INTUIKO
COMMERCE EVERYWHERE

Intuiko for Connected Store

Magento Connector Documentation

V 1411.13

INTUIKO
COMMERCE EVERYWHERE

contact@intuiko.com
(+33) 1.77.530.679

www.intuiko.com
50 rue de Paradis - 75010 Paris

E-COMMERCE AWARDS
PARIS 2013

Contents

This section of the documentation will provide you with information about the Magento Connector developed for Intuiko for Connected Store.

Our Connector supports the following versions of Magento:

- **Community Edition : 1.8.1.0** & **1.9.0**
- **Enterprise Edition : 1.13**

All screenshots featured in the present documentation are taken from a Magento 1.8.1.0 Community Edition.

Also, please keep in mind that all the following information applies to an "Out Of The Box" Magento. Any specific development impacting the relations between Magento and ICS will have to be the subject of a connector override.

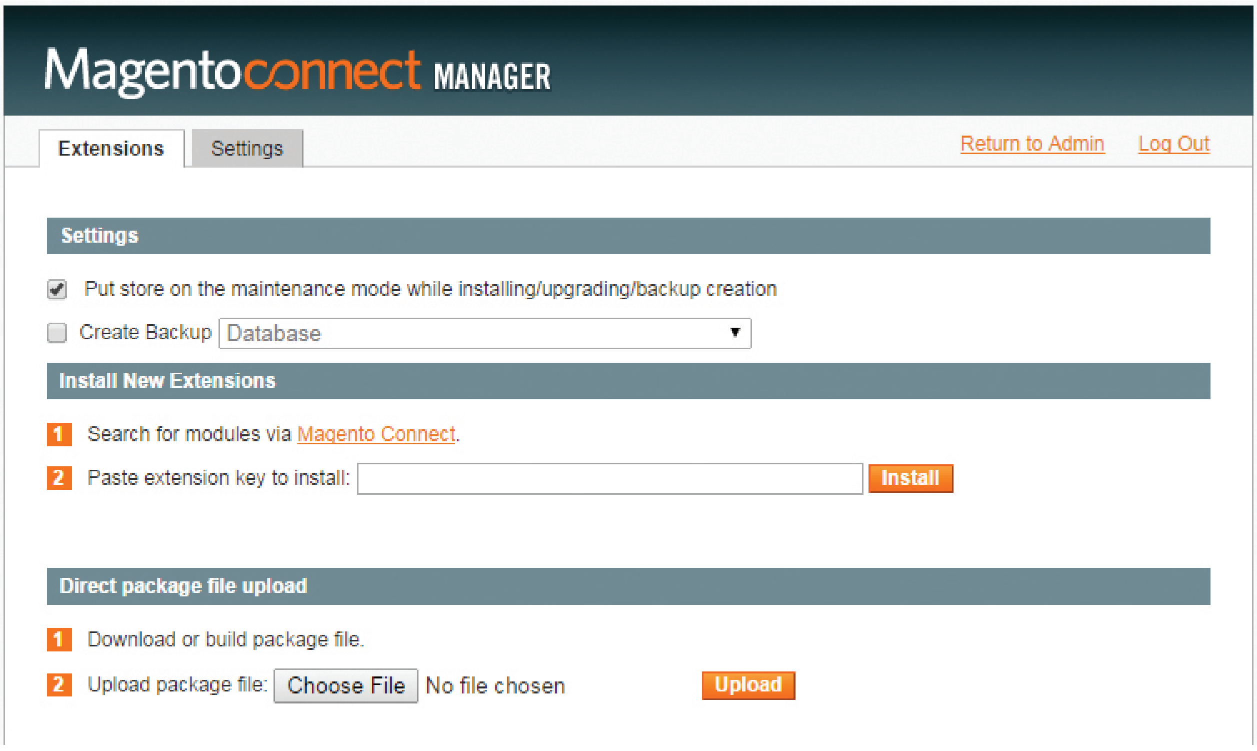
If you find yourself in need of more information about the conceptual model behind ICS and/or our public API, you should have a look at our Documentation Introduction and/or our API Documentation.

Setup & Configuration

INSTALL

To install the connector, you need the latest version of this package: **ics-magento-connector-Vxx.tgz** (where the xx stands for the version number).

Log into your Magento Admin panel and go to System >> Magento Connect >> Magento Connect Manager:


Magentoconnect MANAGER

Extensions | Settings | Return to Admin | Log Out

Settings

[✓] Put store on the maintenance mode while installing/upgrading/backup creation

[ ] Create Backup: Database

Install New Extensions

1 Search for modules via Magento Connect.

2 Paste extension key to install: Install

Direct package file upload

1 Download or build package file.

2 Upload package file: Choose File No file chosen Upload

In the Direct package file upload part, click the Choose File button and select the package **ics-magento-connector-Vxx.tgz**. Please check that your package has the **.tgz** extension before selecting it.

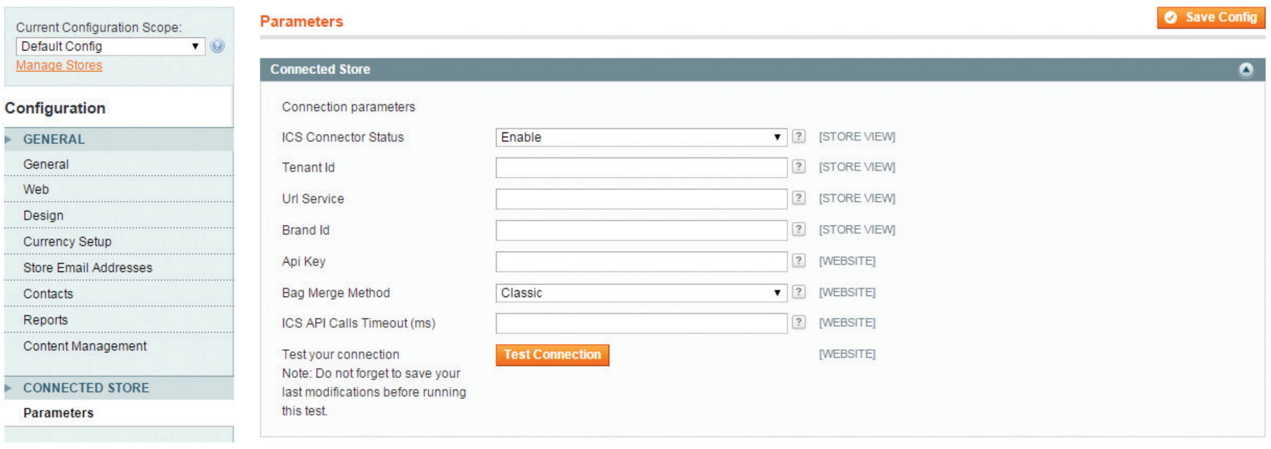
Click the Upload button and refresh the page once the upload is completed.

If the plan comes together—and everyone just loves it when it does—, the module will now appear in the list of packages featured in the Manage Existing Extensions part.

CONFIGURATION

In order to configure the ICS Magento Connector, you first need to log out of your Magento Admin panel and back into it.

Go to System >> Configuration >> Connected Store >> Parameters and fill in all the fields under Connection Parameters:



Parameters

Connection parameters

ICS Connector Status	Enable	[STORE VIEW]
Tenant Id		[STORE VIEW]
Url Service		[STORE VIEW]
Brand Id		[STORE VIEW]
Api Key		[WEBSITE]
Bag Merge Method	Classic	[WEBSITE]
ICS API Calls Timeout (ms)		[WEBSITE]
Test your connection Note: Do not forget to save your last modifications before running this test.	Test Connection	[WEBSITE]

Field	Description
ICS Connector Status	To activate or deactivate the connector (see page 5 of this documentation).
Tenant Id	Your ICS identifier as our customer.
Url Service	The URL of the ICS API (HTTP or HTTPS protocol).
Brand Id	The ICS identifier of your brand.
Api Key	Your key, which ensures the security of all data exchanges with the API.
Bag Merge Method	The method used when two existing bags are merged into a new one (see page 15 of the API Documentation).
ICS API Calls Timeout (ms)	The time limit, in milliseconds, for the calls made to the API (when a HTTP call falls into time-out, the user's calls are not made any more during their session).

