

# Magento Mobile How-to

## Create and configure your own Magento Mobile application and publish it for the Android and iOS platforms

Darko Goleš

[PACKT] PUBLISHING

# Magento Mobile How-to

Create and configure your own Magento Mobile application and publish it for the Android and iOS platforms

**Darko Goleš**

[PACKT] PUBLISHING

BIRMINGHAM - MUMBAI

# Magento Mobile How-to

Copyright © 2012 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: November 2012

Production Reference: 1191112

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-84969-366-0

www.packtpub.com

# Credits

**Author**
Darko Goleš

**Reviewers**
Anutosh Ghosh
Allan MacGregor
Amr Shahin
Marcin Szterling

**Acquisition Editor**
Sarah Cullington

**Commissioning Editor**
Meeta Rajani

**Technical Editors**
Prasad Dalvi
Jalasha D'costa

**Project Coordinator**
Esha Thakker

**Proofreader**
Aaron Nash

**Graphics**
Sheetal Aute

**Production Coordinator**
Prachali Bhiwandkar

**Cover Work**
Prachali Bhiwandkar

**Cover Image**
Darko Goleš

# About the Author

**Darko Goleš** is a Magento Certified Developer who works and lives in Croatia. He has spent over nine years learning and developing both desktop and web applications in different programming environments such as VB6, C++, VB. NET, C#, and later in PHP—Joomla, Wordpress, and so on.

After working on different projects and learning different platforms, he found that his real passion in the developer's world is web development and PHP.

For the last year and a half, he has been working as a Senior PHP Web Developer for Inchoo Ltd (`www.inchoo.net`).

In the beginning of his career with Inchoo, he worked as a web support Mobile team member, specializing in Web Services API development.

During that time, he developed DivineofficeHub (a content management and web service-based data delivery platform) for Surgeworks Ltd, which is used as a data provider for several Surgeworks mobile applications built by Inchoo.

After over 10 months of additional experience with web services using the Symfony2 framework, he started with Magento and then worked on extending Magento Core API features for its usage with the Mageboard iOS application.

Currently, Darko is working as a Magento backend developer on different projects, solving Magento challenges every day.

---

Like most developers and book authors, I have to say that it is impossible to accomplish any success or knowledge without investing a lot of time.

Because of that, I want to use this opportunity to say a big thank you to my family (my wife Tihana, and my kids Renato and Lana) for selfless support, understanding and caring in all of these hours that I spent not just writing this book, but all the precious family time stolen from them to build myself as a professional developer.

---

# About the Reviewers

**Anutosh Ghosh** loves coding, but has worked extensively only in the world of PHP, for almost five years now. He has a good knowledge of Magento, and has worked on the integration of Magento Web Services with SAP Business One for more than two years.

He is trying hard to figure out the jargon of Java, amongst other things.

When bored, he gets some serious recreation by watching cool movies and singing regional stuff. However, he loves to poke his nose in some forums and Stack Overflow, from time to time.

---

Whatever I am today is only because of my family, especially my mother, whose perseverance and experience has always been my base.

---

**Allan MacGregor** is Magento Certified Developer with four years of Magento experience. He started working with Magento as a freelance looking for a better framework to build e-commerce solutions, and is now the Magento Lead Developer at DemacMedia. He's very passionate about development and technology in general. He is constantly working with new technologies and frameworks.

DemacMedia is one of the Top 10 Global Magento Partners and the largest Gold Partner in Canada; it focuses solely on e-commerce and provides reliable solutions that have been built and battle tested with some of the best merchants on the Web, including a few of the Internet Retailer Top 500.

**Amr Shahin** is a technical architect in vardot (`www.vardot.com`) located in Amman/Jordan, who is an aggressive open source supporter. He has been working with computers since 2003 for both fun and to make a living. Amr's favorite programming language is C, but he currently uses web technologies, mostly PHP.

Recently Amr found a new passion—Magento—which he considers an architecture masterpiece. He is planning to become one of the world's most experienced Magento developers.

Amr has a B.Sc. degree in computer science from Hashemite University, Jordan. In his free time he plays Playstation, drives his car real fast, or does other crazy stuff!

**Marcin Szterling** is a Zend certified and Magento Plus Certified Developer who is passionate about the practical use of modern frameworks in real life. In fact, he is an economist who became a programmer to solve everyday problems. As an active maintainer of the popular Magento search extension Blast Search Lucene, he is well recognized in the Magento community. He started his career in Poland in 1998 and then moved to the UK to work as an IT consultant and active programmer for companies such as Incisive Media Publishing and The Collinsons Group. He was also involved in designing and building e-commerce solutions for brands such as Telegraph, Sainsbury, Paul Smith, or Ben Sherman.

