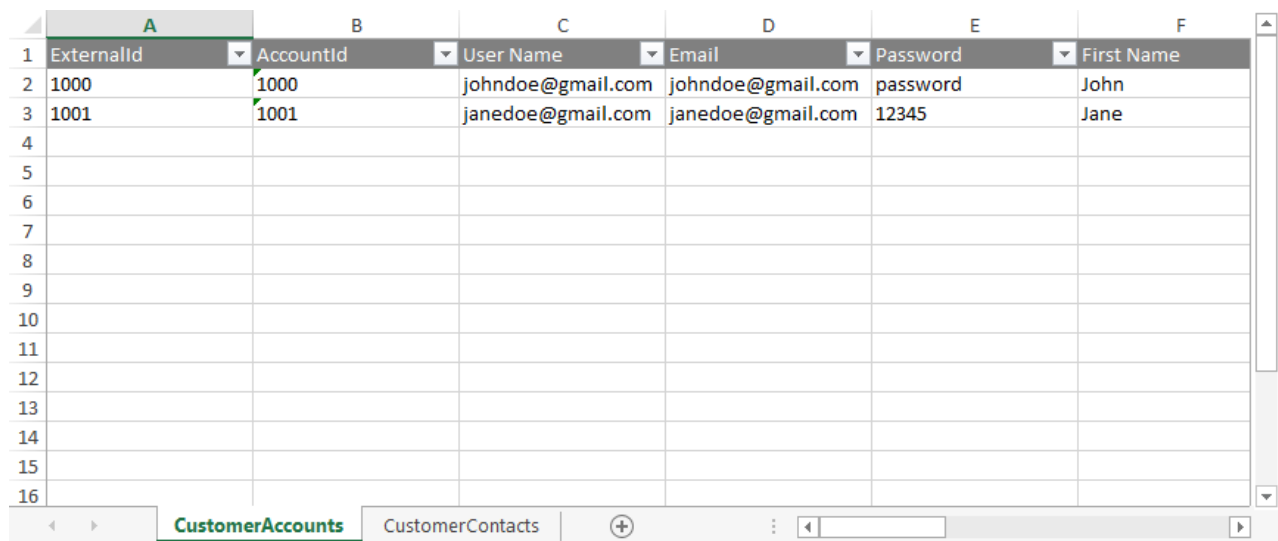


# Customers Template

The Customers template for the [import/export tools](#) describes your customer accounts and contact information. This data is used to manage orders, subscribed messages, promotional emails, newsletters, and so on. If you are preparing your [initial data import for a new Kibo eCommerce tenant](#), import customers *first*.

Merchants can also import/export customers with associated customer segments. An option at the top of the attribute selection dropdown will allow you to specify filtering on customer segment information. The export action exports all segments to a file when "customer segment" is selected. The import action updates existing segments and adds any new segments to the Customer Segment page within admin.



	A	B	C	D	E	F
1	ExternalId	AccountId	User Name	Email	Password	First Name
2	1000	1000	johndoe@gmail.com	johndoe@gmail.com	password	John
3	1001	1001	janedoe@gmail.com	janedoe@gmail.com	12345	Jane
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

## Access the Tools

With the [Kibo eCommerce Import-Export Application](#) installed and enabled, go to **Main > Customers > Import/Export** in Kibo eCommerce Admin.

## Supported File Formats

- **CSV (.csv):** Each sheet is a separate CSV file. For both imports and exports, sheets are compressed into a ZIP file. When importing, the ZIP file can have any file name but each individual CSV file name must match the export file name and the sheet. You can remove CSV files for data you do not want to import.

These CSV files should not be opened in Excel, as that may cause errors with their formatting. Instead, use a different CSV editing program to make changes to the data.

You should only use the default Windows zipper to zip the files, as using different software may cause errors. You can upload multiple files at once as long as they all are zipped.

