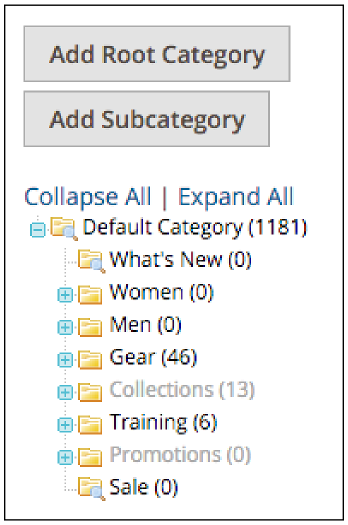
Planning your categories

Before creating your multiple stores, you need to plan your product category structure. In the Magento GWS hierarchy, Websites are assigned to root categories. Root categories are not shown to your visitors, but rather act as the top level under which all your subsequent categories reside.

Categories versus catalogs

There can be some confusion when using the terms categories and catalogs. This is due to the fact that Magento uses root categories as a synonym for catalogs. To keep this straight, a catalog is all the products within one group, represented by the group's root category. Any level below the root category is what we will call categories throughout this book.

The following image shows the category hierarchy of the sample data you may have installed during installation:

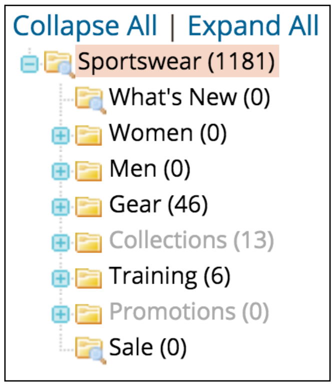


While Magento names the top sample data category, **Default Category**, you can use any name you wish. The categories at the next level below the **Root Category** are the top-level categories which will usually appear on the navigation bar on the site.

To accomplish our example configuration, we need to rename the **Default Category** and create a new root category:

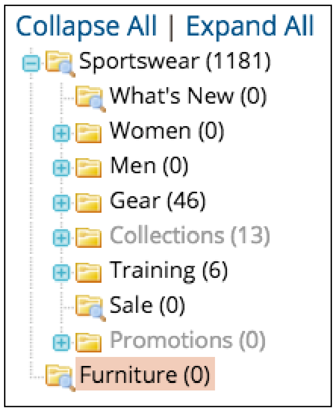
1. Click on **Default Category** shown on the left side of the edit area.
2. In the **Name** field, change **Default Category** to **Sportswear**.

3. Click on **Save Category** in the upper right-hand corner of the screen (a red button).



Our next step is to create the additional **Root Category** we will use for our furniture catalog:

1. Click on **Add Root Category**.
2. In the **Name** field, enter **Furniture**.
3. Select **Yes** for **Is Active**.
4. Click on **Save Category**.

