End User Documentation

Product Development

Exported on Jul 27, 2018

Table of Contents

1 Avlara Extension Installation 3

1.1 Extension Installation 3

1.2 Installation Steps 3

1.3 Related articles 4

2 Creating new Avalara Settings 5

3 Base Store Settings 8

3.1 Base Store Configurations 8

3.2 Related articles 9

4 Product Tax Codes 10

4.1 Creating a new Product Tax Code 10

4.2 Editing an existing Product Tax Code 11

4.3 Deleting an existing Product Tax Code 12

4.4 Related articles 12

5 ArvatoAvalaraTaxConnectorService 13

# Avlara Extension Installation

## Extension Installation

|  |
| --- |
| **Pre-requesites**   1. Minimum hybris 640 platform installation. 2. ArvatoAvalaraExt extension [arvatoAvalaraConnector.zip](https://www.arvatosystems-us.com/confluence/download/attachments/30875597/arvatoAvalaraConnector.zip?version=1&modificationDate=1525187095000&api=v2) 3. An avalara account with a valid company Id and license |

## Installation Steps

1. if hybris is running be sure to stop first by navigating to ${HYBRIS\_HOME}/bin/platform/ and executing one of the two scripts based on whether a linux based or windows based server is running.
   1. **For Linux:** sh hybrisserver.sh stop
   2. **For Windows:**hybrisserver stop
2. Unzip the [arvatoAvalaraConnector.zip](https://www.arvatosystems-us.com/confluence/download/attachments/30875597/arvatoAvalaraConnector.zip?version=1&modificationDate=1525187095000&api=v2) folder into the ${HYBRIS\_HOME}/bin/platform/custom directory
3. edit the localextensions.xml located in ${HYBRIS\_HOME}/config and add the below entry to the <extensions node>

Code Block 1 localextensions.xml

|  |
| --- |
| <extensions>  <path dir='${HYBRIS\_BIN\_DIR}' autoload='false' />  <path dir='${HYBRIS\_BIN\_DIR}/ext-integration/sap/' autoload='false' />  .  .  .  <extension name='arvatoAvalaraConnector' /> </extensions> |

1. Navigate to {PLATFORM\_HOME}/bin/platform and execute the ant all target to compile and build the project
2. Once the build is complete start the hybris server by navigating to the ${HYBRIS\_HOME}/bin/platform/ directory and running one the appropriate script based on the operating system
   1. **For Linux:** sh hybrisserver.sh start
   2. **For Windows:** hybrisserver start
3. One hybris has started run the system update by opening the hybris admin console using the browser of your choice. If installing locally the url would be [https://127.0.0.1:9002/platform/update.](https://127.0.0.1:9002/platform/update)
4. Under the Project Data Settings croll down to the arvatoAvalaraConnector and check the box next to it

|  |
| --- |
| Check Box |
| /confluence/download/attachments/30875597/image2018-5-1%2010%3A10%3A47.png?version=1&modificationDate=1525187095000&api=v2 |

1. Finally click on the Update Button to allow the appropriate tables to be added to the hybris databas

|  |
| --- |
|  |

## Related articles

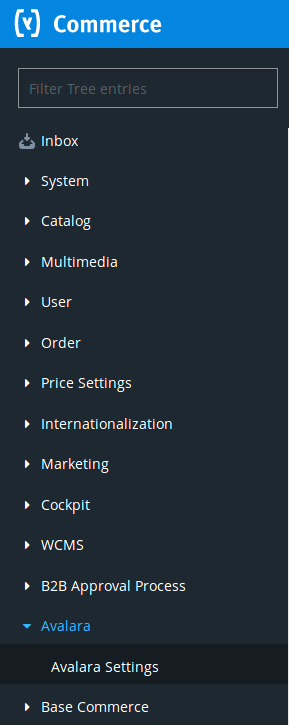
* Page: [API Generation](https://www.arvatosystems-us.com/confluence/display/PRODDEV/API+Generation)
* Page: [Base Store Settings](#scroll-bookmark-6)
* Page: [Avlara Extension Installation](#scroll-bookmark-2)
* Page: [Hybris Services injection and use cases](#scroll-bookmark-7)
* Page: [Product Tax Codes](#scroll-bookmark-8)

[6 related results](https://www.arvatosystems-us.com/confluence/dosearchsite.action?where=PRODDEV&queryString=(labelText:kb-how-to-article)%20AND%20(type:page))

