Dwell Store Technical Documentation

Prepared on	April 10, 2015
Prepared by	Justin Emond, justin@thirdandgrove.com, Third and Grove
Status	DRAFT
Version	20150401-v1

Document Purpose

Third and Grove (TAG) is transitioning support of the Dwell store (store.dwell.com) and Dwell main site (dwell.com) to a new agency. The goal of this document is to speed up the onboarding process for the new development agency supporting the Dwell store and to finalize TAG's work with Dwell.

Server Stack

The Dwell store is hosted on Blackmesh hosting. We have two environments:

Production server

- Magento URL: store.dwell.com
- Magento DB name: dwell_magento_prod
- Drupal URL: dwell.com
- PHP ver: 5.3.29
- MySql ver: 5.5.42
- IP address: 74.121.193.155

Staging server

- Magento URL: store-dev.dwell.com
- Magento DB name: dwell_magento_dev
- htpassword access: dwell/apple
- Drupal URL: dwelldev.prod.acquia-sites.com
- PHP ver: 5.3.29
- MySql ver: 5.5.42
- IP address: 74.121.193.154

Both servers run on Varnish cache, which needs to be cleared whenever any major changes are done. More about it in the "Known issues" section.

Magento Architecture

Magento Enterprise is installed, version 1.12.0.2. Contains about 5000 products which are of simple, configurable and grouped (just a few) type. It is a one website shop with only one store activated, based in the USA with US dollars set as main and only currency.

Core Magento functionality is not heavily customized, regular catalog and ordering processes are used. No third-party checkout systems exists, only standard Magento onepage and multishipping checkout processes are used. Standard coupon code and gift card functionalities are included.

Magento is using Magento's cache system (careful with flushing it, more info in the "Known Issues" section). Compilation system is disabled, and catalog flat tables are not used.

Shop runs on HTTP, while secure pages are on the HTTPS with the valid SSL certificate issued by https://www.digicert.com/ (though checkout pages contain non-https conent, so the SSL validation is not full).

Newsletter subscription - If the user subscribes to a newsletter, he is automatically granted and activated the discount rule (id:31, "15% off promo from newsletter subscribe"). Discount applies only if cart does not contain products which belong to the "Exclude" category (id: 178). This way admin can control to which products a discount can be applied.

Shipping setup - Products whose attribute "Extended Lead Time Product" has value other than "Ships in 1-2 weeks" are unable to be sent with "Expedite shipping" options. For those, only "Free shipping" options are available.

Moreover, some products can't be shipped through "Free shipping", and they are marked by being placed in the "ExcludeFreeShipping" category (id: 283).

If the customer package contains both products which can't be sent through "Expedite shipping", and others which can't be sent through "Free shipping", then additional "Standard shipping" option is activated during the checkout.

Design - The theme is called "dwell2", it is placed in the enterprise package. Skin content is located in the skin/frontend/enterprise/dwell2 folder. CSS files are updated through the SCSS located in the skin/frontend/enterprise/dwell2/sass folder. Both .scss and resulting .css files are part of the GIT repo, so there's no need for generating CSS through Compass on the live server.

Magento Extensions

List of Magento extensions, located in the community code folder:

- Activo Catalog Search extension - overrides core search function, making search results more relevant. More info
- Anarky GDRT extension - for better integration with Google Dynamic Remarketing Tags. More info
- Balke Store Maintenance extension - for making the easy way to set the shop in the maintenance mode, accessed through System->Configuration->Store Maintenance (Advanced Tab). More info
- Bronto extension - Bronto is a third party E-Mail marketing service which can be integrated with Magento. It imports Magento's sales data, runs various analytics, cross-references customers, and allows you to create various kinds of marketing campaigns. It has heavy parametrized configuration for customer and order imports. Newsletter subscriptions are propagated to and run by Bronto. It also has modules for Transactional e-mails, Customer reminders, Post purchase e-mailings, but those are not used. More info
- CLS Paypal extension - stores Credit Card data in the database of those cards which were used on the PayPal. Stored data is **not encrypted**, though it stores only last 4 digits of the CC number, expiry date and CC type.
- Duoplane DPX - custom built module, used for Duoplane API communication, defines only two functions.
- ES Newssubscribers - custom built module, overrides already built custom module (DWELL_Newsletter), sends coupon code to the newly subscribed customers. It's not used anymore, and that functionality is automatized.
- Fooman Speedster extension - for merge and minification of the JS and CSS files. It's slightly customized, a new XML layout attribute is include (minify="exclude") through which files can be excluded from merging and minification, which is good for problematic JS scripts concerning CC validation and similar. Stores minified files in the var/minify_cache folder. More info
- Gigya Social extension - for easy integration with social networks, login, comments and ratings, etc. It is disabled in admin and therefore not used anymore. More info
- Inchoo Featured Products extension - allows you to precisely select products and display them on the home page. It is not used anymore. More info
- Magestore Banner slider extension - is used for management of the main sliding banner on the shop's home page. More info
- Phoenix Moneybookers extension - standard extension for Moneybooker payment system, is integrated with all Magento installations. It is not used on Dwell.
- RocketWeb Google Base Feed Generator extension - is used for customized data feed generation which is used for Google Shopping store. The feed is located on http://store.dwell.com/media/google_base_default.txt . More info

- ™ Easy Catalog Images extension - used for category listing on custom pages. It is not used on Dwell. More info
- Webshopapps - a group of extensions which provides various shipping methods, including Matrixrate, Premiumrate and Productmatrix. out of which only Productmatrix is enabled and used on Dwell. It uses shipping pricing model based on the total package value. Dwell ships only to Canada and USA including while Alaska and Hawaii, though they have a different prices for those two states. More info

List of Magento extensions, located in the local code folder:

- Amasty Color Swatches extension - used on the product page to visualise color attribute with proper color representation. To set the color images visit Catalog->Attributes->Manage Attributes->Color->Attribute Images. Extension has other functionalities as well, but they are not used. More info
- DWELL module is a custom built module that has various functions, mostly used to override core Magento functionality or update it.
    - DWELL_Adminhtml - hardcode disable of the Newrelic extension, which seems is no longer part of the system.
    - DWELL_Catalog - overrides core catalog functions, first, if a users try to access the page of the simple product which is assigned to other configurable product, then the view action is redirected to its configurable parent. Also, setting to skip category URL key in building product URLs is hardcoded to false here.
    - DWELL_Checkout - overrides core checkout function which assigns coupon discount to the customer after he enters the code. The override displays different error messages depending upon the type of the activated coupon - the different messages are shown if the user has activated the “Newsletter coupon” by subscribing to a newsletter.
    - DWELL_Contacts - extends Magento’s core “contact us” function through which customer can send enquiries to the shop owners.
    - DWELL_Core - overrides Magento’s core design setup function with the idea to merge and minify CSS and JS, though Fooman Speedster extension is used for this.
    - DWELL_Directory - used to remove certain regions from the shipping options, such as “American Samoa” or “Armed Forces Africa”.
    - DWELL_DrupalCart - defines the API functions for displaying the cart data in the header (number of items in the cart) on Drupal pages.
    - DWELL_DrupalSSO - used for session integration with Drupal. As already mentioned, customer login and register is done on Drupal system, and the session data is propagated to Magento. This module overrides regular customer login/register function which can be run from customer/account pages and from checkout pages. The function sends the API request to Drupal site, sending the encrypted session ID, and checks for the returned data. If there’s session data on Drupal side, the Magento session is created and from there

