Shop Module Version 2.3.0 - Goals

Shop REST API

To be able to implement a great shopping experience modern front-end technology should be supported, e.g. AngularJS or similar. This is best achieved if the main shop functionality is provided by a REST API. The API should support the following methods

Product actions

Reading product data (Magnolia REST API)

```
// not useful for front-end clients:
GET /nodes/v1/shopProducts/[path_to_shop_products]/?depth=[depth]&amp;excludeNodeTypes=[node,types,to,exclude]
// better (contains what is really needed):
GET /v1/shop/{shopName}/products?category={category}
```

- Standard Magnolia REST API?
- depth describes how many levels down in the JCR tree you want to go (from the starting point)
- excludeNodeTypes is probably only necessary when you fetch all products right from the root > you can exclude "technical" nodes like rep:Activities,rep:system,rep:AccessControl...

NOTE: The standard "nodes" Magnolia REST endpoint does not really return what would be needed by a front-end app: Too much clutter, image uuids instead of links, cannot be limited to a category... Therefore a "products" method in the shop endpoint is more useful:

- returns the list of products
- optionally limited to the products of a category

Search...

TBD

Shopping cart actions

Getting the current users shopping cart

```
GET /v1/shop/{shopName}/carts/current
```

- returns the shopping cart (including cart items)
- returns a 404 if the there is no cart for the current user
- shopName is currently ignored > looks like a bug in the shop module but won't hurt in this case

Creating a new shopping cart

```
POST /v1/shop/{shopName}/carts
```

- creates a new shopping cart and stores it in the session (if no cart existed)
  see ShopUtil.setShoppingCartInSession()
- simply returns the current cart
- *should it return an error if the cart already existed?*
- again: shopName is currently ignored

Updating the shopping cart

```
PUT /v1/shop/{shopName}/carts/current
```

- updates the shopping cart, e.g. billing address etc.

- *should it also update the cart items? (easy? hard?)*

Deleting the shopping cart

```
DELETE /v1/shop/{shopName}/carts/current
```

- removes the shopping cart from session (after order is completed)
- *is probably not needed as this should be handled by the "save" command > see below*

Saving the shopping cart, i.e. processing the order (Magnolia REST API)

```
POST /commands/v1/shop/processOrder
```

- Standard Magnolia "command" REST endpoint call
- custom "processOrder" command will then call other commands
    - save order in database (JCR) and remove it from users session
    - create invoice
    - create delivery slip
    - send contact mail (i.e. mail to shop owner, logistics)
    - send confirmation mail (i.e. mail to customer)
    - log order
- should return the info to be displayed on the confirmation message

Cart item actions

If it's too complicated to implement cart item manipulations via the `PUT /shoppingcart` call, then additional methods need to be provided:

Adding a cart item

```
POST /v1/shop/{shopName}/carts/current/items
```

- Adds cart item to cart
- See ShopSigletonParagraphTemplateModel.addToCart()

Updating a cart item

```
PUT /v1/shop/{shopName}/carts/current/items/{itemName}?quantity={quantity}
```

- Updates a cart item (quantity)
- See ShopUtil.updateItemQuantity()
- NOTE: Items in the Magnolia shopping cart have been referenced with the product id... and the index. Product id is usually good enough to identify the cart item. But since the Magnolia shop allows options (e.g. size, color...) two products with different options configuration can be added and will produce separate cart items. To simplify we're using the itemIndex now.

Delete a cart item

```
DELETE /v1/shop/{shopName}/carts/current/items/{itemName}
```

- removes a cart item from the cart
- See ShopSigletonParagraphTemplateModel.updateItemQuantity()
- *probably not needed > updating the quantity to 0 does exactly the same*

Most of the functionality needed to implement this REST API already exists in ShopUtil. Some functionality however is placed in model classes. The shop module should now be refactored so that all of the functionality is placed in a service class and both the model classes and the REST API make use of this functionality. See MSHOP-187 - Move functionality from model classes to service class IN PROGRESS

Support for multiple languages

The dialogs (e.g. shopProduct) do not reflect the i18n settings. Adding a i18n:true to the title field and adding a mapping for the workspace and path did not have an effect. See ~~MSHOP-190~~ - Support for multi-lingual products RESOLVED .

According to Antti Hietala's comments on https://documentation.magnolia-cms.com/display/DOCS/Language+configuration i18n support in content apps seems to be a generic issue. Therefore it is out of the scope of a shop module upgrade.

Configuration improvements

There is a whole series of problems with the shop setup:

- ~~MSHOP-183~~ - Products app "subApp" wrong node type and missing content RESOLVED
- ~~MSHOP-184~~ - Proper "details" sub app to newly generated products app is missing all labels RESOLVED
- ~~MSHOP-189~~ - Get rid of "cartBeanType" in ShopConfiguration RESOLVED
- ~~MSHOP-185~~ - Creation of a new shop fails when there are already two shops which have a price category with the same name RESOLVED
  - Is the selection of a default price category really necessary? Wouldn't it be easier to simply use the first price category in none other has been selected for the cart? Setup currently is awkward as you create a shop where you should select a default price category which does not exist at that point yet.
  - However: Getting rid of the default price category selection in the shop config and instead simply selecting the first price category in the list might break the update path for some installations!

Fix support for multiple shops

Although multiple shops could theoretically be set up (especially once the config bugs above are fixed), the shop name is completely ignored when the carts are accessed. This means that when you run multiple shops in one Magnolia the carts will get mixed up. See
~~MSHOP-188~~ - Shop ignores cart session variable name in shop configuration RESOLVED

When accessing a shopping cart the shop name needs to be placed as prefix in front of the "shoppingCart" session variable name. That should do the trick. At the same time the shopping cart variable name which is configurable in the shop setup (but ignored at runtime) can be removed. This should not have any negative effects on existing installations as it only affects runtime data.

There are other things which do not properly support a multi-shop setup and therefore need to be fixed:

- ~~MSHOP-192~~ - ShoppingCartParagraphModel.getCheckoutFormLink() does not work on a multi-shop setup RESOLVED

Other bugfixes

- ~~MSHOP-44~~ - Rename package info.magnolia.shop.paragraphs to info.magnolia.shop.components RESOLVED
  - Upgrade path: requires proper version handler task!
- ~~MSHOP-48~~ - Order address attributes are not being respected by the SaveAndConfirmFormProcessor RESOLVED
  - test - does this bug really affect 2.x versions?
- ~~MSHOP-52~~ - Field AGBs is not listed correctly in JCR RESOLVED

Other improvements

- ~~MSHOP-194~~ - resetShoppingCart() should not be called from the template but from the form processor RESOLVED
  When is the cart removed from the user session?
  - From the business point of view: It should be removed as soon as the cart is saved. This will prevent further manipulations on the cart.
  - From the technical point of view: It would be best if it would stick around until the confirmation page has been displayed.

- ~~MSHOP-191~~ - Get rid of Content API in shop RESOLVED
- ~~MSHOP-116~~ - As a shop administrator i can send pdf invoices to the customers RESOLVED
- ~~MSHOP-115~~ - As a shop administrator i can change template of invoice pdf via jcr configuration RESOLVED
- ~~MSHOP-111~~ - As a customer i can log into my account and see my previous and current orders CLOSED
- ~~MSHOP-121~~ - As a shop administrator i can get notified via email/pulse message that new order was placed RESOLVED