# Template Key

For each sheet, we define each column and describe valid values for the column. Some values are not required for an import, but are required Kibo eCommerce data. For example, you do not have to specify a customer's `FirstName` every time you modify the customer account, but all customer accounts require a `FirstName` when they are created. If you have questions or concerns specific to your data, please contact your integration partner or .

## Information Provided for Each Sheet of the Template

Corresponding REST API Resource	Every column in a sheet corresponds to a property in the Kibo eCommerce REST API. For each sheet, we provide a link to any API resources that contain properties in the sheet.
Column Name	The name of the column in the sheet.
Description	A description of the data a column contains.
Valid Values	Lists valid values for the column.

## Sheets

### Accounts

Corresponding REST API Resource: [commerce/customer/accounts](#)

Required With: [CustomerContacts](#)

This sheet defines customer accounts within Kibo eCommerce. Customer accounts can be registered shopper accounts or guest accounts. Registered accounts are created through a registration event, typically completed by the customer. Guest accounts are created when a customer chooses not to register. Guest accounts still have account IDs and associated data necessary to fulfill orders.

Column Name	Description	Valid Values
-------------	-------------	--------------

Column Name	Description	Valid Values
ExternalId	A unique identifier you can use to refer to the customer in import/export operations or the Kibo eCommerce API. Typically, this is a value that is used by an external program to identify a customer in Kibo eCommerce. This value is not exposed in Admin.	Any string that is unique among all customer accounts.
AccountID	The unique identifier for the customer account. Kibo eCommerce automatically generates this value when a customer creates their own account or when you create a new customer account from Admin. This is the same as the <b>Customer Id</b> in other sheets.	Any integer that is unique among all customer accounts. Can be the same or different than the Id.
User Name	The username the customer uses to log in to the account.	Any string. If empty, the username defaults to the Email.

Column Name	Description	Valid Values
Email	The email address associated with the account.	Any string. The email address should be unique across all registered customer accounts. A Kibo eCommerce customer cannot create two separate registered accounts linked to the same email address. However, for import operations, you can use the same email address any number of times, provided it is associated with a unique AccountID each time. Also, customers who check out as guests can use the same email address any number of times.
Password	The password for the account. Setting a password creates a registered customer account.	Any string.
First Name	The first name on the account.	Any string.
Last Name	The last name on the account.	Any string.
Company Or Organization	The company or organization the customer works for.	Any string.
Tax Exempt	Specifies whether the account has tax exempt status.	A Boolean value, specified as <b>Yes</b> or <b>No</b> .
TaxId	The tax ID associated with the account.	Any string.
Accepts Marketing	Specifies whether the customer has agreed to receive marketing communications.	A Boolean value, specified as <b>Yes</b> or <b>No</b> .

Column Name	Description	Valid Values
Is Anonymous	Specifies whether the customer account is anonymous. If <b>Yes</b> the customer is shopping as a guest and not using a registered account.	A Boolean value, specified as <b>Yes</b> or <b>No</b> .
Customer Set	The customer set that the customer account belongs to. This field is optional. Each unique customer account can only belong to one customer set. For more information, refer to the <a href="#">Customer Sets</a> topic.	The name of the customer set you want to assign the customer account to. The customer set must already exist. You cannot create new customer sets through the Kibo eCommerce Import-Export Application.
Customer Segments	The customer segments that the customer account belongs to. This field is optional. For more information, refer to the <a href="#">Customer Segments</a> topic.	The name of the customer segment you want to assign the customer account to. The customer segment must already exist.
Total Order Amount	The total amount, in the currency specified by <code>LocaleCode</code> , of all orders that have been processed on this account. This value includes cancellations and refunds.	Any decimal value.
Total Orders	The total number of orders that have been processed on this account.	Any integer value.

Column Name	Description	Valid Values
Last Order Date	The date and time at which the last order was placed on this account.	A date and time in UTC format.
Last Modified Date	The date and time that the customer account was last edited, either by the customer or by the system.	A date and time in UTC format.