    user is logged in on both systems. If there's no session data on Drupal site, user is redirected to the Drupal side for login or register.
    - DWELL_Gateway - defines API calls for check if cart has zero total amount, for checking the quantity of items in the cart or to check the data about used gift cards in customer session.
    - DWELL_GiftCardAccount - overrides adding gift card to the cart action in order to save totals
    - DWELL_GoogleAnalytics - redefines Google Analytics JS code which is integrated to all Magento pages.
    - DWELL_Invitation - upgrades Magento's invitation functionality in order to achieve integration with Drupal
    - DWELL_Magpal - defines API calls for cart session data. **It has an hardcoded authentication key**.
    - DWELL_Misc - used only for refactoring of the get methods for some Magento core config settings.
    - DWELL_Newsletter - overrides core Magento's subscribe action.
    - DWELL_Paypal - overrides PayPal action by modifying parameters, in particular the "capture_complete" parameter, in a way that if not all order items have been shipped, but only some, then the parameter is set to "N" (as opposed to the default "Y").
    - DWELL_Premiumrate - overrides Webshopapps Premiumrate in order to achieve functionality of disabling "Expedite shipping" for items which are not ready instantly. Since Webshopapps Premiumrate is not used online, this module is not used either.
    - DWELL_Signup - second part of overriding the core Magento's subscribe to a newsletter function
    - DWELL_Targetrule - overrides core Magento's crosssell function, in a way that only an item with a certain SKU (sku: 116160000000, product name: Dwell Outdoor 2014 Special Issue) can be displayed in crosssell blocks. This block is displayed on the cart page, with only this item in it.
    - DWELL_Utility - utility module which contains some of the oftenly used methods, mosly used for **Shipping setup**
- GoMage Feed extension - is used for generating feeds for Google Shopping and other systems, though it is disabled on DWELL and therefore not used. More info
- HWG Attribute manager extension - enables to shop admin to manage customer and category attributes in the same way as catalog product attributes are managed. More info
- Magestore Magenotification extension - used for polling notifications from Magestore's website, concerning extension updates and new releases
- MD QuickView extension - for quick display in popup of the product data on category pages. It is not used on DWELL. More info
- Magento Mercent extension - for integration with third-party service called Mercent. It generates feeds about products and inventory and stores them in media/mercentfeed/

folder. These data are fetched by Mercent service, and from there various analytics data is available. More info
- Ordereditor extension - allows shop admin to edit order without cancelling it and resubmitting it as a new order.
- Redstage Gues Order View extension - allows guest customers to come back to the site and track the status of their order. More info
- VINDESIGN Updatepdp extension - updates product's certain data by pulling it from Drupal site from the following URL: dwell.com/magento-info/ . Basically it is an extension of the DWELL_DrupalSSO module.

Some of the core functionalites are also upgraded/changed directly by rewriting the core Magento block/model files. These changes are located in these 3 folders:

- app/code/local/Mage
    - Customer/Block/Address/Edit.php - prefills the address data with default billing or default shipping address, if they are set in the customer's profile
    - Customer/Block/Widget/Fullname.php - overrides core Magento customer block name widget, in order to use the different template file
    - Customer/Helper/Data.php - hardcodes preventing customers from registering directly in Magento. Registration is done on the Drupal site
    - Customer/Model/Address/Abstract.php - makes customer telephone field not-required, as opposed to the default Magento model where it is required
    - Page/Block/Html/Topmenu.php - redefines the main menu structure by reimplementing its draw function
    - Payment/Model/Method/Cc.php - redefines credit card validation function in order to include credit card type into the info block of the payment
    - Paypal/Model/Payflowpro.php - overrides PayPal payment method in order to include the payment request changes implemented in DWELL_Paypal module (see above for more information).
    - Sales/Model/Order/Creditmemo.php - disables e-mail notification if the order came from Mercent module
    - Sales/Model/Order/Shipment.php - disables e-mail notification if the order came from Mercent module
    - Sales/Model/Order.php - fixes the bug where PayPal orders older than 30 days can not be canceled because payment authorization went out of date.
    -
- app/code/locel/Enteprise - CustomerBalance module is overridden, to add support for Mercent module
- app/code/local/Varien - overrides core Varien function for image resizing by setting the minimal JPEG quality to 90% (even if through API less quality % is set).

Drupal and Magento Integration

The Magento site integrates with the Drupal site (dwell.com) in three important ways: Single sign on, product imports into Drupal, and content imports into the store from Drupal.

Single Sign On

The Magento store implements a custom single sign on (SSO) solution with Drupal (dwell.com).

The primary mechanism of the SSO integration is a simple, custom REST API hosted on the Drupal site (see the dwell_magento custom module). The first REST API takes a Magento session ID as an argument and returns whether or not that Magento session ID is tied to an active Drupal session. If it is, information about the user is returned.