# www.PacktPub.com

## Support files, eBooks, discount offers and more

You might want to visit `www.PacktPub.com` for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version atwww.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.




`http://PacktLib.PacktPub.com`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

## Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

## Free Access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

# Table of Contents


Preface 1

Magento Mobile How-to 7

Installing and configuring Magento Mobile Admin Panel (Must know) 7
Getting acquainted with Magento Mobile Admin Panel (Must know) 10
Creating a separate Store View for mobile application (Must know) 13
Configuring mobile themes for our Magento Mobile Store View (Must know) 15
Preparing and updating product category thumbnail images for mobiles (Must know) 18
Preparing your icons, logos, and other images for mobile applications (Should know) 21
Updating an application's copyright information (Must know) 30
Creating and configuring a basic mobile application (Must know) 31
Styling your mobile application (Must know) 33
Adding static content (Must know) 37
Configuring payment methods (Must know) 41
Configuring social networks integration (Should know) 44
Configuring the Push notification feature (Become an expert) 52
Installing and using the Magento Mobile previewer app (Should know) 54
Enrolling into the iOS Developer Program (Should know) 55
Enrolling into the Android Developer program (Should know) 57
Application submission to Magento Mobile (Must know) 58
Application resubmission and monitoring progress (Should know) 59
Push notification messages administration (Become an expert) 60

# Preface

Dear reader, welcome to the Magento Mobile How-to.

In the last few years, along with the evolution of the smartphone and tablet devices, there has been an increase in the number of visitors using the Internet from their mobile devices. E-commerce websites are no exception, and more and more customers are now using their mobile devices for the purpose of online shopping instead of their desktop computers.

In order to attract more customers on your website and to make sure they visit your site through mobile devices, it is in my opinion a good investment to create a mobile application that will provide you with a full user-experience, and shopping satisfaction to all visitors, both old as well as new.

Assuming that you are a Magento store owner, with some additional investments and a little time spent using Magento® Mobile services, you can build your own branded native storefront mobile application for iOS (iPhone, iPad) or Android based devices.

Throughout this book, in easy and well explained steps, you will learn how to create and configure your own Magento® Mobile application from scratch through the admin panel of your Magento website.

This book does not require any extra programming skills; you just need to spend some time on it and I hope that you will enjoy it as much as I did while writing this book.

## What this book covers

*Installing and configuring Magento Mobile Admin Panel (Must know)*, shows us how to identify to identify our installed Magento version and optionally install the necessary Magento Mobile Admin Panel extension in order to enable Magento® Mobile features for our site, since Magento Mobile support is not natively available on some older Magento versions.

*Getting acquainted with Magento Mobile Admin Panel (Must know)*, shows that it is essential to get familiar with the possibilities available with Magento Mobile Admin Panel in order to better understand the next steps of this book. Also, we are going to look through most of the Admin Panel options and screens.

*Creating a separate Store View for mobile application (Must know)*, will teach us the basic concept of the Website/Store/Store View scopes in Magento and also, how to create a separate Store View for our mobile app in order to separate layout, design, and the other logic from our main store on the web.

*Configuring mobile themes for our Magento Mobile Store View (Must know)*, will help us determine if the mobile theme is already installed on our site and if not, how to install it. We will also learn how to configure Magento to recognize mobile browsers, and apply the mobile theme for these browsers as it is used for products/catalog display in our mobile app.

*Preparing and updating product category thumbnail images for mobiles (Must know)*, will show us that mobile browsers mainly require different image resolutions than desktop browsers, and how to prepare and update separate category images for our mobile application.

*Preparing your icons, logos, and other images for mobile applications (Must know)*, will teach us how to prepare all necessary images and logos for our mobile application and what formats and dimensions they should be in.

*Updating an application's copyright information (Must know)*, will teach us how to update the copyright information for your mobile app through the Magento administration area.

*Creating and configuring a basic mobile application (Must know)*, after all the earlier preparations, in this recipe we are going to learn how to create the basic configuration for our mobile app.

*Styling your mobile application (Must know)*, having already created our first Magento Mobile app, it's time to learn how to design and configure its look and feel.

*Adding static content (Must know)*, will teach us how to add and assign a static content to our app using the Magento admin area.

*Configuring payment methods (Must know)*, shows that in order to have a fully functional mobile store, we need to configure the available payment methods used for a mobile checkout.

*Configuring social networks integration (Should know)*, shows that social networking sites are a very important marketing tool these days, and here we will learn how to configure the implementation for the natively supported social networks, such as Facebook, Twitter, and LinkedIn from inside out in a mobile app.

*Configuring the Push notification feature (Become an expert)*, will demonstrate that the Push notification is a very handy marketing and newsletter feature, and we are going to learn how to configure it so that we can use it with our app.