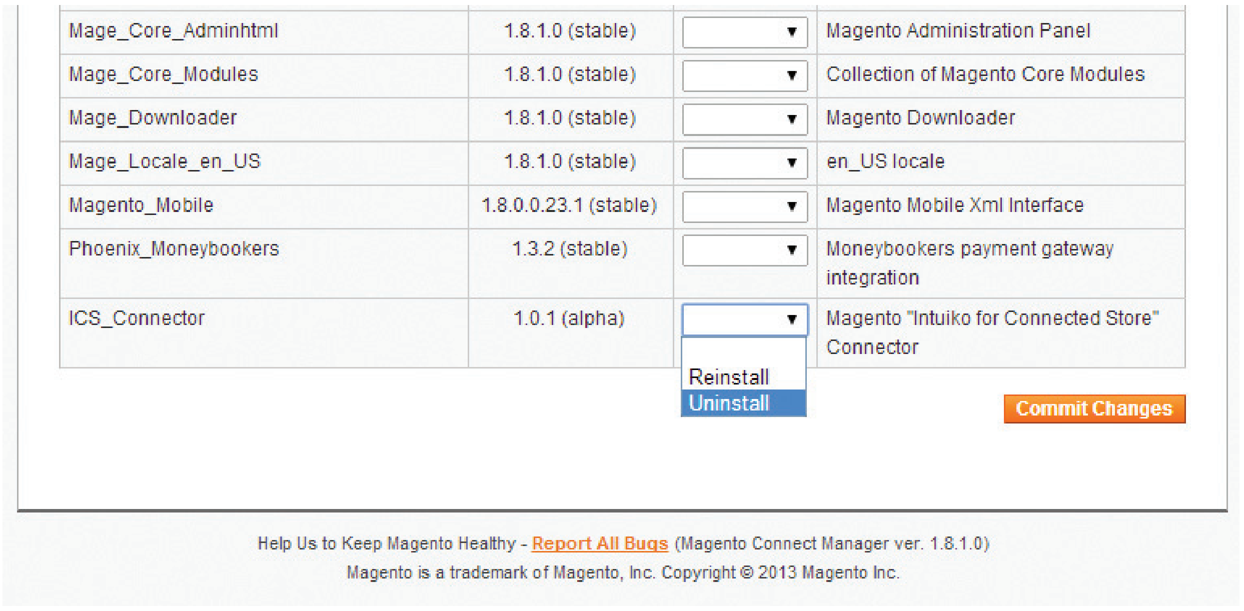
You can validate your connection to the ICS API by clicking the Test Connection button. However, please save the modifications you made to the configuration **before** testing the connection.

A returned success message validates the connection to the API.

UNINSTALL

To uninstall the ICS Magento Connector, log into your Magento Admin panel and go to System >> Magento Connect >> Magento Connect Manager.

Scroll down the list of packages featured in the Manage Existing Extensions part until you find **ICS_Connector**. In the drop-down menu, select Uninstall and click the Commit Changes button:



Mage_Core_Adminhtml	1.8.1.0 (stable)		Magento Administration Panel
Mage_Core_Modules	1.8.1.0 (stable)		Collection of Magento Core Modules
Mage_Downloader	1.8.1.0 (stable)		Magento Downloader
Mage_Locale_en_US	1.8.1.0 (stable)		en_US locale
Magento_Mobile	1.8.0.0.23.1 (stable)		Magento Mobile Xml Interface
Phoenix_Moneybookers	1.3.2 (stable)		Moneybookers payment gateway integration
ICS_Connector	1.0.1 (alpha)		Magento "Intuiko for Connected Store" Connector

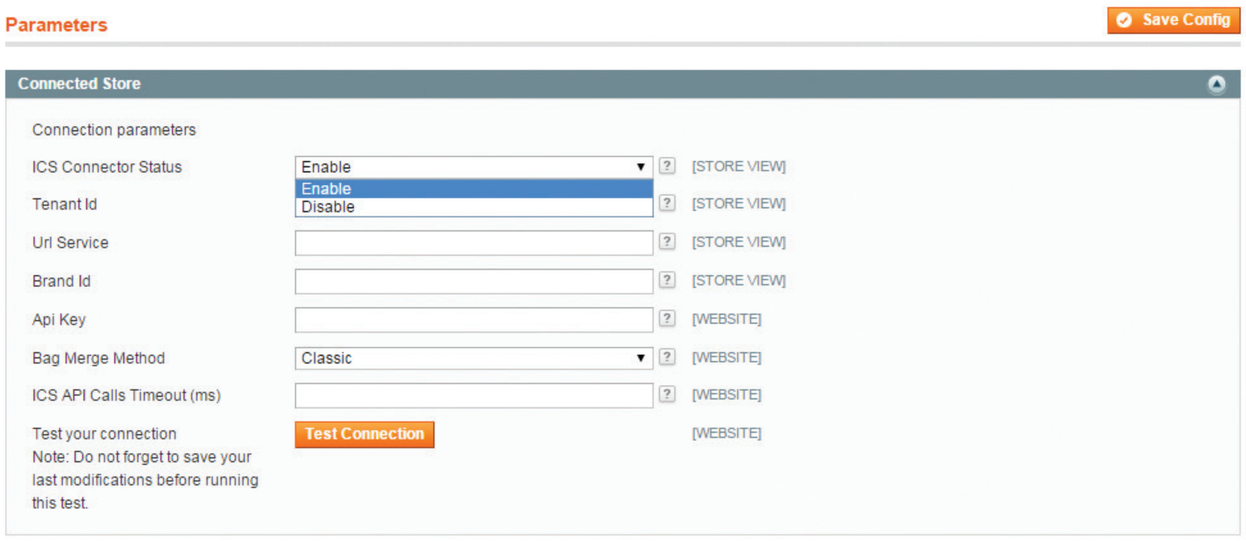
Help Us to Keep Magento Healthy - Report All Bugs (Magento Connect Manager ver. 1.8.1.0)
Magento is a trademark of Magento, Inc. Copyright © 2013 Magento Inc.

ENABLE OR DISABLE

By default, the ICS Magento Connector is deactivated after its installation is completed. If the connector is not active, there is no communication whatsoever between Magento and ICS.

To **activate** the connector, log into your Magento Admin panel and go to System >> Configuration >> Connected Store >> Parameters. In the drop-down menu next to ICS Connector Status, select Enable and click the Save Config button.

To **deactivate** the connector, do exactly what is said in the paragraph above, with the exception of selecting Disable in the drop-down menu next to ICS Connector Status.



Connection parameters

ICS Connector Status	Enable	[STORE VIEW]
Tenant Id		[STORE VIEW]
Url Service		[STORE VIEW]
Brand Id		[STORE VIEW]
Api Key		[WEBSITE]
Bag Merge Method	Classic	[WEBSITE]
ICS API Calls Timeout (ms)		[WEBSITE]
Test your connection Note: Do not forget to save your last modifications before running this test.	Test Connection	[WEBSITE]

Functional scope

In this section of our Magento Connector documentation, you will find more about what it actually *does* and how the API and Magento interact with each other. It is divided into three parts:

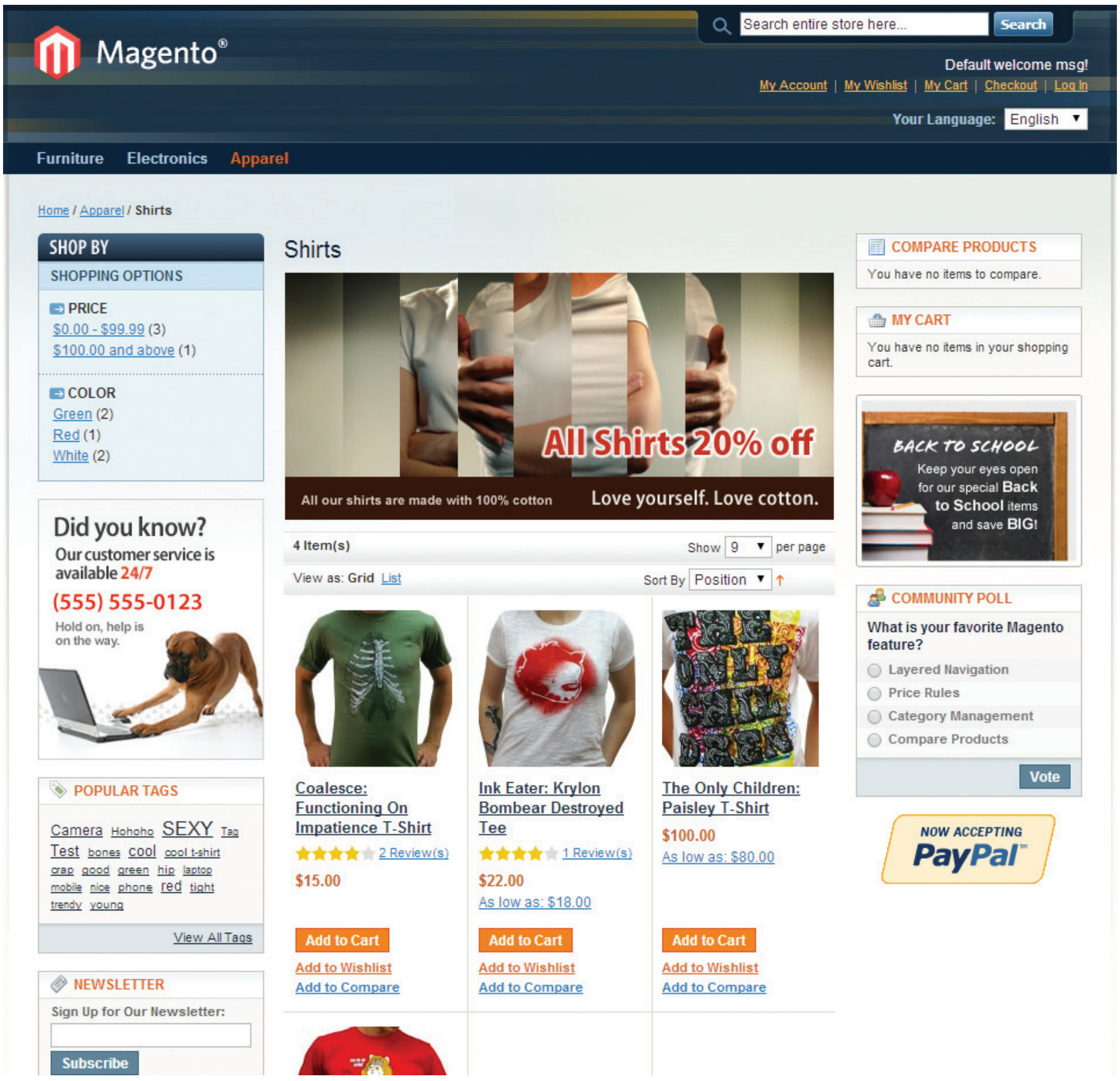
- Catalogue
- Tunnel
- Client

Let's get to it, shall we?

CATALOGUE

Product List / Search Results

On this page, your customer gets the results of a specific search or see a list of products which belongs to a specific category. They have the possibility to add a product to their wishlist or cart—if they do not have to customize it—or to the product comparison page. They can also go to the product details page.


Magento®
Search entire store here...
Search
Default welcome msg!
My Account | My Wishlist | My Cart | Checkout | Log In
Your Language: English
Furniture
Electronics
Apparel
Home / Apparel / Shirts
SHOP BY
SHOPPING OPTIONS
PRICE
$0.00 - $99.99 (3)
$100.00 and above (1)
COLOR
Green (2)
Red (1)
White (2)
Did you know?
Our customer service is available 24/7
(555) 555-0123
Hold on, help is on the way.
POPULAR TAGS
Camera Hohoho SEXY Tag Test bones cool cool t-shirt crap good green hip laptop mobile nice phone red tight trendy young
View All Tags
NEWSLETTER
Sign Up for Our Newsletter:
Subscribe
Shirts
All Shirts 20% off
All our shirts are made with 100% cotton
Love yourself. Love cotton.
4 Item(s)
Show 9 per page
View as: Grid List
Sort By Position
Coalesce: Functioning On Impatience T-Shirt
2 Review(s)
$15.00
Ink Eater: Krylon Bombear Destroyed Tee
1 Review(s)
$22.00
As low as: $18.00
The Only Children: Paisley T-Shirt
$100.00
As low as: $80.00
Add to Cart
Add to Wishlist
Add to Compare
COMPARE PRODUCTS
You have no items to compare.
MY CART
You have no items in your shopping cart.
BACK TO SCHOOL
Keep your eyes open for our special Back to School items and save BIG!
COMMUNITY POLL
What is your favorite Magento feature?
Layered Navigation
Price Rules
Category Management
Compare Products
Vote
NOW ACCEPTING PayPal

ADDING A PRODUCT TO THE CART:

- If the product has discriminants, the customer is redirected to the product details page.
- If the product has no discriminant:
  1. The customer adds a single quantity of the product to their cart.
  2. Magento sends a `controller_action_predispatch` event before modifying its cart.
  3. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS cart (stocked by the controller):

     ```
     GET   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
     ```

  4. This call allows the connector to check that the cart information are up to date (e.g. the ICS permanent cart may have been modified by another application). If need be, the connector refreshes the Magento cart.
  5. Magento adds the product to its cart and sends a `checkout_cart_save_after event`.
  6. The connector picks up this event and recovers the current Magento cart.
  7. The connector saves this recovered cart on ICS by making a REST call:

     ```
     POST  https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
     ```

ADDING A PRODUCT TO THE WISHLIST

1. The customer adds a single quantity of the product to their wishlist.
2. Magento sends a `controller_action_predispatch` event before modifying its wishlist.
3. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS wishlist (stocked by the controller):

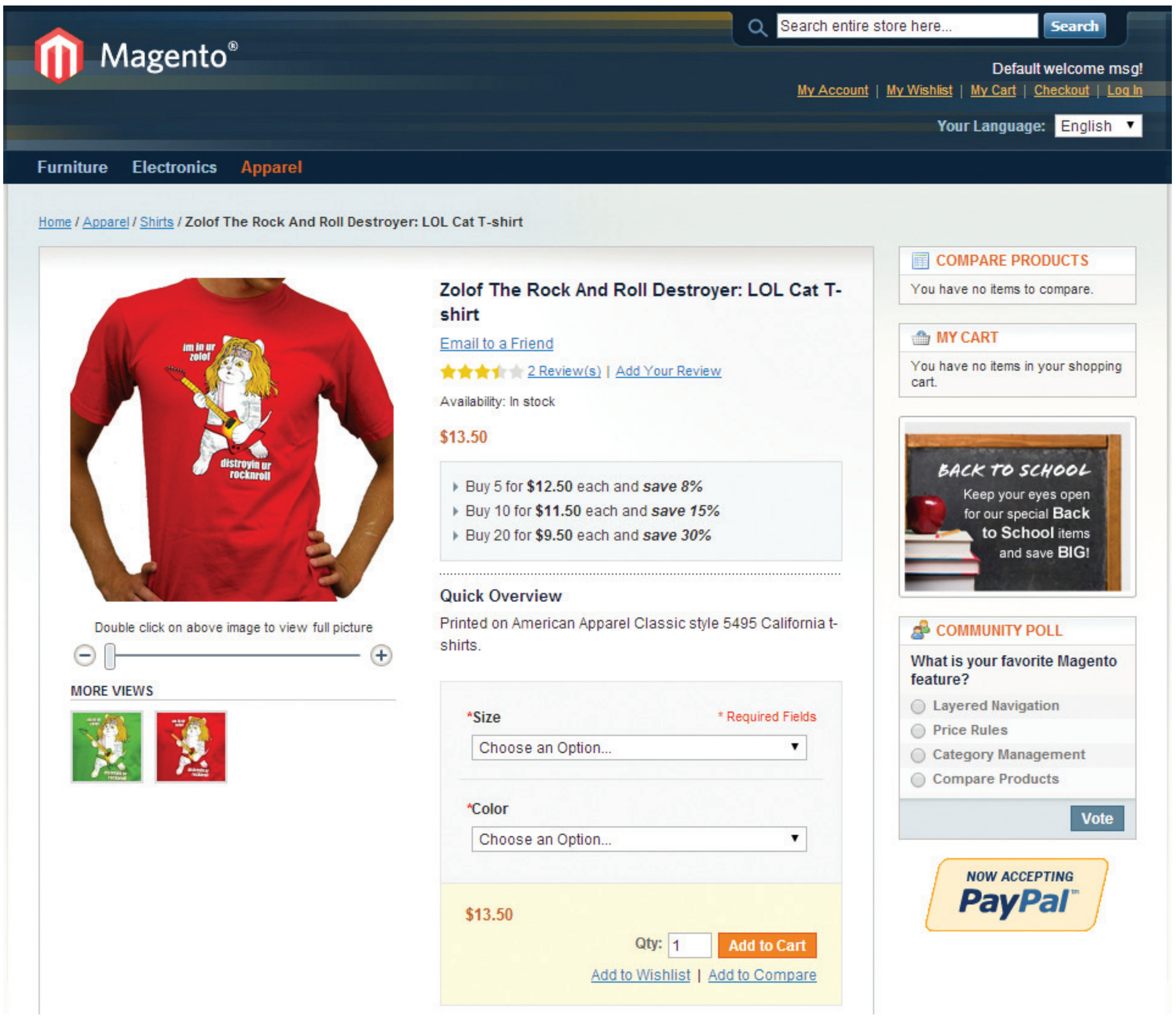
   ```
   GET   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
   ```

4. This call allows the connector to check that the whishlist information are up to date (e.g. the ICS permanent wishlist may have been modified by another application). If need be, the connector refreshes the Magento wishlist.
5. Magento adds the product to its wishlist and sends a `wishlist_items_renewed` event.
6. The connector picks up this event and recovers the current Magento wishlist.
7. The connector saves this recovered wishlist on ICS by making a REST call:

   ```
   POST  https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
   ```

Product details page

This page gives the detailed information of the product to your customer. They have the possibility to customize their product—if it has one or several discriminants—and choose the quantity they want to buy. They can add these quantities to their wishlist or cart or to the product comparison page.



Magento®

Search entire store here... Search

Default welcome msg!