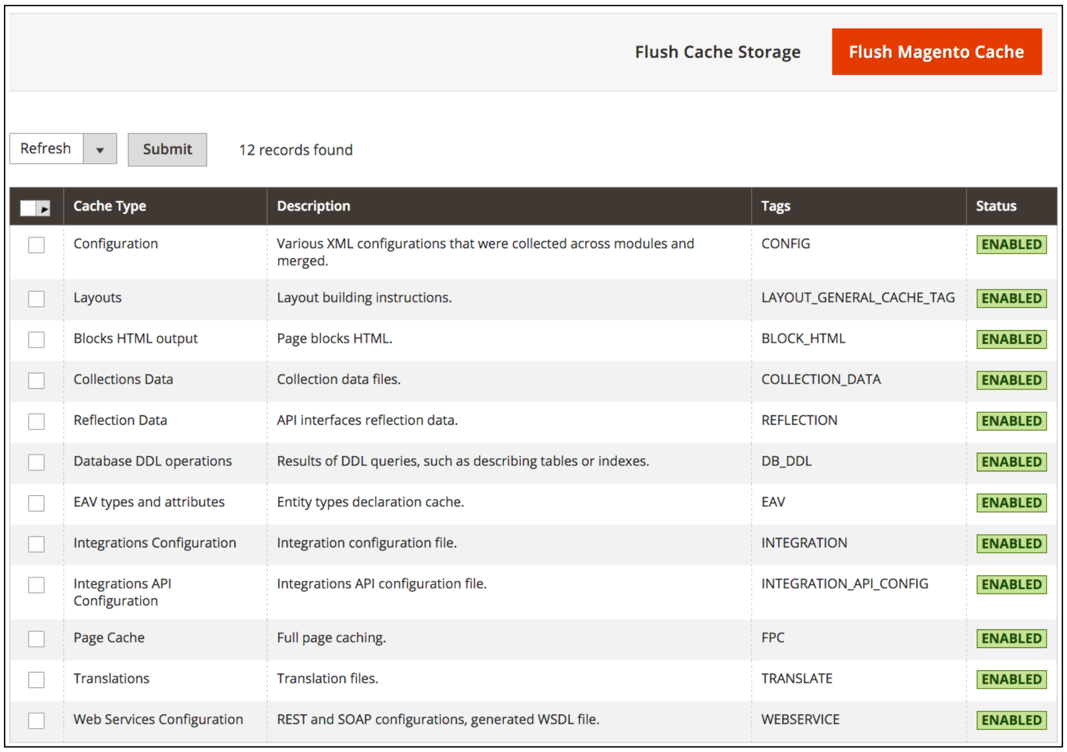


If you want, you can add subcategories within these root categories, but for the purposes of creating multiple stores, it is not necessary at this point.

You may be asking at this point, can multiple Websites share the same root category? The answer is yes. Due to the extensiveness of the GWS hierarchy, you can use the same product catalog for more than one business or website. For example, you might be creating multiple business entities, each selling the same products, but with separate payment gateways, store owners, and so on. While you could create duplicate catalogs with the same products, you can also use one catalog for both businesses. With GWS, you can control pricing, availability, and many other product attributes for each business. We'll be discussing categories and products in more detail in *Chapter 3*, *Managing Products*.

Disabling the cache

While we will discuss caching more fully in *Chapter 9*, *Optimizing Magento*, as you build your stores, categories, CMS pages and more, you should turn off the Magento cache so your changes will appear immediately in your new store. In fact, any time you make changes to your Magento store, it's helpful to turn off the Magento cache.



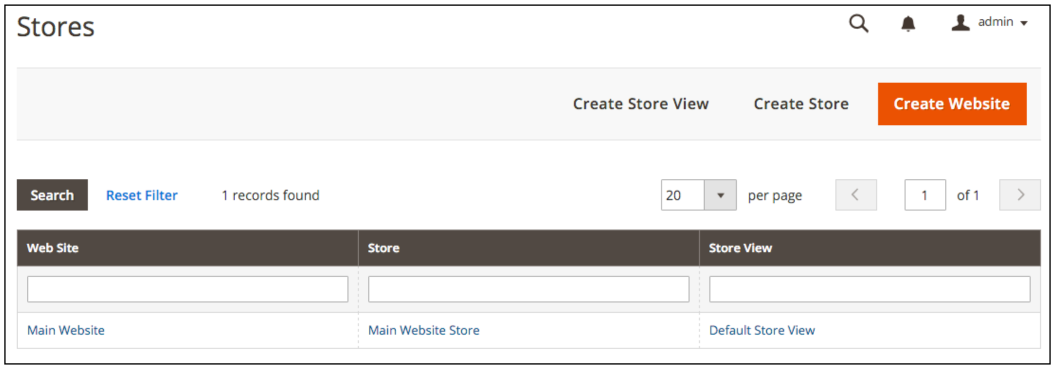
	Cache Type	Description	Tags	Status
☐	Configuration	Various XML configurations that were collected across modules and merged.	CONFIG	ENABLED
☐	Layouts	Layout building instructions.	LAYOUT_GENERAL_CACHE_TAG	ENABLED
☐	Blocks HTML output	Page blocks HTML.	BLOCK_HTML	ENABLED
☐	Collections Data	Collection data files.	COLLECTION_DATA	ENABLED
☐	Reflection Data	API interfaces reflection data.	REFLECTION	ENABLED
☐	Database DDL operations	Results of DDL queries, such as describing tables or indexes.	DB_DDL	ENABLED
☐	EAV types and attributes	Entity types declaration cache.	EAV	ENABLED
☐	Integrations Configuration	Integration configuration file.	INTEGRATION	ENABLED
☐	Integrations API Configuration	Integrations API configuration file.	INTEGRATION_API_CONFIG	ENABLED
☐	Page Cache	Full page caching.	FPC	ENABLED
☐	Translations	Translation files.	TRANSLATE	ENABLED
☐	Web Services Configuration	REST and SOAP configurations, generated WSDL file.	WEBSERVICE	ENABLED