## Contacts

Corresponding REST API Resource: [commerce/customer/accounts/contacts](#)

Required With: [CustomerAccounts](#)

This sheet defines the contacts for customer accounts. Contacts include such information as telephone numbers, shipping and billing addresses, and so on. A single customer account can have multiple contacts associated with it.

Column Name	Description	Valid Values
ExternalId	A unique identifier you can use to refer to the customer in import/export operations or the Kibo eCommerce API. Typically, this is a value that is used by an external program to identify a customer in Kibo eCommerce. This value is not exposed in Admin.	Any string that is unique among all customer accounts.
AccountID	The unique identifier for the customer account. Kibo eCommerce automatically generates this value when a customer creates their own account or when you create a new customer account from Admin. This is the same as the <b>Customer Id</b> in other sheets.	Any integer that is unique among all customer accounts. Can be the same or different than the Id.

Column Name	Description	Valid Values
ContactId	The unique identifier for a specific contact within a customer account. This value is not exposed in Admin.	Any string that is unique among all contact IDs on the tenant. Typically, this will be a number. For accounts with only one contact, the ContactID usually matches the AccountID.
Type	The type of the contact.	A string matching one of the following: <ul style="list-style-type: none"> <li>Shipping</li> <li>Billing</li> </ul>
IsPrimary	Specifies whether the contact is the primary contact of the specified Type for the customer account.	A Boolean value, specified as Yes or No .
Company Or Organization	The company or organization associated with the contact. This can differ from the CompanyOrOrganization on the customer account. For example, a customer with an account from Company X might order a product and ship it to Company Y.	Any string.
First Name	The first name for the contact. This can differ from the name on the customer account.	Any string.
Middle Name	The middle name or initial for the contact.	Any string.
Last Name	The last name or surname for the contact.	Any string.
Email	The email address associated with the contact.	Any string. Kibo eCommerce does not validate email addresses.
Fax Number	The fax number associated with the contact.	Any string. You can include spaces and characters.

Column Name	Description	Valid Values
Home Phone	The home phone number associated with the contact.	<p>Any string. You can include spaces and characters.</p> <p>For a contact to be valid in Kibo eCommerce, it must include at least one of the following: HomePhone, MobilePhone, or WorkPhone.</p>
Mobile Phone	The mobile phone number associated with the contact.	<p>Any string. You can include spaces and characters.</p> <p>For a contact to be valid in Kibo eCommerce, it must include at least one of the following: HomePhone, MobilePhone, or WorkPhone.</p>
Work Phone	The work phone number associated with the contact.	<p>Any string. You can include spaces and characters.</p> <p>For a contact to be valid in Kibo eCommerce, it must include at least one of the following: HomePhone, MobilePhone, or WorkPhone.</p>
Address Type	The type of the address, such as Residential or Commercial.	<p>A string matching one of the following:</p> <ul style="list-style-type: none"> <li>• Residential</li> <li>• Commercial</li> <li>• None</li> </ul>
Address1	The first line of the address.	Any string. Character limit: 200.



Column Name	Description	Valid Values
Address2	The second line of the address.	Any string. Character limit: 200.
Address3	The third line of the address.	Any string. Character limit: 200.
Address4	The fourth line of the address.	Any string. Character limit: 200.
City Or Town	The city or town name for the address.	Any string.
State Or Province	The state, province, or territory for the address, typically expressed as a two-letter code. For example, TX.	Any string. You can enter a location outside the US or Canada.
Postal Or ZipCode	The postal code or zip code for the address.	Any string. This can be a full 9-digit code with a dash or an international postal code.
Country Code	The 2-letter ISO country code for the address.	Any 2-letter country code.
Last Modified Date	The date and time that the contact was last edited, either by the customer or by the system.	A date and time in UTC format.