My Account | My Wishlist | My Cart | Checkout | Log In

Your Language: English

Furniture Electronics Apparel

Home / Apparel / Shirts / Zolof The Rock And Roll Destroyer: LOL Cat T-shirt

Zolof The Rock And Roll Destroyer: LOL Cat T-shirt

Email to a Friend

2 Review(s) | Add Your Review

Availability: In stock

$13.50

- Buy 5 for **$12.50** each and ***save 8%***
- Buy 10 for **$11.50** each and ***save 15%***
- Buy 20 for **$9.50** each and ***save 30%***

Quick Overview

Printed on American Apparel Classic style 5495 California t-shirts.

Double click on above image to view full picture

MORE VIEWS

*Size * Required Fields

Choose an Option...

*Color

Choose an Option...

$13.50

Qty: 1 Add to Cart

Add to Wishlist | Add to Compare

COMPARE PRODUCTS

You have no items to compare.

MY CART

You have no items in your shopping cart.

BACK TO SCHOOL Keep your eyes open for our special Back to School items and save BIG!

COMMUNITY POLL

What is your favorite Magento feature?

- Layered Navigation
- Price Rules
- Category Management
- Compare Products

Vote

NOW ACCEPTING PayPal™

ADDING A PRODUCT TO THE CART

1. The customer chooses a quantity and fills in the discriminant fields when required.
2. The customer then clicks the Add to cart button.
3. Magento sends a `controller_action_predispatch` event before modifying its cart.
4. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS cart (stocked by the controller):

```
GET    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

5. This call allows the connector to check that the cart information are up to date (e.g. the ICS permanent cart may have been modified by another application). If need be, the connector refreshes the Magento cart.
6. Magento adds the product (with the information provided by the customer) to its cart and sends a `checkout_cart_save_after` event.
7. The connector picks up this event and recovers the current Magento cart.

8. The connector saves this recovered cart on ICS by making a REST call:

```
POST   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

ADDING A PRODUCT TO THE WHISHLIST

1. The customer chooses a quantity and fills in the discriminant fields when required.
2. The customer then clicks the Add to wishlist button.
3. Magento sends a `controller_action_predispatch` event before modifying its wishlist.
4. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS wishlist (stocked by the controller):

```
GET    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

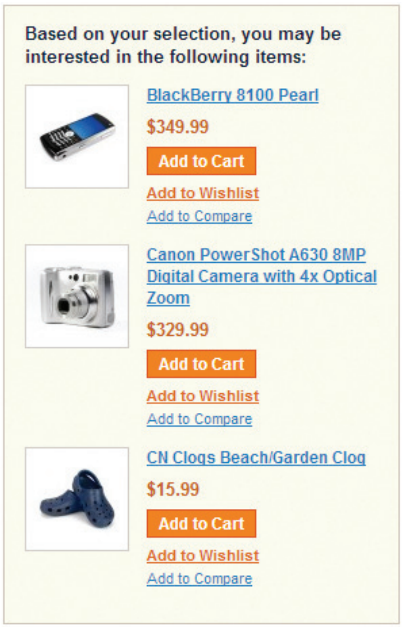
5. This call allows the connector to check that the wishlist information are up to date (e.g. the ICS permanent wishlist may have been modified by another application). If need be, the connector refreshes the Magento wishlist.
6. Magento adds the product (with the information provided by the customer) to its wishlist and sends a `wishlist_items_renewed` event.
7. The connector picks up this event and recovers the current Magento wishlist.
8. The connector saves this recovered wishlist on ICS by making a REST call:

```
POST   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

Cross-selling

With the cross-selling spots of Magento, your customer have the possibility to add a product to their wishlist or cart—if they do not have to customize it—or to the product comparison page. They can also go to the product details page.

The ICS mechanisms at play here are the same as explained in the Product details page section, pages 8 and 9 (oddly enough, they happen to be this very page and the previous one).


Based on your selection, you may be interested in the following items:

BlackBerry 8100 Pearl
$349.99
Add to Cart
Add to Wishlist
Add to Compare

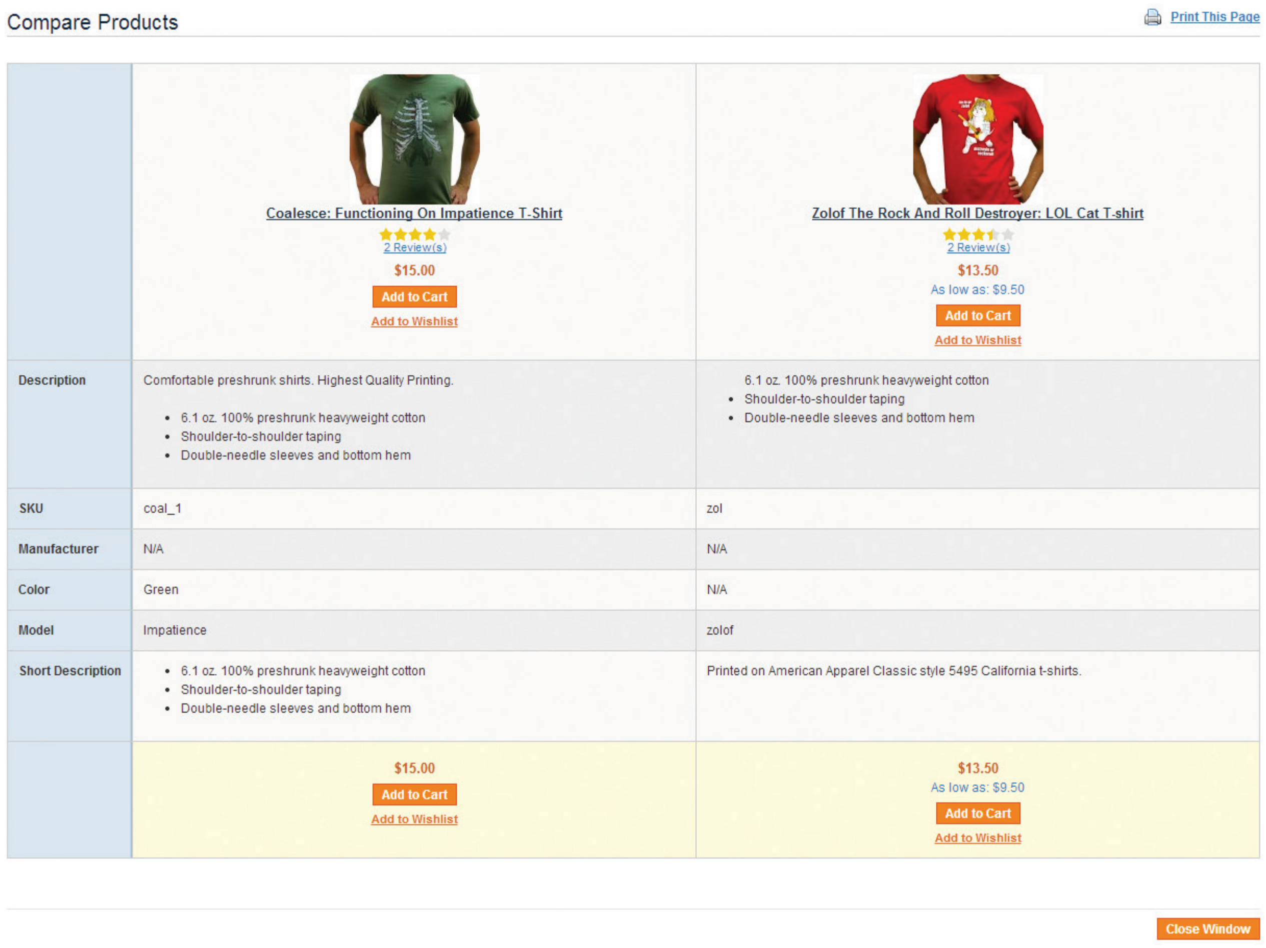
Canon PowerShot A630 8MP Digital Camera with 4x Optical Zoom
$329.99
Add to Cart
Add to Wishlist
Add to Compare

CN Clogs Beach/Garden Clog
$15.99
Add to Cart
Add to Wishlist
Add to Compare

Product comparison page

On this page, your customer have the possibility to compare the details of the various products they added to it If they want to do so, they may add one product at a time to their wishlist or cart.

The ICS mechanisms at play here are the same as explained in the Product details page section, pages 8 and 9.