The Drupal registration flow is also modified to associate a Magento session to a Drupal session and to redirect users back to Magento after a successful login or registration event.

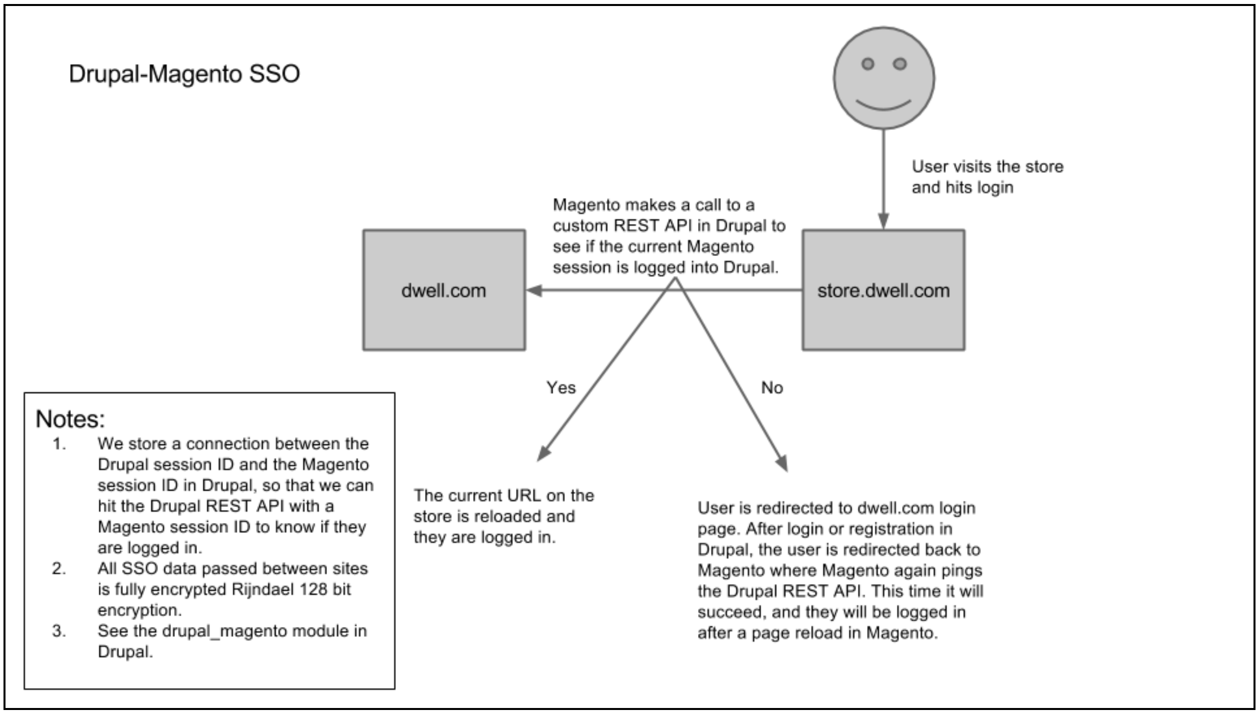
The API calls are fully encrypted using encryption best practices.

Here is the SSO step by step process for a new user:

1. A Magento user clicks register
2. Magento starts a user session
3. Magento calls the Drupal API to see if the current Magento session is associated to a Drupal session
4. If the response is no, Magento redirects the user to a custom endpoint in Drupal
    a. The custom endpoint captures the Magento session ID redirects the user instantly to the registration page.
    b. After completing login or registration Drupal associates the user's Drupal session ID with the Magento session ID
    c. Drupal redirects the user back to the page they came from on the Magento site
5. If the response is yes, Magento programmatically logs the user in and reloads the page to complete the login.

See the Magento extension DWELL_DrupalSSO and in Drupal the dwell_magento module.

This diagram shows the process:



Drupal-Magento SSO

User visits the store and hits login

Magento makes a call to a custom REST API in Drupal to see if the current Magento session is logged into Drupal.

The current URL on the store is reloaded and they are logged in.

User is redirected to dwell.com login page. After login or registration in Drupal, the user is redirected back to Magento where Magento again pings the Drupal REST API. This time it will succeed, and they will be logged in after a page reload in Magento.