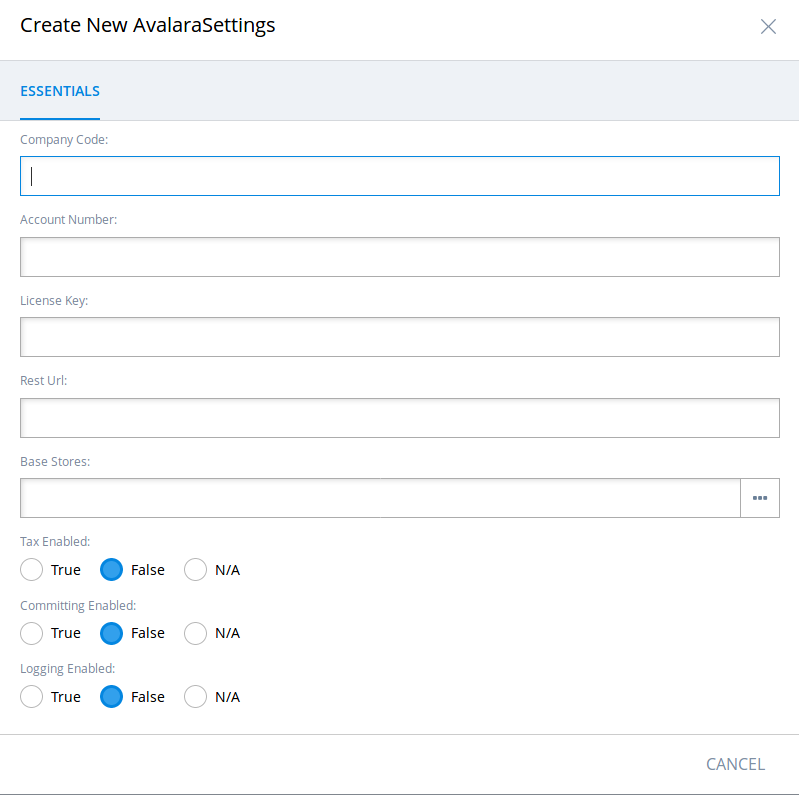
# Creating new Avalara Settings

1) Login into the Backoffice, in development environments you can use *admin/nimda*.

2) In the navigation tree click on *Avalara Settings* found under *Avalara*.



3) Click on the *+* icon to open the Creation Wizard popup.



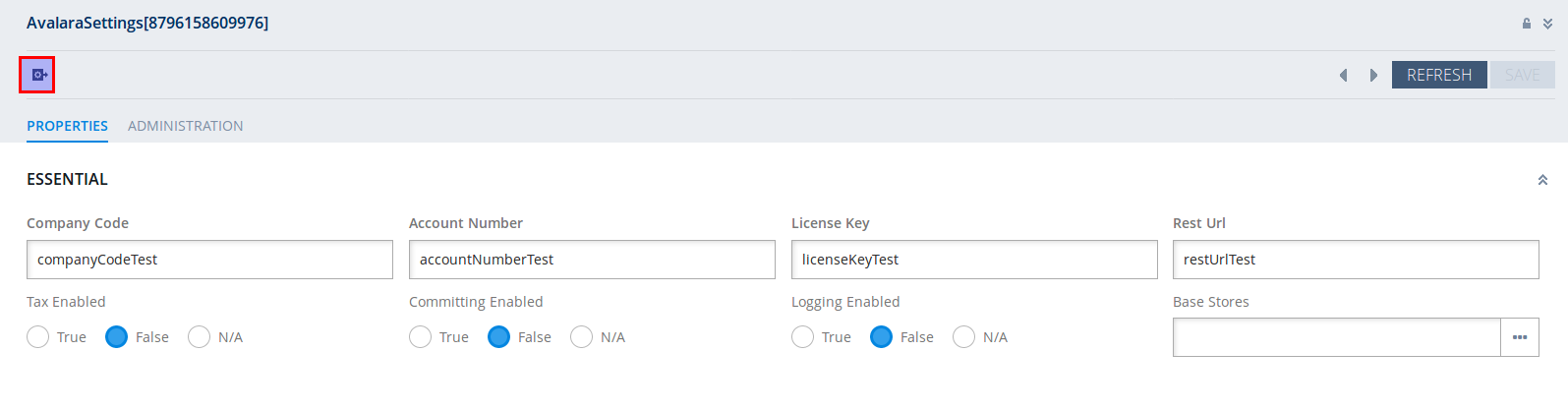
4) The mandatory fields are Company Code, Account Number, License Key, and Rest Url. Once these are provided the *Done* button will be available to be clicked on.