Compare Products

Print This Page

	Coalesce: Functioning On Impatience T-Shirt 2 Review(s) $15.00 Add to Cart Add to Wishlist	Zolof The Rock And Roll Destroyer: LOL Cat T-shirt 2 Review(s) $13.50 As low as: $9.50 Add to Cart Add to Wishlist
Description	Comfortable preshrunk shirts. Highest Quality Printing. • 6.1 oz. 100% preshrunk heavyweight cotton • Shoulder-to-shoulder taping • Double-needle sleeves and bottom hem	6.1 oz. 100% preshrunk heavyweight cotton • Shoulder-to-shoulder taping • Double-needle sleeves and bottom hem
SKU	coal_1	zol
Manufacturer	N/A	N/A
Color	Green	N/A
Model	Impatience	zolof
Short Description	• 6.1 oz. 100% preshrunk heavyweight cotton • Shoulder-to-shoulder taping • Double-needle sleeves and bottom hem	Printed on American Apparel Classic style 5495 California t-shirts.
	$15.00 Add to Cart Add to Wishlist	$13.50 As low as: $9.50 Add to Cart Add to Wishlist

Close Window

Compatible product types

At this time, the ICS Magento connector and its services are compatible with all product types provided by Magento, whether they include options or not. Those types are:

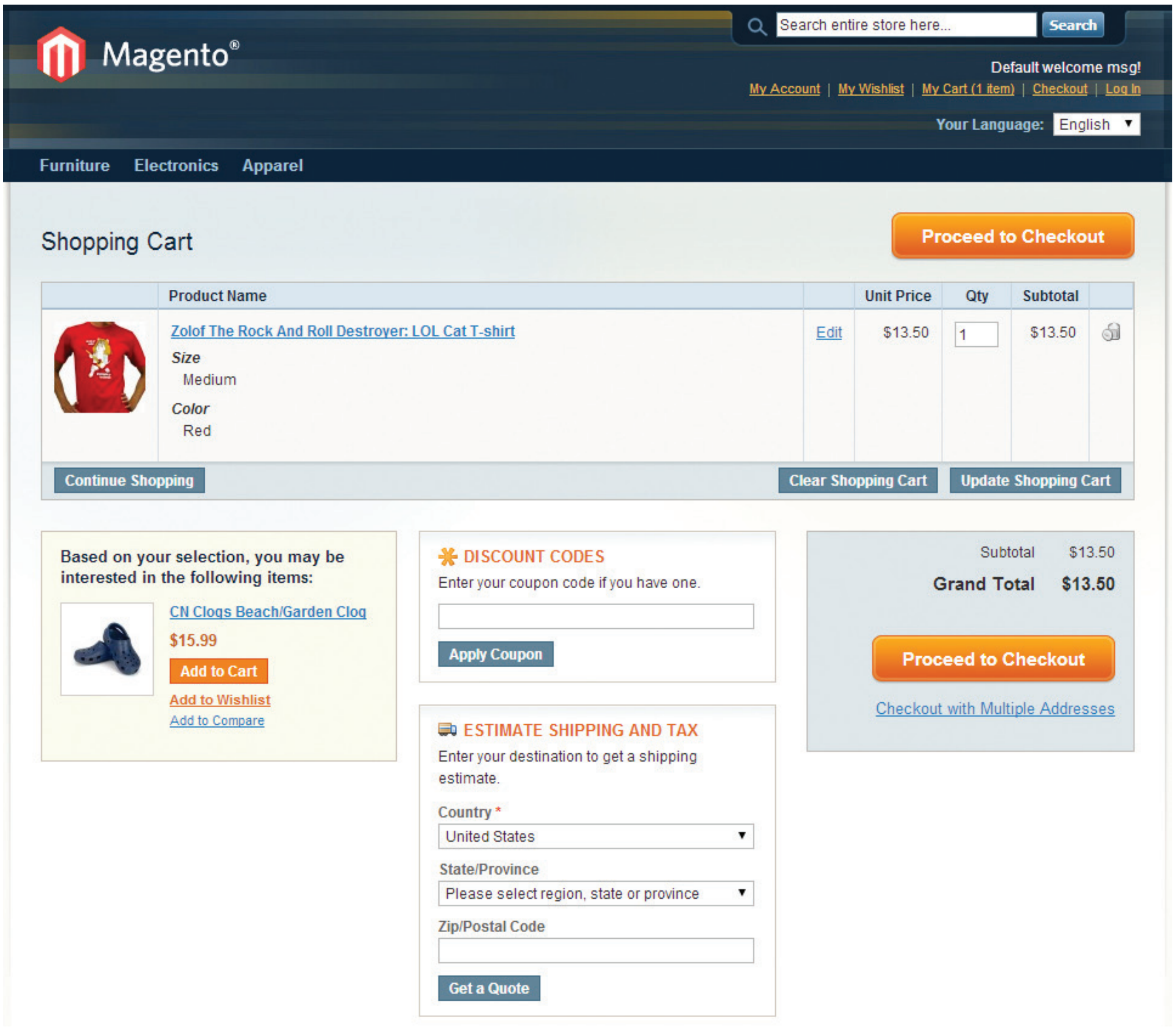
- simple
- configurable
- groups
- bundle
- downloadable
- virtual

However, **omni-channel functionalities** are **only compatible** with the **simple** type—whether it has discriminants or not—but without any option.

Cart page

This page sums the contents of your customer's current cart up. From it, they have the possibility to:

- modify or customize a product;
- delete a product;
- add a product to their wishlist;
- empty their cart;
- modify quantities on various lines at one time;
- consult a product details page;
- add a coupon;
- modify or delete a current coupon;
- proceed to a checkout order.



Shopping Cart

	Product Name		Unit Price	Qty	Subtotal	
	Zolof The Rock And Roll Destroyer: LOL Cat T-shirt *Size* Medium *Color* Red	Edit	$13.50	1	$13.50	

Based on your selection, you may be interested in the following items:

CN Clogs Beach/Garden Clog
$15.99

DISCOUNT CODES
Enter your coupon code if you have one.

ESTIMATE SHIPPING AND TAX
Enter your destination to get a shipping estimate.
Country *
United States
State/Province
Please select region, state or province
Zip/Postal Code

Subtotal	$13.50
Grand Total	**$13.50**

CONSULTING THE CART

1. The customer requests the display of their cart.
2. Magento sends a `controller_action_predispatch` event.

3. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS cart (stocked by the controller):

```
GET    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

4. This call allows the connector to check that the cart information are up to date (e.g. the ICS permanent cart may have been modified by another application). If need be, the connector refreshes the Magento cart.

MODIFYING THE CART

1. The customer modifies something (e.g. modifying a quantity, deleting a product, adding, modifying or deleting a coupon) and clicks the button which validates this modification.
2. Magento sends a `controller_action_predispatch` event before modifying its cart.
3. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS cart (stocked by the controller):

```
GET    https://api-ics.intuiko.com/api/enants/tenantId/brands/brandId/bags/
```

4. This call allows the connector to check that the cart information are up to date (e.g. the ICS permanent cart may have been modified by another application). If need be, the connector refreshes the Magento cart.
5. Magento saves the modifications and sends a `checkout_cart_save_after` event.
6. The connector picks up this event and recovers the current Magento cart.
7. The connector saves this recovered cart on ICS by making a REST call:

```
POST   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

MOVING A PRODUCT TO THE WISHLIST

1. The customer clicks the Move to wishlist button.
2. Magento sends a `controller_action_predispatch` event before modifying its cart.
3. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS cart (stocked by the controller):

```
GET    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

4. This call allows the connector to check that the cart information are up to date (e.g. the ICS permanent cart may have been modified by another application). If need be, the connector refreshes the Magento cart.
5. Magento modifies its cart by deleting the product and sends a `checkout_cart_save_after` event.
6. The connector picks up this event and recovers the current Magento cart.
7. The connector saves this recovered cart on ICS by making a REST call:

```
POST   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

8. Magento sends a `controller_action_predispatch` event before modifying its wishlist.
9. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS wishlist (stocked by the controller):

```
GET    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

10. This call allows the connector to check that the wishlist information are up to date (e.g. the ICS permanent wishlist may have been modified by another application). If need be, the connector refreshes the Magento wishlist.

11. Magento modifies its wishlist by adding the product and sends a `wishlist_items_renewed` event.
12. The connector picks up this event and recovers the current Magento wishlist.
13. The connector saves this recovered wishlist on ICS by making a REST call:

```
POST   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

EDITING A PRODUCT

1. The customer clicks the Edit button and is redirected to the product details page.
2. They can modify the discriminants and/or quantities of the product.
3. They validate their modifications by clicking the Update cart button, which replaces the Add to cart button.
4. What happens next is described in the Product details page section of this documentation, page 8.