*Installing and using the Magento Mobile previewer app (Should know)*, will feature the Magento Mobile previewer app, a free mobile application for iOS and Android that is provided by Magento in order to test the created mobile application configuration before publishing it to iStore/Google Play. We are going to learn how to install it and to use it in order to preview the mobile application we are currently building.

*Enrolling into the iOS Developer Program (Should know)*, will teach us how to enroll on the iOS developer program in order to get the necessary credentials for iStore.

*Enrolling into the Android Developer Program (Should know)*, is going to teach us all the necessary steps for enrolling on the Android Developer Program in order to be able to publish our mobile app on Google Play.

*Application submission to Magento Mobile (Must know)*, after the application is configured, it's time to learn all the steps for application submission to Magento Mobile, who will deploy our created app on iStore/Google Play for us.

*Application resubmission and monitoring progress (Should know)*, will teach us how to re-submit our application if we made some core changes to it, and also how to track the progress of our submitted app.

*Push notification messages administration (Become an expert)*, will teach us how to create either a plain text or a rich HTML template for our Push notification messages, and also how to queue the messages and track their progress.

# What you need for this book

In order to be able to follow the recipes in this book, you will need a fully functional Magento web commerce site, or just a test Magento installation for learning purposes.

For preparing images, we can use any image editing software, such as Photoshop, GIMP, and so on.

If you are going to create and publish a real Magento Mobile application, it will be necessary for you to buy a Magento Mobile license for your app. More about Magento Mobile's plans and pricing can be found here: `http://www.magentocommerce.com/product/mobile#mobile-tabs`

A device (smartphone, tablet) with an iOS/Android operating system will be necessary for testing our mobile application with the Magento Mobile previewer app.

# Who this book is for

This book is mainly aimed at merchants – Magento store owners, but also for developers and other curious readers, as it does not require any special knowledge about programming and Magento. By following the simple, practical steps in this book, every interested reader can become a real expert in the planning and configuration of Magento Mobile applications.

## Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "Choose your app targeted platform(s) and pricing plan from the `Plans` and `Pricing` table."

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Check the **I agree to the terms and conditions** checkbox and click on the **Add to Cart** button".



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on `www.packtpub.com` or e-mail `suggest@packtpub.com`.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/support`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

## Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

## Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

# Magento Mobile How-to

Welcome to *Magento Mobile How-to*.

Over the last few years, use of mobile platforms has expanded quickly in every part of modern society. The Magento e-commerce platform is no exception and Magento store owners are aware that they will get more customers at their stores inside the mobile users community.

It doesn't matter if you are a Magento store owner, a developer, or just a curious reader, this book will provide you with useful information on how you can create your own Magento Mobile application, through which you can reach out to more customers, who are the mobile platform users.

## Installing and configuring Magento Mobile Admin Panel (Must know)

Depending on the Magento version on which the website runs, it may be necessary to install and set up the Magento Mobile Admin Panel extension in order to be able to start creating your own Magento Mobile application.

Magento Enterprise and Professional versions 1.10 and higher and Community Edition 1.5 and higher already have this extension built-in, so extra installation is not necessary.

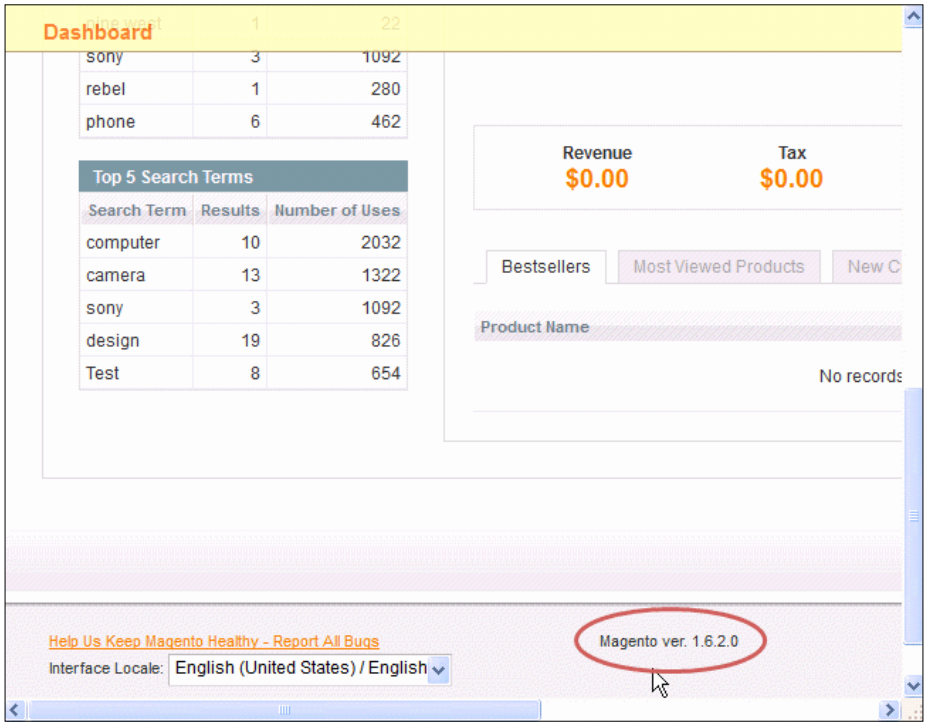
In this task, we will first determine if the Magento Mobile Admin Panel extension is already included in our Magento installation and then explain how to install it if it is not present.