By default the following attributes are set to false:

* taxEnabled: If this is *false,* requests will not be sent to Avalara.
* committingEnabled: If this is *false*, transactions will not be marked as *Committed* in Avalara and therefore will not be available to be reported to a tax authority by Avalara Managed Returns.
* logginEnabled: If this is *false*, requests between Hybris and Avalara will not be logged in the console.

By default the Base Stores attribute will be emtpy. When used in a store context, the first Avalara Settings found will be used if none is specified.

5) Once created, the settings can be modified and tested in the editor view. To test a Setting click on the Ping icon on the view's action bar. You should see a Ping Success or Error message.



# Base Store Settings

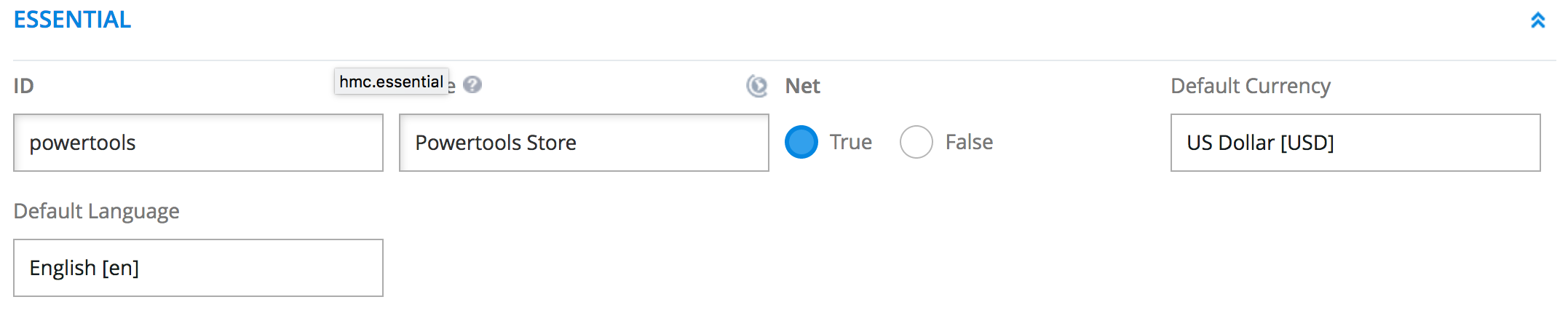
|  |
| --- |
| In order for avlara to be used every store will need to be configured as below. |

## Base Store Configurations

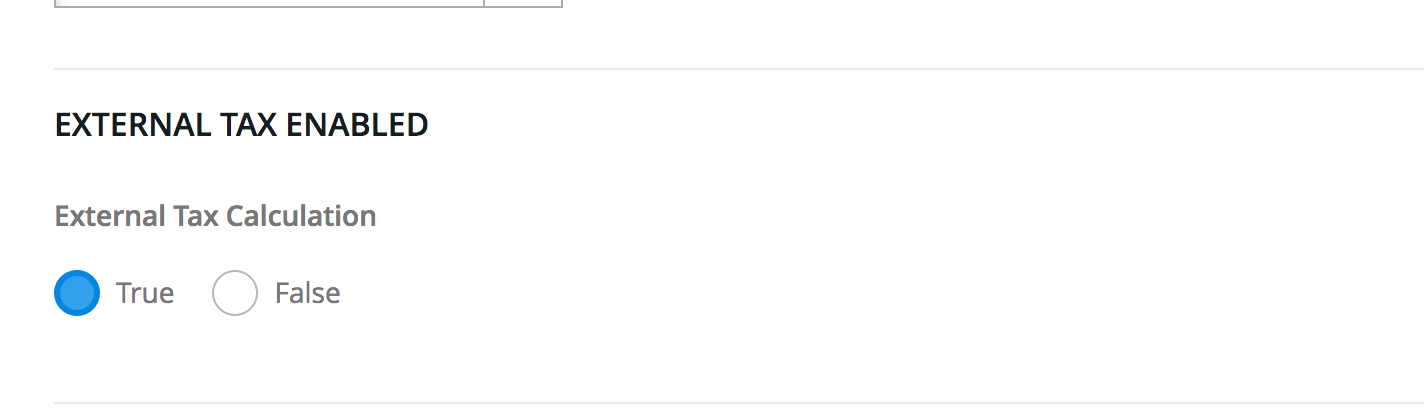
1. In the backoffice navigation tree expand the Base Commerce Node and click on Base Store to open the Base Store search window. Click the store from the result screen to populate the detail view.