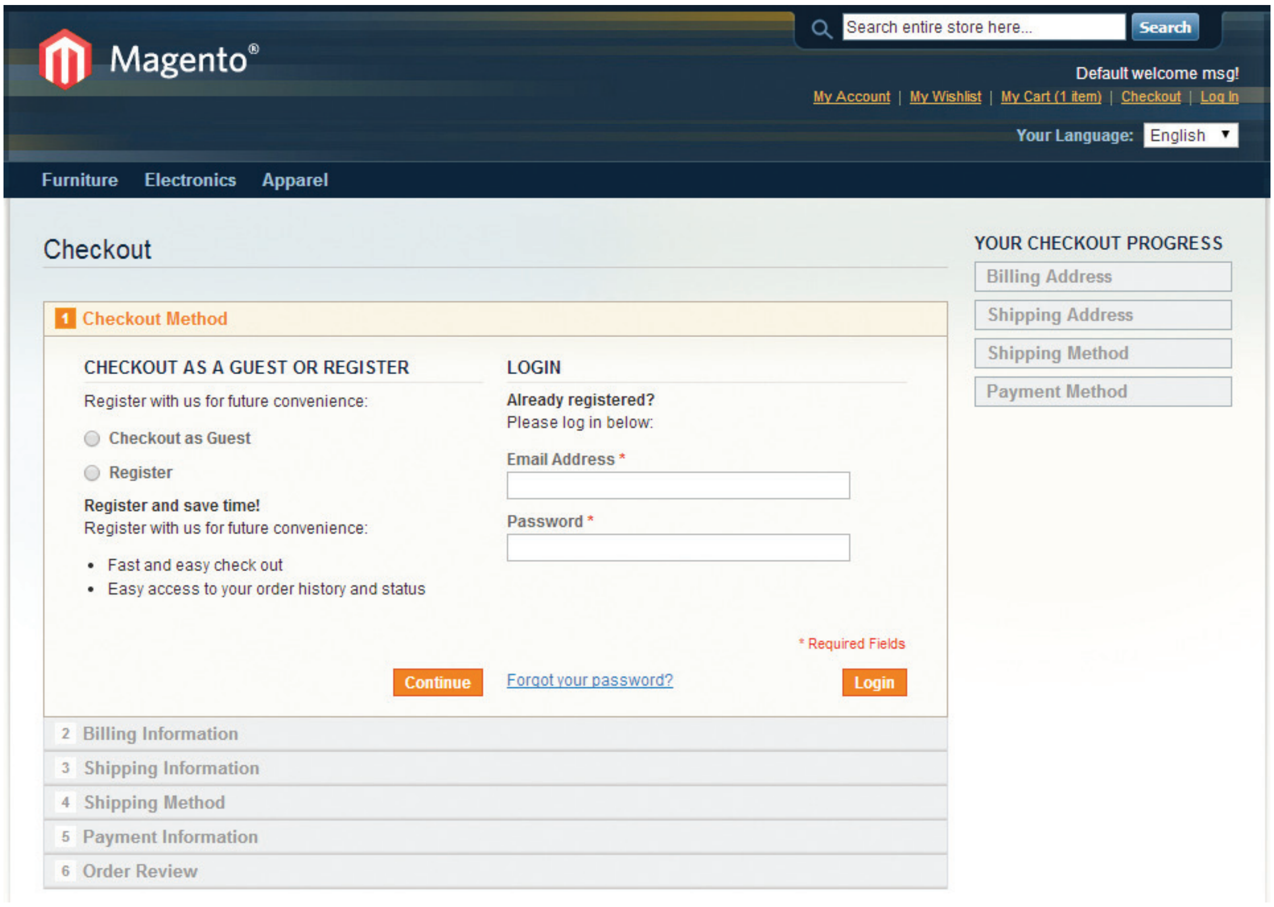
Automatic synchronization

The `controller_action_predispatch` event allows an automatic synchronization between a customer's Magento cart and ICS cart. The same goes for the customer's Magento whishlist and ICS whishlist.

Thus, for each action made by the customer on Magento, the connector will check if the cart and/or wishlist are up to date with ICS and will modify their data on Magento, if need be.

Checkout page

On this page, your customer will complete the order process. They will have the possibility to create a new account or to log into an existing one, if they have not done so already, or to complete their order as a guest. They will have to enter all the required information and then validate the order.



Magento®
Search entire store here...
Search
Default welcome msg!
My Account | My Wishlist | My Cart (1 item) | Checkout | Log In
Your Language: English
Furniture
Electronics
Apparel

Checkout

YOUR CHECKOUT PROGRESS
Billing Address
Shipping Address
Shipping Method
Payment Method

1 Checkout Method

CHECKOUT AS A GUEST OR REGISTER
Register with us for future convenience:
Checkout as Guest
Register
Register and save time!
Register with us for future convenience:
- Fast and easy check out
- Easy access to your order history and status

Continue

LOGIN
Already registered?
Please log in below:
Email Address *
Password *
* Required Fields
Forgot your password?
Login

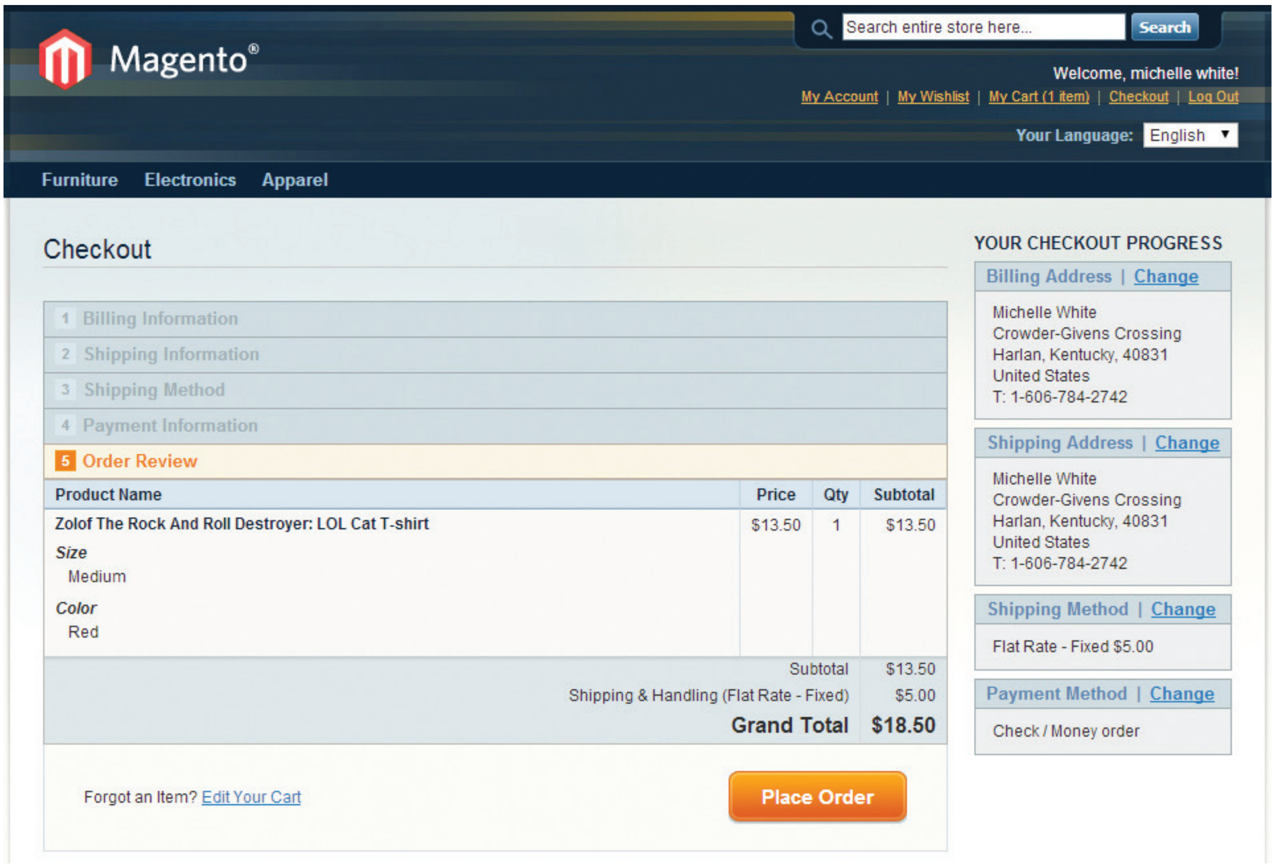
2 Billing Information
3 Shipping Information
4 Shipping Method
5 Payment Information
6 Order Review

- If the customer logs into their account, the ICS mechanisms at play are the same as explained in the Login page section, page 15.
- If the customer creates a new account, the connector makes a REST call to associate the current cart to the newly created account, thus making it the customer's reference cart:

```
POST   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

- If the customer chooses to remain anonymous—guest mode—, ICS will only know their e-mail address.

FINAL CHECKOUT



Search entire store here... Search

Welcome, michelle white!

My Account | My Wishlist | My Cart (1 item) | Checkout | Log Out

Your Language: English

Furniture Electronics Apparel

Checkout

1 Billing Information

2 Shipping Information

3 Shipping Method

4 Payment Information

5 Order Review

Product Name	Price	Qty	Subtotal
Zolof The Rock And Roll Destroyer: LOL Cat T-shirt *Size* Medium *Color* Red	$13.50	1	$13.50
		Subtotal	$13.50
		Shipping & Handling (Flat Rate - Fixed)	$5.00
		Grand Total	**$18.50**

Forgot an Item? Edit Your Cart

Place Order

YOUR CHECKOUT PROGRESS

Billing Address | Change

Michelle White
Crowder-Givens Crossing
Harlan, Kentucky, 40831
United States
T: 1-606-784-2742

Shipping Address | Change

Michelle White
Crowder-Givens Crossing
Harlan, Kentucky, 40831
United States
T: 1-606-784-2742

Shipping Method | Change

Flat Rate - Fixed $5.00

Payment Method | Change

Check / Money order

1. Magento sends a `controller_action_predispatch` event before modifying its cart.
2. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS cart (stocked by the controller):