Notes:

1. We store a connection between the Drupal session ID and the Magento session ID in Drupal, so that we can hit the Drupal REST API with a Magento session ID to know if they are logged in.
2. All SSO data passed between sites is fully encrypted Rijndael 128 bit encryption.
3. See the drupal_magento module in Drupal.

Drupal Product Import

Magento product information is consumed in Drupal as nodes, so that the content management tools in Drupal can be used to show product information alongside Dwell.com content.

The custom module developed for this integration has been released to contrib: https://www.drupal.org/project/magento_integration

In addition, the Drupal dwell_magento module has some custom enhancements to the integration.

There are two main parts to this integration.

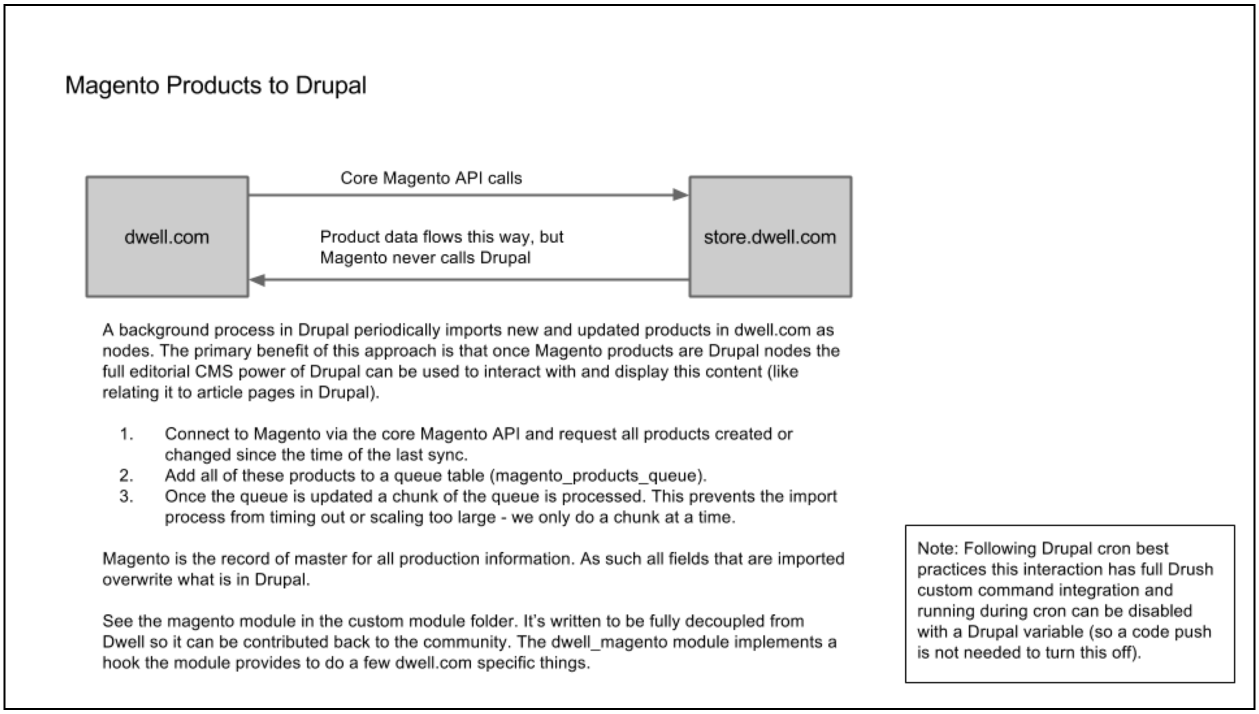
Every cron run Drupal pings the Magento API to get a list of products changed or added in Magento since the last time Drupal checked. (This date is stored in a Drupal variable.) It then adds all the product IDs for all updated/created products to a queue table in Drupal.

In a separate process triggered by cron, the module imports the full product information into Drupal nodes. We keep a link between the Drupal node ID and the Magento product ID so that products in the queue already imported into Drupal will update the existing node and not create a new one in Drupal. The import process updates data on existing Drupal, Magento linked nodes, so changes on these fields in Drupal will be lost.