|  |  |
| --- | --- |
| Navigation Tree | Detail View |
| /confluence/download/attachments/30875605/image2018-5-1%2011%3A46%3A26.png?version=1&modificationDate=1525189573000&api=v2 | /confluence/download/attachments/30875605/image2018-5-1%2011%3A48%3A51.png?version=1&modificationDate=1525189718000&api=v2 |

1. Under the Properties tab select true on the net property



1. Scroll down the window to locate the External Tax Enabled Setting and select true on that property.



|  |
| --- |
|  |

## Related articles

* Page: [API Generation](https://www.arvatosystems-us.com/confluence/display/PRODDEV/API+Generation)
* Page: [Base Store Settings](#scroll-bookmark-6)
* Page: [Avlara Extension Installation](#scroll-bookmark-2)
* Page: [Hybris Services injection and use cases](#scroll-bookmark-7)
* Page: [Product Tax Codes](#scroll-bookmark-8)

[6 related results](https://www.arvatosystems-us.com/confluence/dosearchsite.action?where=PRODDEV&queryString=(labelText:kb-how-to-article)%20AND%20(type:page))

# Product Tax Codes

## Creating a new Product Tax Code

1. log in to the hybris backoffice <https://127.0.0.1:9002/backoffice/>
2. In the navigation tree expand the Avlara node and double click on Product Tax Codes to open the TaxCode Listing.

|  |  |
| --- | --- |
| Navigation Tree | Result List |
| /confluence/download/attachments/30875470/image2018-4-20%208%3A59%3A22.png?version=1&modificationDate=1524230637000&api=v2 | /confluence/download/attachments/30875470/image2018-4-20%209%3A0%3A39.png?version=1&modificationDate=1524230637000&api=v2 |

1. In the upper left click the plus sign to open the item creation form The fields to enter are as below. All fields must be unique there can not be duplicated records to Product Code and Tax Area. For example a product with code 2231913 can not have more than one code in the US area.
   1. Tax Area - The two character Iso code of the country i.e. US for United States, GB for Great Britain, JP for Japan. **This field is not editable once the entry is created**
   2. Product Code - the code of the Product in hybris**. This field is not editable once the entry is created**
   3. Tax Code - the avalara tax code i.e. PC080601

|  |
| --- |
| Product Tax Code entry form |
| /confluence/download/attachments/30875470/image2018-4-20%209%3A7%3A46.png?version=1&modificationDate=1524230637000&api=v2 |

1. Once the form has been entered click done at the lower left. If there is already an entry in hybris for the Product Code An error will show at the top of the form like below. If this happens the existing entry for the product code will need to be edited.

|  |
| --- |
| Error |
| /confluence/download/attachments/30875470/image2018-4-20%209%3A11%3A9.png?version=1&modificationDate=1524230637000&api=v2 |

## Editing an existing Product Tax Code

1. In the Backoffice explorer tree navigate to Avlara Settings -> Product Tax Codes (see screenshot above)
2. In the TaxCode listing enter either the Product Code or TaxCode and click Search to narrow down the results. Select the Entry you would like to delete and click the trash can at the upper left to remove it from the hybris database. A confirmation dialog will open and click yes to confirm the deletion

|  |  |
| --- | --- |
| Searching for product 23193 | Confirmation dialog |
| /confluence/download/attachments/30875470/image2018-4-20%209%3A16%3A48.png?version=1&modificationDate=1524230637000&api=v2 | /confluence/download/attachments/30875470/image2018-4-20%209%3A17%3A57.png?version=1&modificationDate=1524230637000&api=v2 |

## Deleting an existing Product Tax Code

1. In the Backoffice explorer tree navigate to Avlara Settings -> Product Tax Codes (see screenshot above)
2. In the TaxCode listing enter either the Product Code or TaxCode and click Search to narrow down the results (see screen shot above).
3. Double click the entry matching the country that needs to be edited in order to populate the detail view in the center panel. The only field that may be changed is the tax code. If any other field needs to be changed the record must be deleted and then re-created.

|  |
| --- |
| Detail view |
| /confluence/download/attachments/30875470/image2018-4-20%209%3A23%3A34.png?version=1&modificationDate=1524230637000&api=v2 |

## Related articles

* Page: [API Generation](https://www.arvatosystems-us.com/confluence/display/PRODDEV/API+Generation)
* Page: [Base Store Settings](#scroll-bookmark-6)
* Page: [Avlara Extension Installation](#scroll-bookmark-2)
* Page: [Hybris Services injection and use cases](#scroll-bookmark-7)
* Page: [Product Tax Codes](#scroll-bookmark-8)