```
GET   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

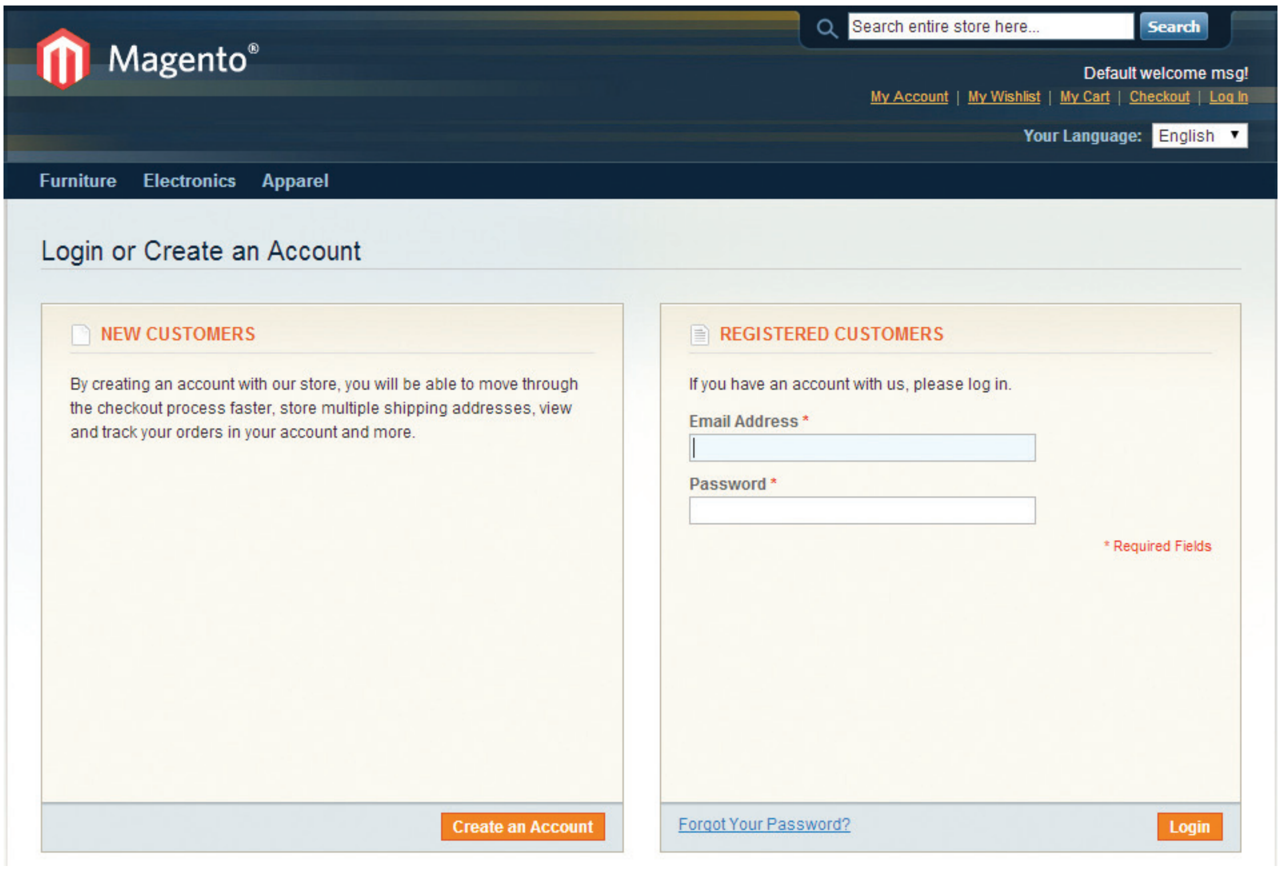
3. This call allows the connector to check that the cart information are up to date (e.g. the ICS permanent cart may have been modified by another application). If need be, the connector refreshes the Magento cart.
4. Magento saves the modifications and sends a `checkout_submit_all_after` event.
5. The connector picks up this event and recovers the current Magento cart.
6. The connector saves this recovered cart as checked out on ICS by making a REST call:

```
POST   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

CUSTOMER

Login page

On this page, an anonymous customer has the possibility to log into their existing account—by using their e-mail and password—or to create a new account.



Search entire store here... Search

Default welcome msg!

My Account | My Wishlist | My Cart | Checkout | Log In

Your Language: English

Furniture Electronics Apparel

Login or Create an Account

NEW CUSTOMERS

By creating an account with our store, you will be able to move through the checkout process faster, store multiple shipping addresses, view and track your orders in your account and more.

Create an Account

REGISTERED CUSTOMERS

If you have an account with us, please log in.

Email Address *

Password *

* Required Fields

Forgot Your Password? Login

1. The customer enters their e-mail address and password, assuming they have created an account by now, and Magento sends a `customer_login` event.
2. The connector picks up this event and makes a first REST call to save the customer's information on ICS:

POST https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/customers/

3. The connector makes another REST call to look for the identifier of a possible current cart stocked on ICS:

GET https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/_search

- **If** this possible cart does exist and if an anonymous cart was being filled:
  a. The connector makes a REST call to merge the two carts:

  POST https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/_merge

  b. The result of this merge becomes the customer's reference cart. The anonymous cart is then deleted by another REST call:

  PUT https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/_delete

- **If** this possible cart does not exist and if an anonymous cart was being filled, this not-so-anonymous-cart-now becomes the customer's reference cart.
- **If** this possible cart does not exist and if no anonymous cart was being filled, the ICS cart remains the customer's reference cart.
- **If** a Magento customer who does not exist on ICS has a permanent Magento cart and if no anonymous cart was being filled, the permanent Magento cart becomes the reference cart of the customer, who will then be created on ICS.

4. The reference cart is loaded into Magento and then saved on ICS (thus, ICS makes sure that all of the products in the cart may be added to the Magento cart—stock, price, etc.):

```
POST   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

5. The connector makes another REST call to recover the authenticated customer's possible wishlist:

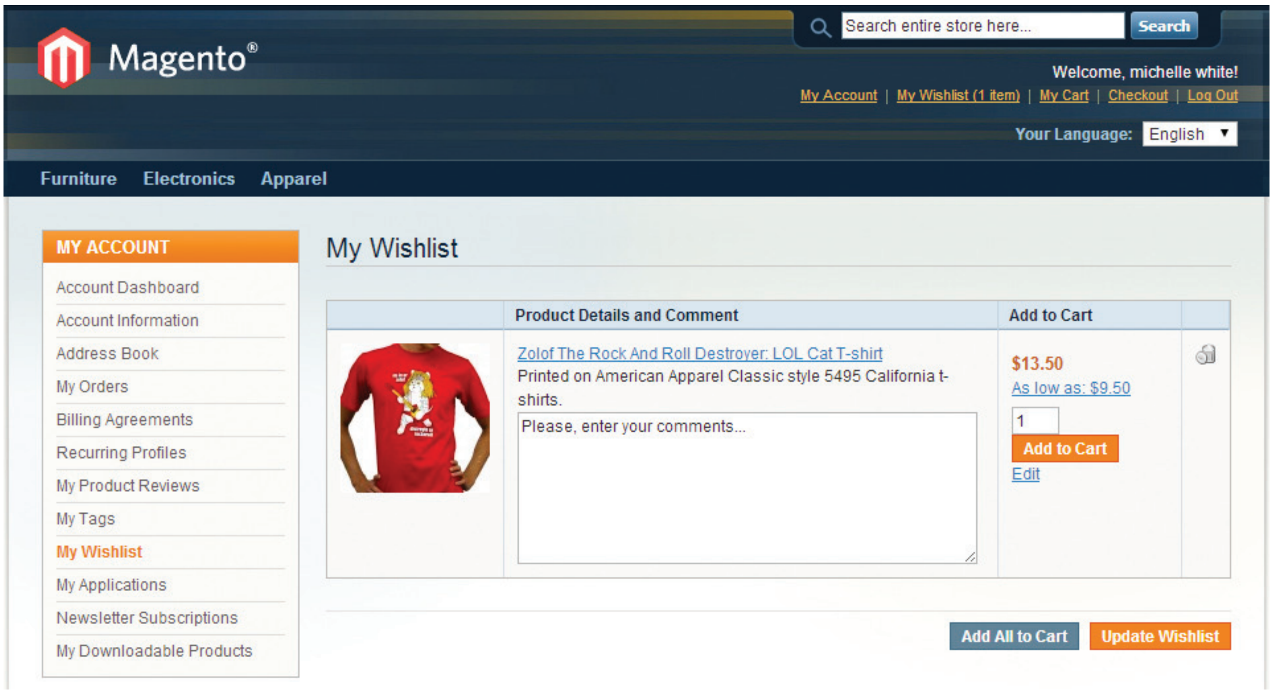
```
GET    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/_search
```

6. If there is a wishlist, the connector uploads it into Magento.
7. The dates of the latest modifications made on the ICS cart and/or ICS wishlist are saved on the customer's Magento session. This will allow an automatic synchronization process between the connector and ICS.