## PurchaseOrderAccounts

Corresponding REST API Resource: [commerce/customer/accounts/purchaseOrder](#)

This sheet defines [purchase order](#) accounts within Kibo eCommerce.

Column Name	Description	Valid Values
ExternalId	A unique identifier you can use to refer to the customer in import/export operations or the Kibo eCommerce API. Typically, this is a value that is used by an external program to identify a customer in Kibo eCommerce. This value is not exposed in Admin.	Any string that is unique among all customer accounts.

Column Name	Description	Valid Values
AccountID	The unique identifier for the customer account. Kibo eCommerce automatically generates this value when a customer creates their own account or when you create a new customer account from Admin. This is the same as the <code>Customer Id</code> in other sheets.	Any integer that is unique among all customer accounts. Can be the same or different than the Id.
Enabled	Whether purchase orders are enabled as a payment method for the customer account.	<code>Yes</code> or <code>No</code> .
Credit Limit	The total the customer can spend using purchase orders.	Any decimal amount.
Overdraft Allowance Type	Specifies the type and units of the <code>overdraftAllowance</code> amount.	<code>Amount</code> — specifies that the overdraft allowance amount is a decimal amount.  <code>Percent</code> — specifies that the overdraft allowance amount is a percentage amount.
Overdraft Allowance	The total monetary amount or percentage the customer can spend over their purchase order credit limit. The overdraft allowance type controls whether this value is an amount or a percentage.	Any decimal amount.
Available Balance	The available balance the customer has on their purchase order account. This balance is the result of the credit limit minus the customer's total amount due.	Any decimal amount.

## PurchaseOrderPaymentTerms

Corresponding REST API Resource: [commerce/customer/accounts/purchaseOrder](#)

This sheet allows you to assign [payment terms](#) within a site to specific customer accounts.

Column Name	Description	Valid Values
ExternalId	A unique identifier you can use to refer to the customer in import/export operations or the Kibo eCommerce API. Typically, this is a value that is used by an external program to identify a customer in Kibo eCommerce. This value is not exposed in Admin.	Any string that is unique among all customer accounts.
AccountID	The unique identifier for the customer account. Kibo eCommerce automatically generates this value when a customer creates their own account or when you create a new customer account from Admin. This is the same as the <code>Customer Id</code> in other sheets.	Any integer that is unique among all customer accounts. Can be the same or different than the Id.
SiteId	The unique identifier for a site.	Any number. This number must be unique among your sites.
Site Name	The name of a site.	Any string.
Code	User-defined code that uniquely identifies the payment term.	Any string.

## PurchaseOrderTransactions

Corresponding REST API Operation: [GetCustomerPurchaseOrderTransactions](#)

This sheet defines [purchase order](#) transactions for customer accounts.

Column Name	Description	Valid Values
AccountID	The unique identifier for the customer account. Kibo eCommerce automatically generates this value when a customer creates their own account or when you create a new customer account from Admin. This is the same as the <code>Customer Id</code> in other sheets.	Any integer that is unique among all customer accounts. Can be the same or different than the Id.

Column Name	Description	Valid Values
Site Name	The name of a site.	Any string.
Order Number	The order number that the customer sees on the storefront when they place the order.	Any integer that is unique among all order numbers.
Order Type	Specifies how the order was placed.	Either <b>Offline</b> or <b>Online</b> .
Purchase Order Number	A unique identifier for the purchase order.	Any alphanumeric string.
Transaction Amount	The amount of the purchase order transaction.	Any decimal value. A positive value increases the customer's available balance, while a negative number decreases the available balance.
Transaction Type Id	A number that identifies the type of transaction and displays it within the purchase orders transaction log.	<p>The following list describes the types of transactions each number corresponds to:</p> <ul style="list-style-type: none"> <li>1 — <b>Line of Credit Change</b></li> <li>2 — <b>Payment Requested</b></li> <li>3 — <b>Payment Collected</b></li> <li>4 — <b>Payment Voided</b></li> <li>5 — <b>Payment Refund</b></li> <li>6 — <b>Manual Adjustment</b></li> </ul>