### How to do it...

1. Log in to **Magento Admin Panel** and determine the Magento version:

   Visit `http://yourstore.com/admin` and log in with your existing admin credentials.

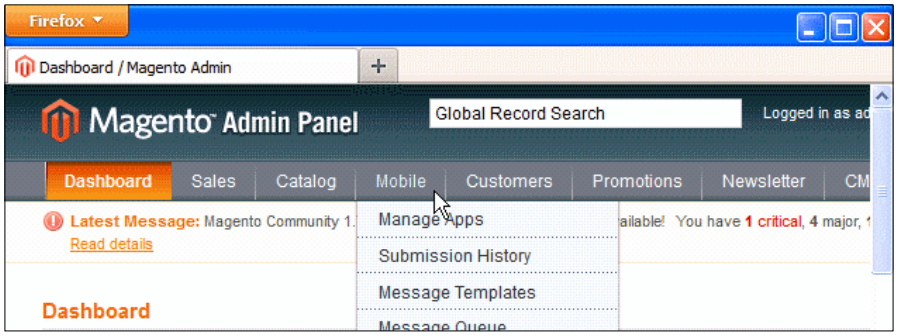
After successful login, on the bottom of Magento's **Dashboard** panel, we can see the currently installed Magento version. For example, **Magento ver. 1.6.2.0** or similar as shown in the following screenshot:




If the installed Magento version is Professional or Enterprise, it will be written near the top left Magento logo image; otherwise, Community edition is installed.

2. Check existence of the Magento Mobile extension:

   We can find out if Magento Mobile Admin Panel is already installed, looking at the main horizontal menu—there should be a  menu item named **Mobile**:



3. If the menu **Mobile** is not present, you will need to install this extension by following the next step.
4. Install the Magento Mobile extension, if it is not installed:
    1. Register the account and log in to Magento Connect:

        Open your favorite web browser and go to `http://www.magentocommerce.com/magento-connect/` in order to download the Magento Mobile extension.

        We can find the link **LOG IN** in the upper-right corner of the screen.

        When we click on **LOG IN**, a pop-up form will be shown offering fields for logging in as an existing user and the **Register** button for a new user.

        If you do not have an account on Magento Connect already, just click on **Register** and on Register form, fill in and submit the required information.

    2. Get the extension key from Magento connect:

        Open the Magento mobile extension link, `http://www.magentocommerce.com/magento-connect/magento-mobile-6497.html` in your browser.

        Click on the **Install now** button and choose an appropriate option for your installation:

        - **Magento Connect 1.0**: For Magento versions older then Professional/ Enterprise 1.9 and for Magento CE older then 1.4.2.0 RC1
        - **Magento Connect 2.0**: For other Magento versions

        Check the **I agree to the extension license agreement** field and click on the **Get Extension Key** button.

        Next, click on the **Select Key** button and copy the key in the clipboard.



3. Install the extension through **Magento Connect Manager**:

   Log in to your **Magento Admin Panel** and in the main menu, click on **System | Magento Connect | Magento Connect Manager**.

   The Magento Connect Manager **Log in** screen will be shown, where you need to enter your Admin credentials in order to log in.

   After you log in, just paste the key from the clipboard into the provided input box and click on the **Install** button.

   The extension will be automatically installed and ready to use.

## How it works...

The installed Magento Mobile Admin Panel provides a nice user interface in the Magento admin area for creating your own mobile applications through Magento Mobile. On newer Magento versions, it is included in the Magento installation by default.

Magento Connect is Magento's official place for all kinds of Magento extensions, and Magento Connect Manager is used to make the extension install process easy and accessible to all users, not only to developers.

## There's more...

It's also possible to install this extension manually, and more detailed directions are described on the Magento Connect extension's overview page.

If installing manually, we have to choose a proper extension version (depending on the Magento installation version) for download, and compatible versions and download links are also provided on the Magento Connect extension's overview page.