[6 related results](https://www.arvatosystems-us.com/confluence/dosearchsite.action?where=PRODDEV&queryString=(labelText:kb-how-to-article)%20AND%20(type:page))

# ArvatoAvalaraTaxConnectorService

1. The default hybris out of the box usage will only make use of one method in the CalculateExternalTaxesStrategy which is calculateExternalTaxes(AbstractOrderModel). This will do one of two actions based on the type of object that is passed in.
   1. **CartModel** will make a quote request against avalara to get the calculated tax. This will not create any documents in the avalara administration console
   2. **OrderModel** will create a post request against avalara to calculate the taxes and create a document in COMMITTED status in the avalara administration conosle.
2. There is also support for cancellations, invoices, checks for previously created documents, adjustments, and tests for avalara connectivity. With the exception of pingThe return value for all of these will be an Optional<ExternalTaxDocument>. If there is nothing returned by avalara the response will be Optional.empty().
   1. **CANCEL**
      1. In order to void you will need to use the ArvatoAvalaraTaxConnectorService interface and inject it into the hybris cancellation service that is used. Consider the sample java class below that will set an order status to CANCELLED and then cancel the tax document. Also note the spring bean definition immediately after.

Code Block 2 Cancel Example

|  |
| --- |
| public class SampleCancelRequestExecutor implements OrderCancelRequestExecutor {   private ModelService modelService;  private ArvatoAvalaraTaxConnectorService arvatoAvalaraTaxConnectorService;   public void processCancelRequest(final OrderCancelRequest orderCancelRequest,  final OrderCancelRecordEntryModel cancelRequestRecordEntry) throws OrderCancelException  {  AbstractOrderModel orderModel = cancelRequestRecordEntry.getConsignment().getOrder();  orderModel.setStatus(OrderStatus.CANCELLED);  arvatoAvalaraTaxConnectorService.cancel(orderModel);  } |

* 1. **INVOICE**
     1. This method is used to make a request to avalara to get the tax calculations. It will also create a sales invoice in uncommitted status in the avalara admin console. This would be used on a newly created order. Consider the below example for a CalculationService that will calculate the order total and send an invoice request to avalara

Code Block 3 Invoice Example

|  |
| --- |
| public class OrderCalculationService implements CalculationService{  private ArvatoAvalaraTaxConnectorService arvatoAvalaraTaxConnectorService;   public void calculate(AbstractOrderModel order){  double subtotal = 0.0;  for (final AbstractOrderEntryModel e : order.getEntries())  {  someMethodThatCalculatesTheOrderEntryTotal(e);  subtotal += e.getTotalPrice().doubleValue();  }  order.setTotalPrice(Double.valueOf(subtotal));  if(order instanceof OrderModel){  arvatoAvalaraTaxConnectorService.invoice(order);  }  } } |

* 1. **EXISTS** 
     1. this method can be used to see it there is already a tax documented created for an existing order in avalara. Building on the above example we can add a check to first see there was already a tax request and for the order and if there is it will not make an invoice request.

Code Block 4 Exists example

|  |
| --- |
| public class OrderCalculationService implements CalculationService{  private ArvatoAvalaraTaxConnectorService arvatoAvalaraTaxConnectorService;   public void calculate(AbstractOrderModel order){  double subtotal = 0.0;  for (final AbstractOrderEntryModel e : order.getEntries())  {  someMethodThatCalculatesTheOrderEntryTotal(e);  subtotal += e.getTotalPrice().doubleValue();  }  order.setTotalPrice(Double.valueOf(subtotal));  if(!order instanceof OrderModel){  return;  }  if(arvatoAvalaraTaxConnectorService.exists(order).empty()){  arvatoAvalaraTaxConnectorService.invoice(order);  }  } } |

* 1. **ADJUST**
     1. This can be used to re-calculate taxes on an order whose amount has changed. Consider the below piece of code that calculates an existing OrderEntry and once done will send a request to avalara to adjust the existing tax document. Building yet again on the above example if there was already a tax document instead of just making no call to avalara an adjustment call can be made in this case.