Column Name	Description	Valid Values
Transaction Description	The string that describes the transaction type in the purchase orders transaction log.	<p>This field is read-only and corresponds to the transaction type ID:</p> <p>1 — Line of Credit Change</p> <p>2 — Payment Requested</p> <p>3 — Payment Collected</p> <p>4 — Payment Voided</p> <p>5 — Payment Refund</p> <p>6 — Manual Adjustment</p>
Author	The author of the purchase order transaction.	A string that corresponds to the Admin user name of the transaction author or to the name of the API version or application that created the transaction.
Transaction Date	The date the transaction took place.	A date. For example, 6/8/2016 7:35:23 PM .
Additional Transaction Detail	Additional transaction details or notes that you want to associate with the transaction.	Any string.
Credit Limit	The total the customer can spend using purchase orders.	Any decimal amount.

Column Name	Description	Valid Values
Available Balance	The available balance the customer has on their purchase order account. This balance is the result of the credit limit minus the customer's total amount due.	Any decimal amount.

## StoreCredits

Corresponding REST API Resource: [commerce/customer/credits](#)

This sheet allows you to assign [store credit](#) to specific customer accounts.

Column Name	Description	Valid Values
Activation Date	The date the store credit is available for the customer to use.	A date. For example, <code>9/27/2016 12:00:00 AM</code> . If blank, the store credit is immediately available.
Code	User-defined code that uniquely identifies the store credit.	Any string.
Credit Type	The type of store credit.	Either <code>StoreCredit</code> or <code>GiftCard</code> .
Currency Code	The currency for the store credit.	Any three-letter currency code. For example: <code>USD</code> .
Current Balance	The current balance to set for the store credit.	Any decimal value.
Customer Id	The unique identifier for the customer account. Kibo eCommerce automatically generates this value when a customer creates their own account or when you create a new customer account from Admin. This is the same as the <code>AccountId</code> in other sheets.	Any integer that is unique among all customer accounts.

Column Name	Description	Valid Values
Expiration Date	The date the store credit is no longer available for the customer to use.	A date. For example, 9/27/2016 12:00:00 AM . If blank, the store credit does not expire.
Initial Balance	The initial balance to set for the store credit.	Any decimal value.

## StoreCreditTransactions

Corresponding REST API Resource: [commerce/customer/credits/transactions](#)

This sheet allows you to assign store credit transactions based on a store credit code.

Column Name	Description	Valid Values
Code	User-defined code that uniquely identifies the store credit.	Any string that matches an existing store credit code.
Comments	An optional field that allows users to make notes about the store credit transaction.	Any string.
Id	The unique identifier for the transaction.	This is a read-only, system-generated ID. If you are creating an import of a store credit transaction, leave this field blank.
Impact Amount	The amount to subtract or add from the store credit balance.	Any decimal value. Use a negative value for debits, and a positive value if adding more credit.
Transaction Type	Identifies the transaction as either a debit or a credit to the store credit balance.	Either <code>Debit</code> or <code>Credit</code> .

## Customer Segments

Corresponding REST API Resource: [commerce/customer/segments](#)

This sheet defines customer segments within Kibo eCommerce.

Column Name	Description	Valid Values
Customer Segment Id	The unique identifier for the customer segment.	This is a read-only, system-generated ID. If you are creating an import of a customer segment, leave this field blank.
Code	User-defined code that uniquely identifies the customer segment.	Any string that matches an existing customer segment code.
Name	The name of the customer segment as it appears in Admin.	Any string.
Description	An optional description of the customer segment.	Any string.

## Best Practices

### Use the Kibo eCommerce API to Export > 50,000 Records

For optimal performance, Kibo eCommerce recommends only using the import/export tool if you are working with fewer than 50,000 customer records. If you need to export or import more than 50,000 records, refer to the samples below for more information about using the Kibo eCommerce API to import your customer records.