Of course, in the case of manual installation, you must have at least **FTP (File Transfer Protocol**) access to your server's Magento installation in order to manually copy the extension files there.

# Getting acquainted with Magento Mobile Admin Panel (Must know)

In order to start creating our own Magento Mobile application, it is necessary to get familiar with the possibilities available with Magento Mobile Admin Panel and find out what we can do with it.

Here we will learn how to work with Magento Mobile Admin Panel and what configuration options it provides.

## Getting ready

First of all, we need to log in to our Magento installation admin area by typing the URL `http://ourmagentostore.com/admin`.

On Magento's dashboard menu, we can find a menu item named **Mobile**.

From there we can access all available options for configuring our application.

## How to do it

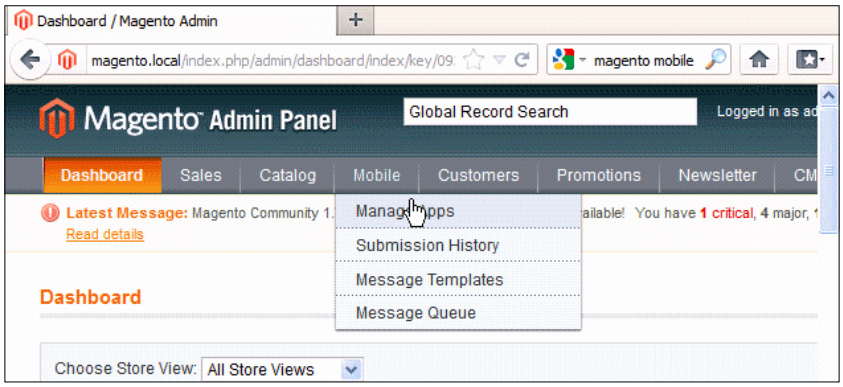
Let's look at what we have on this menu:

1. Click on the menu item named **Mobile** to expand it and you will see all the submenu options.

   We can see that we have four submenu items shown:

   - **Manage Apps**
   - **Submission History**
   - **Message Templates**
   - **Message Queue**

2. Click on the **Manage Apps** option and a screen with a grid will be shown.
3. This is the screen where created apps (if any) are shown. Here we have columns with input boxes for filtering the possible list of created applications.
4. We haven't created any application yet, so no item will be listed on the screen.




5. Click on the **Add App** button on the top-right of the screen to open the **New App** page. This is the page from where we will start creating our first app.
6. Let's now click on a menu item named **Mobile** again and then choose **Submission History** from the drop-down menu.

The same kind of grid with filters is shown. On this screen, **App Submission History** will be shown (if there is any).



Whenever we make some change in our application, the new version needs to be resubmitted to App Store/Android Market and all this version history will be shown in the **Submission History** screen.

7. Now click on the **Mobile** menu item again and then click on **Message Templates**. The next screen will show our push-notification message templates.
8. Let's click on the menu item **Mobile** again and then choose **Message Queue** from the drop-down menu. This is the screen where the queue will be shown for all our push-notification messages.

## How it works...

**Manage Apps** is the main area where we will spend most of our time while creating the Magento Mobile application. From there we will get all the other options and possibilities for creating, configuring, designing, and preparing our mobile application for submission.

When our app is submitted to Magento Mobile, they submit our application depending on the device version to App Store/Android Market and from the **Submission History** screen we can see all submitted versions of the application. We can also use filtering possibilities to easily find what we are looking for.

Message templates are pre-defined messages that we can send as push-notifications to our mobile application users and **Message Queue** is an area where we can track the progress of our push notifications.

## There's more...

For the basic Magento Mobile application creation process, only the **Manage Apps** area is necessary. Whether we should use other areas or not, depends on what additional capabilities we want to use.

### We will also need...

After finishing our mobile application creation we will need to purchase a paid Magento Mobile license in order to submit our application.

### If we want to use push notifications...

Using push-notification messages is optional, and in that case we will need to register an account with the Magento partner—Urban Airship—for a paid license. They are also providing a 60-day free trial license.