To disable the Magento cache, follow these steps:

1. Go to **System** | **Cache Management** in your Magento backend.
2. Using the **Mass Actions** drop-down menu (the checkbox at the top of the first category), choose **Select All**.
3. Select **Disable** in the **Actions** drop-down menu.
4. Click on the **Submit** button.

Set up websites, stores, and store views

Next, let's go to **Stores** | **All Stores** in your Magento backend. The following screenshot shows the sample data configuration:



Web Site	Store	Store View
Main Website	Main Website Store	Default Store View

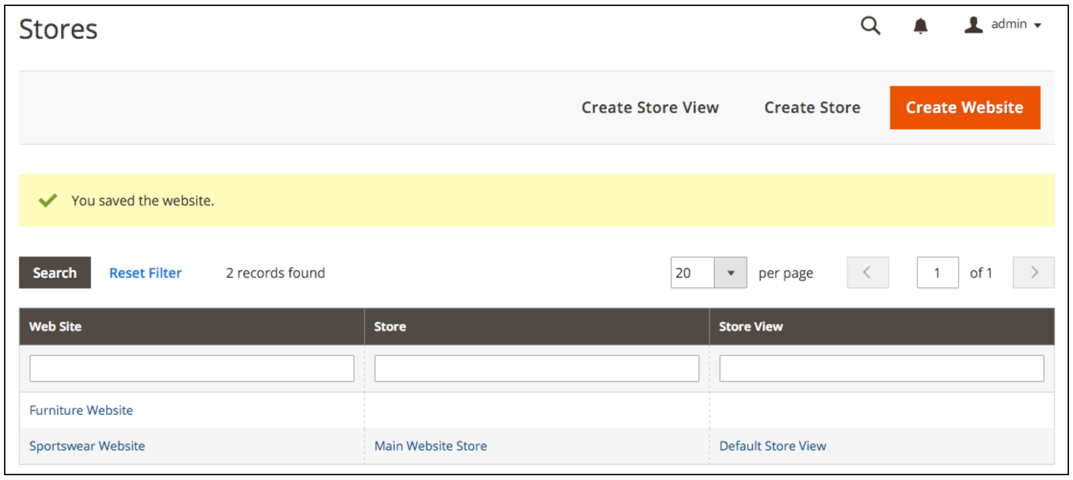
First, let's change the name of the default website to coincide with our planned hierarchy:

1. Click on **Main Website**.
2. Change the **Name** to **Sportswear Website**.
3. If you wish, you can also change code to any lowercase word (with no spaces), such as **sportswear**.
4. Click on **Save Website**.

Next, let's create the furniture website:

1. Click on **Create Website**.
2. Enter **Furniture Website** for **Name**.
3. Enter **furniture** for **Code**.
4. Click on **Save Web Site**.

Your **Stores** panel should now look like this:



You saved the website.

2 records found

Web Site	Store	Store View
Furniture Website		
Sportswear Website	Main Website Store	Default Store View

Next, let's create the stores for our example:

1. Click on **Create Store** at the top of the screen.
2. Make sure that **Furniture Website** is chosen in the **Web Site** dropdown menu.
3. Enter **Furniture Store** for **Name** (you can use any name you wish).
4. Choose **Furniture** for the **Root Category** to assign the product catalog containing your furniture products to this store.
5. Click on **Save Store**.