## Customer Import Code Samples

The following sample code demonstrates how to import customers using the Kibo eCommerce API. These samples were implemented using the [Kibo eCommerce .NET SDK](#), but the general implementation is the same across all the Kibo eCommerce SDKs.

### Required Behaviors

Any application that implements code to import or update customers must have the following [behaviors](#) enabled:

- Customer Create
- Customer Read



- Customer Update

## Import Customers

The following code sample organizes various calls to import customers to Kibo eCommerce:

```

private System.Collections.Concurrent.ConcurrentDictionary ImportCustomers(IEnumerable accountAuthCollection)
{
    var accountResource = new Mozu.Api.Resources.Commerce.Customer.CustomerAccountResource(_apiContext);
    var contactResource = new Mozu.Api.Resources.Commerce.Customer.Accounts.CustomerContactResource(_apiContext);

    var accountExternalIds = new System.Collections.Concurrent.ConcurrentDictionary();

    Parallel.ForEach(accountAuthCollection, new System.Threading.Tasks.ParallelOptions() { MaxDegreeOfParallelism = 1 }, accountAuthInfo =>
    {
        var filter = string.Format("ExternalId eq '{0}'", accountAuthInfo.Account.ExternalId);

        var importedResultStatus = string.Empty;

        Mozu.Api.Contracts.Customer.CustomerAccount existingAcct = null;

        var existingAcctCollection = accountResource.GetAccountsAsync(filter: filter).Result;

        if (existingAcctCollection.Items.Count == 0)
        {
            try
            {
                existingAcct = accountResource.AddAccountAsync(accountAuthInfo.Account, "Id, ExternalId").Result;
                importedResultStatus = string.Format("Added-New-Account-Id:{0}", existingAcct.Id);

                var existingAcctAuthTicket = accountResource.AddLoginToExistingCustomerAsync(new Mozu.Api.Contracts.Customer.CustomerLoginInfo()
                {
                    EmailAddress = accountAuthInfo.Account.EmailAddress,
                    IsImport = accountAuthInfo.IsImport,
                    Password = accountAuthInfo.Password,
                    Username = accountAuthInfo.Account.UserName,
                }, existingAcct.Id).Result;

                foreach (var contact in accountAuthInfo.Account.Contacts)
                {
                    var existingContact = contactResource.AddAccountContactAsync(contact, existingAccount.Id, "Id").Result;
                }
            }
            catch (Exception ex)
            {
                importedResultStatus = "Error-Encountered";
            }

            accountExternalIds.TryAdd(existingAcct.ExternalId, importedResultStatus);
        }
    });

    return accountExternalIds;
}

```

## Add a Customer Account

The following code sample is a unit test that demonstrates the uses of various customer import methods:

```
[TestMethod]
public void Add_Customer_Accounts_Test()
{
    var tblAccounts = GetAccountTestData();
    var tblContacts = GetContactTestData();
    var accountAndAuths = new List();

    foreach(System.Data.DataRow drAcct in tblAccounts.Rows)
    {
        var customerLoginInfo = MapCustomerLoginInfo(drAcct);
        var account = MapCustomerAccount(drAcct);

        System.Data.DataRow[] drContacts = tblContacts.Select(string.Format("Id = '{0}'", drAcct["I
d"]).ToString());

        account.Contacts = new List();
        foreach(System.Data.DataRow drContact in drContacts)
        {
            var contact = MapCustomerContact(drContact);
            account.Contacts.Add(contact);
        }

        accountAndAuths.Add(new Mozu.Api.Contracts.Customer.CustomerAccountAndAuthInfo()
        {
            Account = account,
            IsImport = Convert.ToBoolean(customerLoginInfo.IsImport),
            Password = customerLoginInfo.Password
        });
    }

    var importedAcctIds = ImportCustomers(accountAndAuths);
}
```

