  
 "Region": "MN",  
 "Tax": 0.17,  
 "TaxName": "MN SPECIAL TAX"  
 }  
 ]  
 }  
 ],  
 "TaxAddresses": [  
 {  
 "Address": "110 North 5th Street",  
 "AddressCode": "1111",  
 "City": "Minneapolis",  
 "Region": "MN",  
 "Country": "US",  
 "PostalCode": "55403",  
 "Latitude": null,  
 "Longitude": null,  
 "TaxRegionId": 2076413,  
 "JurisCode": "2705343000"  
 }  
 ],  
 "ResultCode": "Success"  
}**

## **Error Handling**

**If there are any errors in processing the tax request from Avalara there will be a status code of 500 returned with a detailed message of the error. Any errors that are produced during calls to the OrderCloud.io API will return our standard status code error responses.**

### **Avalara Error Response**

**If the incorrect or missing companyCode is sent you will receive this response:**

**HTTP/1.1 201 Created  
Content-Type: application/json  
  
{  
 "ResultCode": "Error",  
 "Messages": [  
 {  
 "Summary": "Company not found. Verify the CompanyCode.",  
 "RefersTo": "CompanyCode",  
 "Severity": "Error",  
 "Source": "Avalara.AvaTax.Services.Tax",  
 "Details": "00450"  
 }  
 ]  
}**

### **Four51 Error Response**

**If an incorrect buyerID was passed in:**

**HTTP/1.1 201 Created  
Content-Type: application/json  
  
{  
 "Errors": [{  
 "ErrorCode": "NotFound",  
 "Message": "Buyer not found: 1234",  
 "Data": null  
 }]  
}**

**If an incorrect orderID was passed in:**

**HTTP/1.1 201 Created  
Content-Type: application/json  
  
{  
 "Errors": [{  
 "ErrorCode": "NotFound",  
 "Message": "Order not found: 888",  
 "Data": null  
 }]  
}**