The contributed Magento module includes a custom hook we wrote so the imported $node could be modified before saving.

The Magento product import queue can be processed manually with drush calls (we wrote full drush integration), in the event the queue is backed up.

This diagram provides an overview of the product import:



Magento Products to Drupal
Core Magento API calls
dwell.com
Product data flows this way, but
Magento never calls Drupal
store.dwell.com

A background process in Drupal periodically imports new and updated products in dwell.com as nodes. The primary benefit of this approach is that once Magento products are Drupal nodes the full editorial CMS power of Drupal can be used to interact with and display this content (like relating it to article pages in Drupal).

1. Connect to Magento via the core Magento API and request all products created or changed since the time of the last sync.
2. Add all of these products to a queue table (magento_products_queue).
3. Once the queue is updated a chunk of the queue is processed. This prevents the import process from timing out or scaling too large - we only do a chunk at a time.

Magento is the record of master for all production information. As such all fields that are imported overwrite what is in Drupal.

See the magento module in the custom module folder. It's written to be fully decoupled from Dwell so it can be contributed back to the community. The dwell_magento module implements a hook the module provides to do a few dwell.com specific things.

Note: Following Drupal cron best practices this interaction has full Drush custom command integration and running during cron can be disabled with a Drupal variable (so a code push is not needed to turn this off).

Magento Content Import

The Magento store imports content from Drupal. Specifically, Magento uses an API provided by the Drupal Services module to consume information on brands stored in Drupal for display on the product detail pages in Magento.

Third-Party Integrations

E-mail newsletter signup pop-up is set to be displayed automatically on the first opening of the home-page. If subscribed, the user automatically receives discount which is implemented through the Shopping Cart Price Rule called “15% off promo from newsletter subscribe”.

The Magento store is integrated with Duoplane for order and shipping management, though integration seemed a bit buggy and did not work properly at the time. Integration is done by following Duoplane’s guide and using Magento API users.

Bronto is used as an e-mail marketing service, it pulls up Magento’s data, and enables shop owner to create various e-mail campaigns based on the data analysis.

Branching Workflow

Here is the branching process we have followed:

- Create a new development branch from master for the next release, say v1.5.
- This new branch is either committed to directly or team members branch off of this branch for individual JIRA tickets, which are then merged in.
- Typically dev will have the development branch checked out.
- The sprint would continue with QA and feature branches being merged into the development branch and the dev server being reset from production as needed.
- At the time of deployment to production the first step is to merge the development branch into master, tag master with the release (say v1.5), and deploy that tag to development.
- Clear cache procedure (as described in *Known Issues, point 2*)
- When the deployment is complete the development branch can be deleted and a new one can be cut from master.

The deployment process is programmed in the script tools/deploy.sh, although the script is not currently configured to work with the current hosting stack.

Additional Information

Repository contains folder named “tools” which contains different SSH scripts which can be useful and ease some of the necessary tasks. Shell scripts are used for database copy and update, while other PHP scripts are used for coupons generation, order removal, etc.

For example, Magento cache can be easily cleared by calling

PHP clearCache.php --all

Approach on High Priority JIRA Tickets

- DS-79 - Update email signup thank you text - a simple template file update is necessary, file is located in the template/newsletter/subscribe path.
- DS-82 - New Social Share Buttons for Products - a redesign of the social icons on the single product page. Right now they are managed and displayed by Gigya module, and depending upon the agreed solution it can be continued like that, or the Gigya module can be removed from this page and icons can be installed manually. If the requested design contains non-standard share icons, then different share functions must be called than those implemented by Gigya module.