## Add Account Login

The following code sample maps customer account login information to a customer object in Kibo eCommerce:

```
private Mozu.Api.Contracts.Customer.CustomerLoginInfo MapCustomerLoginInfo(System.Data.DataRow drAccount)
{
    var mappedAccountAndAuthInfo = new Mozu.Api.Contracts.Customer.CustomerLoginInfo()
    {
        IsImport = true,
        EmailAddress = drAccount["Email"].ToString(),
        Password = drAccount["Password"].ToString(),
        Username = drAccount["Username"].ToString(),
    };

    return mappedAccountAndAuthInfo;
}
```

## Get Account Test Data

The following code sample retrieves test data for customer accounts that you can then map to a customer object in Kibo eCommerce:

```
private System.Data.DataTable GetAccountTestData()
{
    var tblAccount = new System.Data.DataTable();
    tblAccount.Columns.Add("Id");
    tblAccount.Columns.Add("AccountID");
    tblAccount.Columns.Add("UserName");
    tblAccount.Columns.Add("Email");
    tblAccount.Columns.Add("Password");
    tblAccount.Columns.Add("FirstName");
    tblAccount.Columns.Add("LastNameOrSurname");
    tblAccount.Columns.Add("CompanyOrOrganization");
    tblAccount.Columns.Add("TaxExempt");
    tblAccount.Columns.Add("TaxId");
    tblAccount.Columns.Add("AcceptsMarketing");
    tblAccount.Columns.Add("LocaleCode");
    tblAccount.Columns.Add("UserRole");
    tblAccount.Columns.Add("IsActive");
    tblAccount.Columns.Add("IsAnonymous");
    tblAccount.Columns.Add("Total Order Amount");
    tblAccount.Columns.Add("Total Orders");
    tblAccount.Columns.Add("Last Order Date");
    tblAccount.Columns.Add("Total Contacts");
    tblAccount.Columns.Add("LastModifiedDate");

    System.Data.DataRow newRow = tblAccount.NewRow();
    newRow["Id"] = "101"; //externalId
    newRow["UserName"] = "vip_shopper";
    newRow["Email"] = "vip_shopper@mozu.com";
    newRow["Password"] = "IBuyOnline2015";
    newRow["FirstName"] = "Jon";
    newRow["LastNameOrSurname"] = "Smithe";
    newRow["CompanyOrOrganization"] = "Main Corp";
    newRow["TaxExempt"] = false;
    newRow["AcceptsMarketing"] = true;
    newRow["LocaleCode"] = "en-US";
    newRow["IsActive"] = true;
    newRow["IsAnonymous"] = false;
    tblAccount.Rows.Add(newRow);

    return tblAccount;
}
```

## Map Account Data

The following code sample maps customer account data to a customer object:

```
private Mozu.Api.Contracts.Customer.CustomerAccount MapCustomerAccount(System.Data.DataRow drAccount)
{
    var mappedAccount = new Mozu.Api.Contracts.Customer.CustomerAccount()
    {
        ExternalId = drAccount["Id"].ToString(),
        EmailAddress = drAccount["Email"].ToString(),
        FirstName = drAccount["FirstName"].ToString(),
        LastName = drAccount["LastNameOrSurname"].ToString(),
        CompanyOrOrganization = drAccount["CompanyOrOrganization"].ToString(),
        TaxExempt = Convert.ToBoolean(drAccount["TaxExempt"]),
        TaxId = drAccount["TaxId"].ToString(),
        AcceptsMarketing = Convert.ToBoolean(drAccount["AcceptsMarketing"]),
        LocaleCode = drAccount["LocaleCode"].ToString(),
        IsActive = Convert.ToBoolean(drAccount["IsActive"]),
        IsAnonymous = Convert.ToBoolean(drAccount["IsAnonymous"]),
        UserName = drAccount["UserName"].ToString(),
    };

    return mappedAccount;
}
```