# Creating a separate Store View for mobile application (Must know)

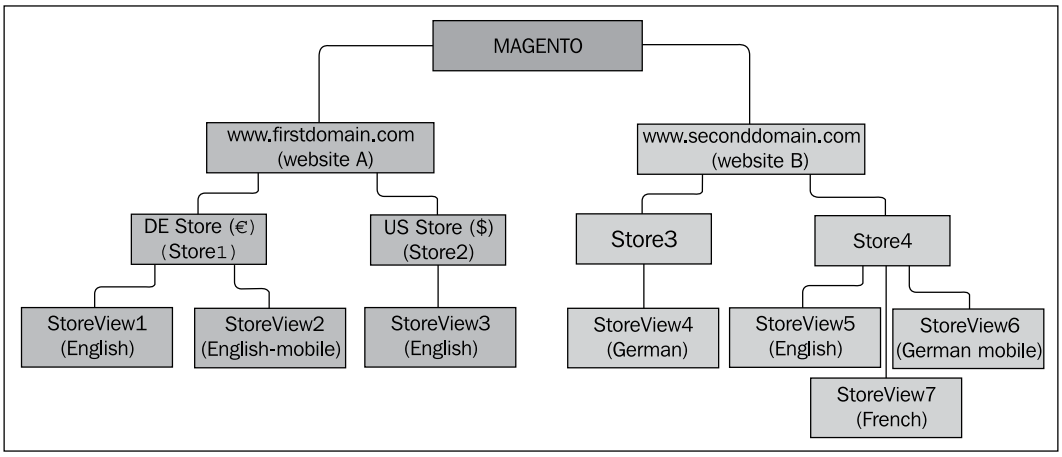
Here we are going to create a separate Magento Store View for our mobile application.

Of course, we have an option to use existing Store View, but it's much better to create separate one for mobile application, which does not depend on changes made to layout, design, and so on of the main website Store Views.

Also this way the orders created via mobile application are separated from the website's orders and it's much easier to track or filter them later.

Before we continue, let me first explain the basic concept of the Website/Store/Store View features in Magento:

- We can manage multiple Websites from one Magento installation
- These Websites can be defined for different domain names (for example, `www.firstdomain.com`, `www.seconddomain.com`)
- Each Website can have one or more Stores defined
- Inside each Store, we can define one or more Store Views




- This concept is implemented in Magento through different levels of possible configuration values, called **Scopes**

Configuration Scopes in Magento:

- **Global**: Configuration level that applies to entire installation.
- **Website**: Website is the parent of the Store. Different Websites can be defined on different domain names. Each Website consists of one or more Stores. Websites can share the same customer data between them, depending on the configuration we choose.
- **Store (group of Store Views)**: Stores are a Website's children level of configuration. Categories and products are managed at this level, which means that we can define a different root category for each Store and each store can have completely different catalog structure.
- **Store View**: Store Views are children configuration levels of the Stores. Every Store needs to have one or more Store Views defined to be browsable in the frontend of Website. The catalog structure of each Store View inside one Store will be the same. Each Store View allows different presentation of the data in the frontend (for example, different language, different theme, and so on).

## How to do it...

1. Log in to Magento admin area.
2. In top menu of the the Magento admin, click on **System** and then choose the **Manage Stores** menu option.
3. Here a list of Websites, Stores, and Store Views that are currently present in your Magento installation will be shown. Click on the **Create Store View** button in the top-right corner of the screen in order to create a new Store View.
4. In the **New Store View** screen, we have to choose the Website/Store group in which our Store View will be created.
5. Under the **Name** field, enter the name of new Store View for our mobile application. For example, English mobile.
6. **Field Code** must be filled with a unique string (for example: `my_mobile_store`) that will be used by Magento to uniquely identify the created Store View.
7. Next, we want to set the **Status** field to the **enabled** option in order to make it active for our mobile application.
8. We can leave the **Sort Order** field empty as it is not essential for our current needs. Click on the **Save Store View** button and our new Store View will be displayed on the **Manage Stores** list.

## How it works...

In Magento, we can define multiple Websites, multiple Stores, and multiple Store Views. Stores are basically virtual groups of Store Views.

All Store Views under one Store share the same catalog products. This is important because when we create a new Store View and set it to be under our current website store, all categories that are defined in our main website store will automatically be inside our mobile Store View too.

If we make some global product/category change under the main Store, it will also be reflected on our mobile Store View.

## There's more...

In Magento, we can also change the categories and products on a Store View level and it will not affect other Store Views under that Store. So if we want to display some product in a different way, add a different description or price, or change some other attribute for just the mobile Store View and you will find that it has not affected the other Store Views under that Website; it is very easy to implement. That is the main reason for creating a separate Store View for mobile applications.

### Did you know that...

Store Views inside the same Store also share the same customers, so the customers that are registered on our website can easily buy from their mobile devices on our mobile Store View with the same user credentials.

# Configuring mobile themes for our Magento Mobile Store View (Must know)

Considering that the products in our mobile application are basically shown in the mobile browser that is integrated inside the iPhone/iPad/Android application, we need to use a separate theme for our app in order to display categories and products correctly on targeted devices/operating systems.

We are going to configure a mobile theme inside Magento to be used for our mobile Store View.

## Getting ready

Since I am using Magento 1.6 CE, while writing this book, the default Magento Mobile theme is already included with the Magento installation and we just have to configure it to be displayed when the website is shown on mobile device browsers and of course inside our app.