Code Block 5 Adjust Example

|  |
| --- |
| public class OrderCalculationService implements CalculationService{  private ArvatoAvalaraTaxConnectorService arvatoAvalaraTaxConnectorService;   public void calculate(AbstractOrderModel order){  double subtotal = 0.0;  for (final AbstractOrderEntryModel e : order.getEntries())  {  someMethodThatCalculatesTheOrderEntryTotal(e);  subtotal += e.getTotalPrice().doubleValue();  }  order.setTotalPrice(Double.valueOf(subtotal));  if(!order instanceof OrderModel){  return;  }  if(arvatoAvalaraTaxConnectorService.exists(order).empty()){  arvatoAvalaraTaxConnectorService.invoice(order);  }  else{  arvatoAvalaraTaxConnectorService.adust(order);  } } |

* 1. **PING**
     1. This method cn be used to ensure there is an active connection to avalara. It will fail if there is a networking issue or if the user credentials are misconfigured in hybris. Building on the Calculation example from above

Code Block 6 Ping Example

|  |
| --- |
| public class OrderCalculationService implements CalculationService{  private ArvatoAvalaraTaxConnectorService arvatoAvalaraTaxConnectorService;   public void calculate(AbstractOrderModel order){  double subtotal = 0.0;  for (final AbstractOrderEntryModel e : order.getEntries())  {  someMethodThatCalculatesTheOrderEntryTotal(e);  subtotal += e.getTotalPrice().doubleValue();  }  order.setTotalPrice(Double.valueOf(subtotal));  Optional<Boolean> pingResponse = arvatoAvalaraTaxConnectorService.ping();  if(!order instanceof OrderModel || pingResponse.empty() || Boolean.False.equals(pingResponse.get())){  return;  }  if(arvatoAvalaraTaxConnectorService.exists(order).empty()){  arvatoAvalaraTaxConnectorService.invoice(order);  }  else{  arvatoAvalaraTaxConnectorService.adust(order);  } } |

* 1. **POST**
     1. This method can be used to mark a previously invoiced tax document as committed. This may be used after an order has shipped to commit the tax document in avalara. Consider the below implementation of a Warehouse2ProcessAdapter that will set the consignmentStatus to shipped and then check to see if all consignments have been shipped. If all consignments are shipped the post method will be invoked. The default implementation of the arvatoAvalaraTaxConnectorService will always send the document as committed when first created so it should not be necessary to manually commit documents.

Code Block 7 Post example

|  |
| --- |
| public class ShippingAdapter implements Warehouse2ProcessAdapter{  private ModelService modelService;  private ArvatoAvalaraTaxConnectorService arvatoAvalaraTaxConnectorService;   public void receiveConsignmentStatus(final ConsignmentModel consignment, final WarehouseConsignmentStatus status){  consignment.setStatus(ConsignmentStatus.SHIPPED);  modelService.save(consignment);  boolean allShipped = true;  for(ConsignmentModel consignment2 : consignment.getOrder().getConsignments()){  allShipped = allShipped && (ConsignmentStatus.SHIPPED.equals(consignment2.getStatus()));  }  if(allShipped){  arvatoAvalaraTaxConnectorService.post(consignment.getOrder());  }  } } |

**ArvatoAddressVerificationService**

1. The ArvatoAddressVerificationService Does not have any new functionality. It is an implementation of the hybris out of the box AddressVerificationService. To inject the bean add the below into your spring.xml of your project.

Code Block 8 AvsInjection

|  |
| --- |
| <alias name="arvatoAddressVerificationService" alias="addressVerificationService"/>  <bean id="arvatoAddressVerificationService"  class="com.arvatosystems.avalara.service.hybris.ArvatoAddressVerificationService"/> |

**AvalaraDecideExternalTaxesStrategy**

1. We have implemented the DecideExternalTaxesStrategy interface to customize the default shouldCalculateExternalTaxes method. This returns a boolean that defines if tax calls will be made to Avalara. To modify this logic you will need to change the following alias in the spring.xml of your project.

Code Block 9 decideExternalTaxesStrategy

|  |
| --- |
| <alias name="avalaraDecideExternalTaxesStrategy" alias="decideExternalTaxesStrategy"/> |

* Page: [API Generation](https://www.arvatosystems-us.com/confluence/display/PRODDEV/API+Generation)
* Page: [Base Store Settings](#scroll-bookmark-6)
* Page: [Avlara Extension Installation](#scroll-bookmark-2)
* Page: [Hybris Services injection and use cases](#scroll-bookmark-7)
* Page: [Product Tax Codes](#scroll-bookmark-8)

[6 related results](https://www.arvatosystems-us.com/confluence/dosearchsite.action?where=PRODDEV&queryString=(labelText:kb-how-to-article)%20AND%20(type:page))