## Get Contact Test Data

The following code sample retrieves test data for customer contacts that you can then map to a customer object in Kibo eCommerce:

```

private System.Data.DataTable GetContactTestData()
{
    var tblContact = new System.Data.DataTable();
    tblContact.Columns.Add("Id");
    tblContact.Columns.Add("Type");
    tblContact.Columns.Add("IsPrimary");
    tblContact.Columns.Add("CompanyOrOrganization");
    tblContact.Columns.Add("FirstName");
    tblContact.Columns.Add("MiddleNameorInitial");
    tblContact.Columns.Add("LastNameOrSurname");
    tblContact.Columns.Add("Email");
    tblContact.Columns.Add("FaxNumber");
    tblContact.Columns.Add("HomePhone");
    tblContact.Columns.Add("MobilePhone");
    tblContact.Columns.Add("WorkPhone");
    tblContact.Columns.Add("AddressType");
    tblContact.Columns.Add("Address1");
    tblContact.Columns.Add("Address2");
    tblContact.Columns.Add("CityOrTown");
    tblContact.Columns.Add("StateOrProvince");
    tblContact.Columns.Add("PostalOrZipCode");
    tblContact.Columns.Add("CountryCode");

    System.Data.DataRow newRow = tblContact.NewRow();
    newRow["Id"] = "101"; //externalId
    newRow["Type"] = "Billing";
    newRow["IsPrimary"] = true;
    newRow["CompanyOrOrganization"] = "Main Corp";
    newRow["FirstName"] = "Jon";
    newRow["LastNameOrSurname"] = "Smithe";
    newRow["Email"] = "vip_shopper@mozu.com";
    newRow["WorkPhone"] = "555-555-5555";
    newRow["AddressType"] = "Commercial";
    newRow["Address1"] = "100 Main Corp Way";
    newRow["Address2"] = "Building 9";
    newRow["CityOrTown"] = "New York";
    newRow["StateOrProvince"] = "NY";
    newRow["PostalOrZipCode"] = "10026";
    newRow["CountryCode"] = "US";
    tblContact.Rows.Add(newRow);
    return tblContact;
}

```

## Map Contact Data

The following code sample maps customer contact data to a customer object:

```

private Mozu.Api.Contracts.Customer.CustomerContact MapCustomerContact(System.Data.DataR
ow drContact)
{
    var mappedContact = new Mozu.Api.Contracts.Customer.CustomerContact()
    {
        CompanyOrOrganization = drContact["CompanyOrOrganization"].ToString(),
        Email = drContact["Email"].ToString(),
        FaxNumber = drContact["FaxNumber"].ToString(),
        FirstName = drContact["FirstName"].ToString(),
        LastNameOrSurname = drContact["LastNameOrSurname"].ToString(),
        MiddleNameOrInitial = drContact["MiddleNameOrInitial"].ToString(),
        Types = new List()
        {
            new Mozu.Api.Contracts.Customer.ContactType()
            {
                Name = drContact["Type"].ToString(),
                IsPrimary = Convert.ToBoolean(drContact["IsPrimary"]),
            }
        },
        Address = new Mozu.Api.Contracts.Core.Address()
        {
            Address1 = drContact["Address1"].ToString(),
            Address2 = drContact["Address2"].ToString(),
            AddressType = drContact["AddressType"].ToString(),
            CityOrTown = drContact["CityOrTown"].ToString(),
            StateOrProvince = drContact["StateOrProvince"].ToString(),
            PostalOrZipCode = drContact["PostalOrZipCode"].ToString(),
            CountryCode = drContact["CountryCode"].ToString()
        },
        PhoneNumbers = new Mozu.Api.Contracts.Core.Phone()
        {
            Home = drContact["HomePhone"].ToString(),
            Mobile = drContact["MobilePhone"].ToString(),
            Work = drContact["WorkPhone"].ToString(),
        }
    };
    return mappedContact;
}

```