We can find out if we already have the mobile theme included; if we browse our Magento installation source files and if we find the `iphone` folder inside `App/Design/Frontend/Default`, we already have the mobile theme installed and we are good to go.



For older Magento versions that don't have a mobile theme included in the installation files, it is necessary to download and install mCommerce - iPhone Optimized Theme from Magento Connect (`http://www.magentocommerce.com/magento-connect/mcommerce-iphone-optimized-theme.html`).

We have to be logged into our Magento admin area first.

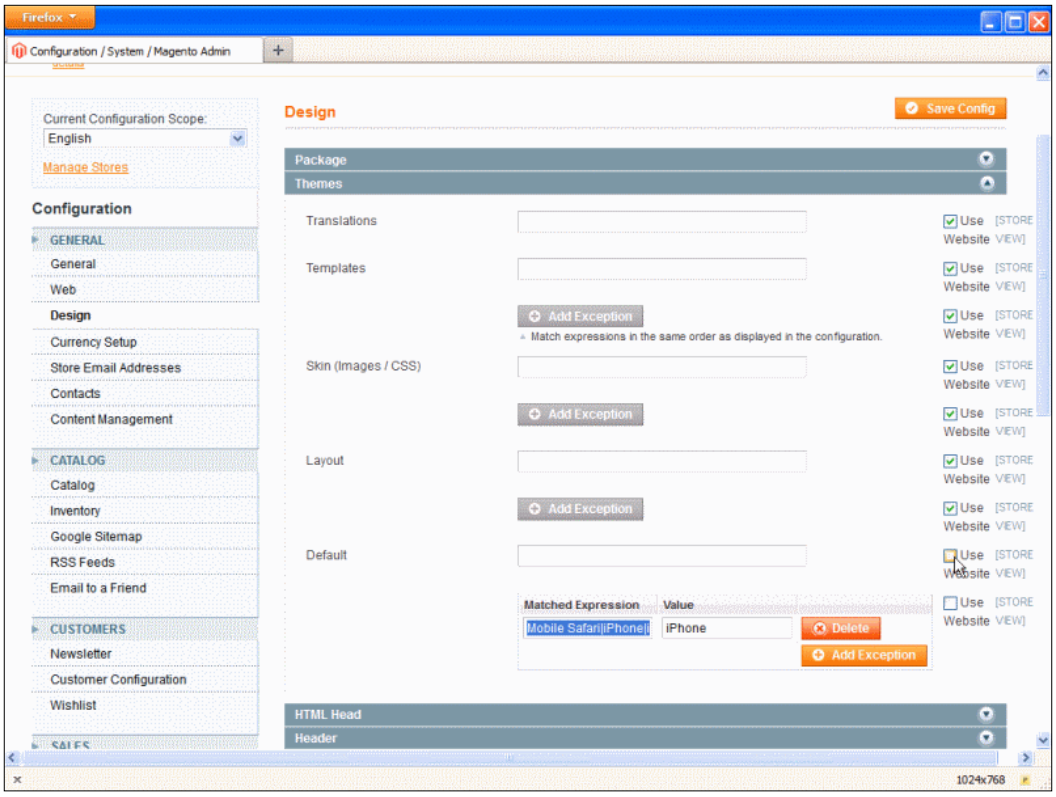
## How to do it...

1. In the Magento **Dashboard** main menu, navigate through **System | Configuration**. Magento's main configuration screen is shown with tabs listed on the left-hand side of the screen.
2. First, let's choose which Store View we want to apply this mobile theme configuration to:
    a. In the top-left corner of the screen, there is a list box with the label **Current Configuration Scope**. Let's choose our newly created Store View from the list in order to apply configuration only for it.
    b. That way, the other Store Views will not be affected with our configuration values.
3. Now locate the **Design** tab on the left and click on it in order to access the necessary configuration tabs.
4. In the middle of the screen, click on the **Themes** row to expand it.

    Locate the input box labeled **Default** under the **Themes** tab.

    Under the input box, there is a button named **Add Exception**. In order to press it, first uncheck the **Use Website** checkbox located to the right of that button.

    Finally, press the button **Add Exception**, and the **Matched Expression** and **Value** fields will be shown.
5. Inside the **Matched Expression** field, enter `Mobile Safari|iPhone|iPad` and inside the **Value** field, enter `iPhone`.
6. Next, click on the **Save Config** button in the top-right corner and we are done!




## How it works...

A mobile theme is specially designed HTML/CSS website theme to be used with mobile browsers, because of the difference in screen sizes and resolutions, and so on.

Without a mobile theme, if we want to open our Magento website on a mobile device, we have to scroll right to left to see the whole website screen, and the mobile theme resizes it to fit the screen.