Wishlist page

This page sums the contents of your customer's current wishlist up. From it, they have the possibility to:

- modify or customize a product;
- delete a product;
- add a product to their cart;
- add all products to their cart at one time;
- modify quantities on various lines at one time;
- consult a product details page.



My Wishlist

	Product Details and Comment	Add to Cart	
	Zolof The Rock And Roll Destroyer: LOL Cat T-shirt Printed on American Apparel Classic style 5495 California t-shirts. Please, enter your comments...	$13.50 As low as: $9.50 1 Add to Cart Edit	

CONSULTING THE WISHLIST

1. The customer requests the display of their wishlist.
2. Magento sends a `controller_action_predispatch` event.
3. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS wishlist (stocked by the controller):

```
GET    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

4. This call allows the connector to check that the wishlist information are up to date (e.g. the ICS permanent wishlist may have been modified by another application). If need be, the connector refreshes the Magento wishlist.

MODIFYING THE WISHLIST

1. The customer modifies something (e.g. modifying a quantity, deleting a product, modifying a comment) and clicks the button which validates this modification.
2. Magento sends a `controller_action_predispatch` event before modifying its wishlist.
3. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS wishlist (stocked by the controller):

```
GET    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

4. This call allows the connector to check that the wishlist information are up to date (e.g. the ICS permanent wishlist may have been modified by another application). If need be, the connector refreshes the Magento wishlist.
5. Magento saves the modifications and sends a `wishlist_items_renewed` event.
6. The connector picks up this event and recovers the current Magento wishlist.
7. The connector saves this recovered wishlist on ICS by making a REST call:

```
POST   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

MOVING A PRODUCT OR ALL PRODUCTS TO THE CART

1. The customer clicks the Add to cart or Add all to cart button.
2. Magento sends a `controller_action_predispatch` event before modifying its wishlist.
3. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS wishlist (stocked by the controller):

```
GET    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

4. This call allows the connector to check that the wishlist information are up to date (e.g. the ICS permanent wishlist may have been modified by another application). If need be, the connector refreshes the Magento wishlist.
5. Magento modifies its wishlist by deleting the product(s) and sends a `wishlist_items_renewed` event.
6. The connector picks up this event and recovers the current Magento wishlist.
7. The connector saves this recovered wishlist on ICS by making a REST call:

```
POST   https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

8. Magento sends a `controller_action_predispatch` event before modifying its cart.
9. The connector picks up this event and makes a REST call containing the date of the latest modification made to the ICS cart (stocked by the controller):

```
GET     https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

10. This call allows the connector to check that the cart information are up to date (e.g. the ICS permanent cart may have been modified by another application). If need be, the connector refreshes the Magento cart.
11. Magento modifies its cart by adding the product(s) and sends a `checkout_cart_save_after` event.
12. The connector picks up this event and recovers the current Magento cart.
13. The connector saves this recovered cart on ICS by making a REST call:

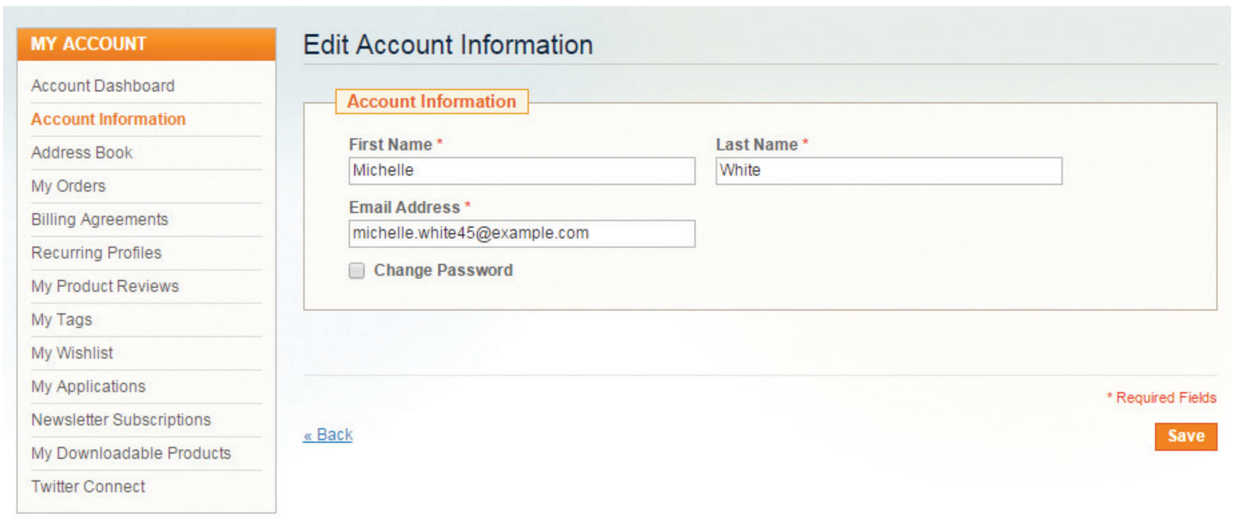
```
POST    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/bags/
```

EDITING A PRODUCT

1. The customer clicks the Edit button and is redirected to the product details page.
2. They can modify the discriminants and/or quantities of the product.
3. They validate their modifications by clicking the Update wishlist button.
4. What happens next is described in the Product details page section of this documentation, page 8.

Account information

On this page, a customer can modify their account information:



Edit Account Information

Account Information

First Name * Michelle

Last Name * White

Email Address * michelle.white45@example.com

Change Password

* Required Fields

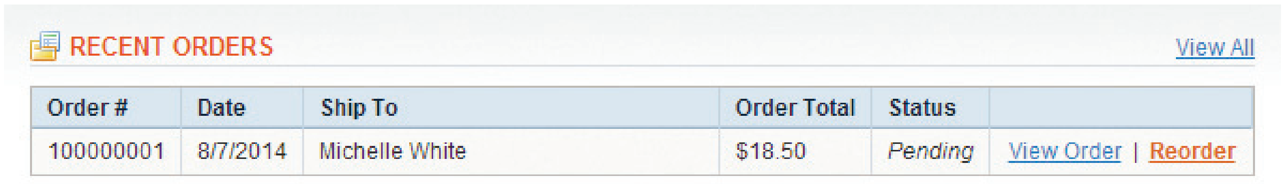
1. The customer changes their information and clicks the Save button.
2. Magento saves the information and sends a `controller_action_predispatch` event.
3. The connector picks up this event and the customer's information, and makes a REST call to save the customer's account on ICS:

```
POST    https://api-ics.intuiko.com/api/tenants/tenantId/brands/brandId/customers/
```

Recent orders

This graphical component gives your customer a list of all their checked-out orders, allowing theme to reorder all the products of a previously ordered cart.

The ICS mechanisms at play here are the same as explained in the Product details page section, pages 8 and 9. However, they do not apply to only one product but to **all** products featured in a previous order.



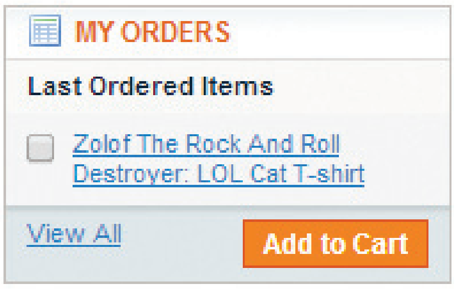
RECENT ORDERS View All

Order #	Date	Ship To	Order Total	Status	
100000001	8/7/2014	Michelle White	$18.50	*Pending*	View Order | Reorder

My orders

This graphical component allows your customer to add a previously ordered product to their cart. They also have the possibility to select several products and add them all to their cart at one time.

The ICS mechanisms at play here are the same as explained in the Product details page section, pages 8 and 9.



Check-in

A check-in is an action made by a customer to indicate that they want to report their presence in a particular store. This allows them, among other things, to request personal advice from a sales associate.

Beware, though: any operation performed by the ICS connector on the cart cancels the check-in.

Omni-channel considerations

MAGENTO PRODUCT TYPES

The Magento product types listed below are stocked in ICS with a specific Magento data frame (“buyRequest”). Thus, a product of any of those types can only be added to the cart by an application that understands this specific date frame.

Those types are:

- groups (wishlists only)
- downloadable (carts and wishlists)
- bundle (carts and wishlists)
- configurable, not complete (wishlist only)

CUSTOM OPTIONS

Magento allows you to implement custom options to your products.

Any product with custom options, wether it is added to a cart or wishlist, is stocked in ICS with a specific Magento data frame (again, the one known as “buyRequest”). Thus, it can only be added to the cart by an application that understands this specific date frame.