- DS-74 - Product bug: Sales items with multiple attributes showing full price - Run a diagnostics, change the shop theme to the base Magento and test the issue to see if the issue is theme-related. My guess is that this is most likely the case since “NAN” is almost never returned by PHP, and in this case is a result of the Javascript calculations, which are ran after the configurable options are picked. Also, check if the JSON object which contains all available product options and prices is not corrupted.
- DS-68 - 'From our Store' price layout - a simple template/css update is necessary to fix this
- DS-84 - MineWhat Plugin - should be a pretty straightforward extension installation, setup and testing. Although, we should keep in mind that this extension on Magento connect is not available for Magento Enterprise, only for Community edition, so a careful approach is necessary (or even maybe to contact the extension vendor).
- DS-85 - Need log for product feed - quickest and easiest solution would be to make a symlink to these files in the “media” folder since the media folder is accessible by public users, and change the file permissions if necessary.
- DS-77 - Need log for product feed - couldn’t reproduce this issue, but is most likely attached to the varnish-magento cache collision described in the “Known Issues”. The questionable URLs are proper Magento image URLs, Magento system is caching resized images in order to make the faster fetch and delivery. This is how it should be.

Local Setup

Passwords need to be changed in tool scripts as well in order to maintain their functionality, for example the app/etc/local.xml script where DB passwords should be set.

To set up the Magento store locally:

1. Add store-loc.dwell.com to /etc/hosts:
   ```
   127.0.0.1   store-loc.dwell.com
   ```
2. Go to Sites directory and clone git repo into a new directory dwell_mage
3. Copy database configuration for local:
   ```
   cd dwell_mage
   cp docroot/app/etc/local.xml.vagrant docroot/app/etc/local.xml
   ```
4. Follow the instructions described here: https://docs.vagrantup.com/v2/synced-folders/nfs.html to bypass the password prompt for the nfs share.
5. Go to 'tools/vagrant' & run './vagrant-boot.sh'
6. Go to http://store-loc.dwell.com:6243

Known Issues

There are several known issues on the store site:

1) Sometimes when activating a new branch on the dev, the site displays the PHP error from a random Magento class. In that case the database needs to be refreshed by pulling it from production server (manually delete all tables from current database and then run **tools/dbcopy/dwell-db-copy-blackmesh.sh**) and all contents from var/cache folder should be manually deleted.

2) When activating changes on both dev and production, some images may appear broken, due to Magento image cache/Varnish cache collision. To prevent this always clear cache in the following procedure:
    a) Clear entire magento cache
    b) Clear magento image thumbnail cache
    c) IMMEDIATELY clear varnish cache

3) Some scripts contained in the tools folder aren't currently support (deploy.sh for example), but they can be helpful as a guide for steps necessary to complete the task.

4) Occasionally the store search will return no results, due to an indexing problem. To fix, go to Admin > System > Index Management, select all & reindex.

Login Information

In the interest of security, login information will not be included in this document. Third and Grove has already securely delivered to Mark Lavoritano at Dwell a text file with the complete list of access information we have on file for Dwell.

Third and Grove maintains an encrypted wiki to store client login information. This information may be purged after May 1, 2015 and thus may not be recoverable by Dwell after this date.

Transitioning from Third and Grove

For security reasons it is important to limit access to the store technology stack only to those teams actively working on the site. The following is a list of recommended changes to completely remove TAG access to the Dwell store:

1. Remove TAG IPs from Blackmesh firewalls
2. Change root SSH passwords
3. Change SSH access password for production server
4. Change SSH access password for staging server (user: cwadmin)
5. Change MySQL database passwords
6. Change the Magento admin password
7. Disable TAG Magento admin accounts
8. Change the popup login on the development site
9. Disable our Blackmesh support portal accounts

10. Remove us (if added) as authorized users on the Blackmesh servers
11. Remove or update our Bronto account
12. Change the Blackmesh Varnish cache clear portal password
13. Change the Duoplane password
14. Change the Magento API users password (attached with Duoplane)
15. Change the Paypal password (be sure to verify this change will not impact store checkout connectivity)
16. Change the Paypal Manager login (be sure to verify this change will not impact store checkout connectivity)
17. Set up a new GitHub repository and transfer all code to this new repository.
18. Change the Magmi login.
19. Update the Magento API connection details used between Drupal and Magento (ensure this change does not break single sign on connectivity).
20. Update the Magento Connect account password.
21. Update the AWS Cloudfront password (ensure this change does not break CDN connectivity).

Additional Information

Kindly contact Justin Emond for any additional information, questions, or concerns:

Justin Emond
justin@thirdandgrove.com
http://www.thirdandgrove.com
Cell: (213) 840-1721