Now, let's rename the sportswear website store:

1. Click on the **Main Website Store** link in the **Stores** panel.
2. Change the **Name** to **Sportswear Store**.
3. Click on **Save Store**.

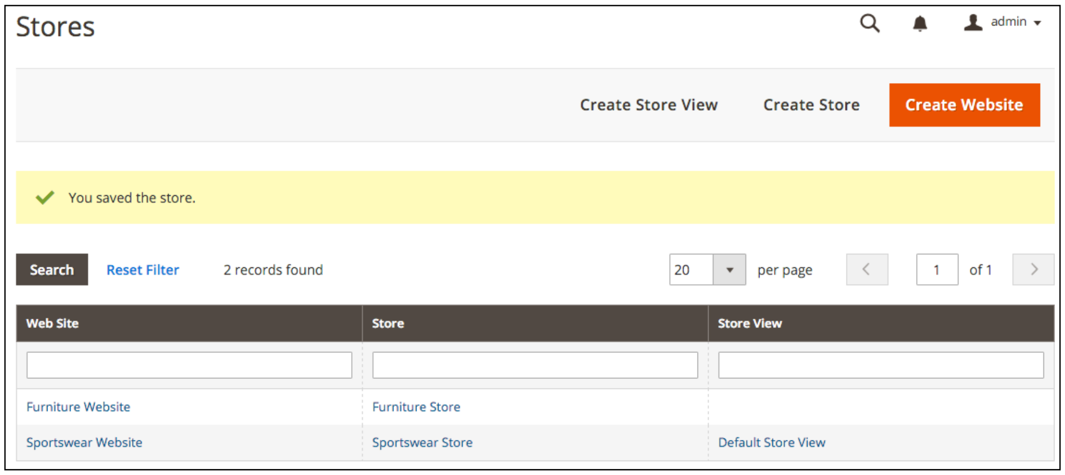
Now, let's stop a moment to review what we just did:

- We created two websites, one for each business (sportswear and furniture)
- Then, we created two stores (sportswear and furniture)

But wait! The example shows four stores; sportswear is to have three stores, one each in English, French, and German.

Since all the sportswear stores will sell the same products, just in different languages (and, most likely, currencies), there really is only one sportswear store. There will be, as we'll create next, three store views. This is where the store/store view nomenclature can get a tad confusing. Think of stores as analogous to different physical stores in different cities. Store views are different *entrances* to those stores. In our example, we have one store with different entrances for English-speaking, French-speaking, and German-speaking customers.

At this point, our **Manage Stores** screen should look like this:



You saved the store.

2 records found

Web Site	Store	Store View
Furniture Website	Furniture Store	
Sportswear Website	Sportswear Store	Default Store View

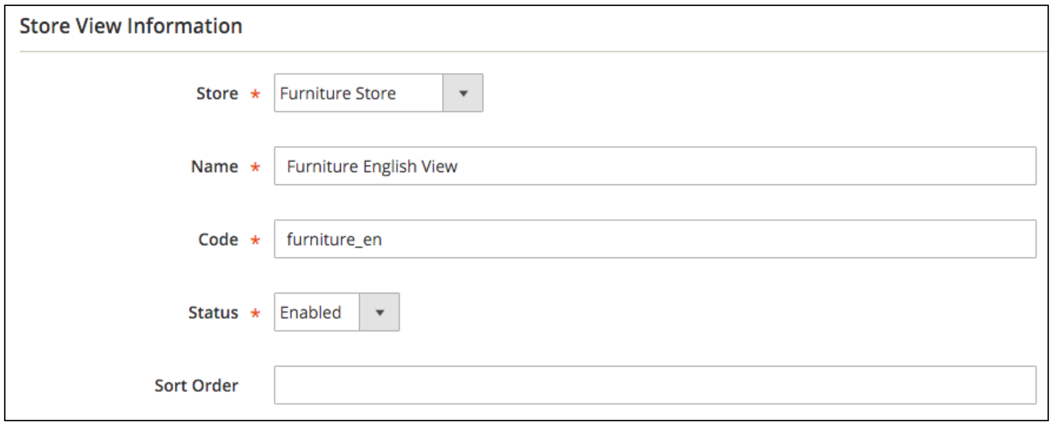
Finally, we will create our necessary store views:

1. Click on **Create Store View** at the top of the **Stores** panel screen.
2. Choose **Furniture Store** for **Store**.
3. Enter **Furniture English View** for **Name** (you can use whatever name you wish).
4. Enter a lowercase **Code**. You can use underscores ("_") as well, but no spaces or other characters other than letters and numbers.

> We often use a language abbreviation in the code, such as `furniture_en` for Furniture, English. You can, of course, use `furniture_english`, or if you're not going to create multiple languages, simply use `furniture` or `furniture_store`.

5. Choose **Enabled** for **Status,** unless you don't want the store view active for any reason.

6. Click on **Save Store View**.



Store View Information

Store *	Furniture Store
Name *	Furniture English View
Code *	furniture_en
Status *	Enabled
Sort Order	

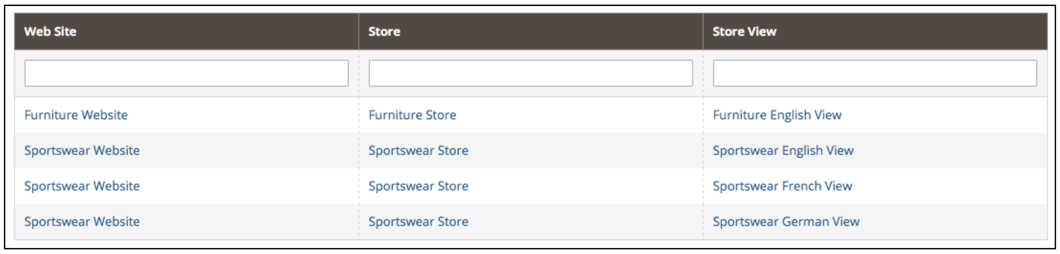
Now we can turn our attention to creating the first store view for our sportswear store:

1. Click on **Default Store View** shown in the list of stores.
2. Update the **Name** to something such as **Sportswear English View**.
3. You can leave the **Code** as **default**, but for consistency's sake, you may want to change it to **sportswear_en** or similar.
4. Click on **Save Store View**.

As with the website code, consult your developer if you're working on a live Magento installation before changing.

To create the additional sportswear store views, use the same technique you used to create the **Furniture English Store View**, substituting the appropriate language nomenclature where appropriate (for example, `sportswear_fr`, `sportswear_de`, for French and German respectively).

When completed, your **Stores** panel should look as follows:



Web Site	Store	Store View
Furniture Website	Furniture Store	Furniture English View
Sportswear Website	Sportswear Store	Sportswear English View
Sportswear Website	Sportswear Store	Sportswear French View
Sportswear Website	Sportswear Store	Sportswear German View

You have now created four new store views for two stores of two websites.

Nginx versus Apache

Before our stores are "reachable" by visitors, you'll have to undertake one additional bit of work, specifically, setting the `mage_run_code` and `mage_run_type` variables for the webserver. Traditionally, nearly all Magento hosting has used the Apache web server. The Nginx web server has been gaining popularity of late, however, and has reached something of a critical mass when it comes to hosting Magento. For this reason, we'll be including multistore webserver configurations for both Nginx and Apache.

Configuring Apache

The most popular method for configuring multiple stores in Magento is modifying the `.htaccess` file. It's also possible to configure multiple stores using Apache's **virtual host declaration**, so we will cover both methodologies here.

Modifying the .htaccess file

The `.htaccess` file resides at the root of your Magento installation directory. This important file controls access to the directory, as well as setting certain parameters for handling HTTP requests from visitors to your site.

For our purposes, we need to add additional code that will direct requests to the proper online store.

Before editing the `.htaccess` file, it is always a good practice to make a copy as a backup in case you need to revert to the original.