Magento by default uses its default theme for all User-Agents (browsers) that visit the site.

When connecting to some website, the server has access to our browser type and some other information.

There are three basic possible variations of default browsers on mobile devices—browser on Android, on iOS (iPhone, iPad)—and when we connect with them, the server will read the information, called the User-Agent, and recognize the browser to which we are connected.

This helps us to tell Magento the following:

> *Dear Magento, for the User-Agent—Mobile Safari or iPhone or iPad—please make an exception and do not use the default website theme, but rather the theme we specified under the Value field (we specified iPhone theme which is also Android OS compatible).*

## There's more...

We configured a mobile theme to be used for just one Store View (our Store View created for a mobile app), but if we want, we can set those rules to be applied in the Default Scope, which means that these rules will be applied by default for all defined Store Views in our Magento installation.

To accomplish this, we need to choose **Default** in the list box with the label **Current Configuration Scope**, before configuring the mobile theme exceptions.

# Preparing and updating product category thumbnail images for mobiles (Must know)

For a beautiful look and feel to our mobile application's catalog, it is necessary to add thumbnail images for main categories.

We are going to prepare images for each of the top-level categories and update categories to use those images in our mobile app.

## Getting ready

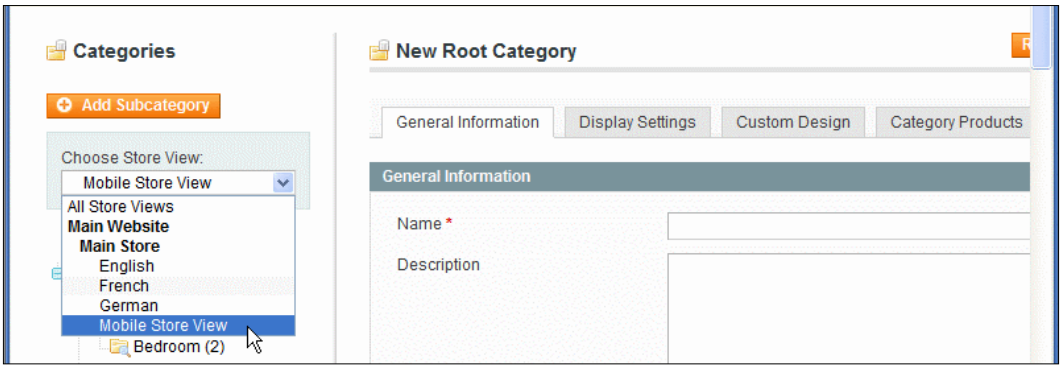
It doesn't matter if we want to use the existing category images, or prepare new images, first of all, we have to resize/crop them to 80px x 80px at 72 dpi in order to be displayed properly on targeted mobile devices. If we don't resize them, they will break the layout of our mobile application and the category listing will not look very nice to the mobile users.

We can use any image-editing program to resize/crop them as we need.

When we resize our images and save them in `png` or `jpg` format, we can upload and assign them to our product categories.

## How to do it...

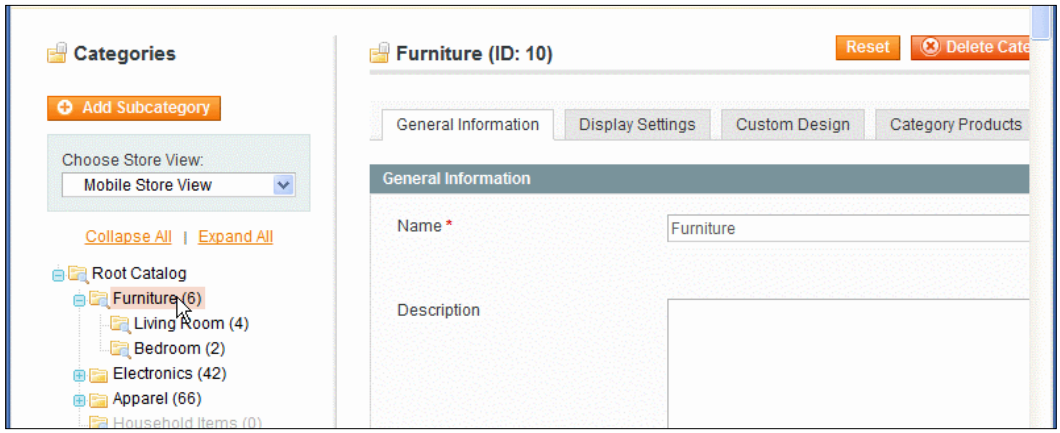
1. Log in to Magento Admin Panel and navigate through **Catalog | Manage Categories**.
2. Locate the **Choose Store View** list located in the left column above the **Categories** tree and select our Store View that we defined for our mobile application.





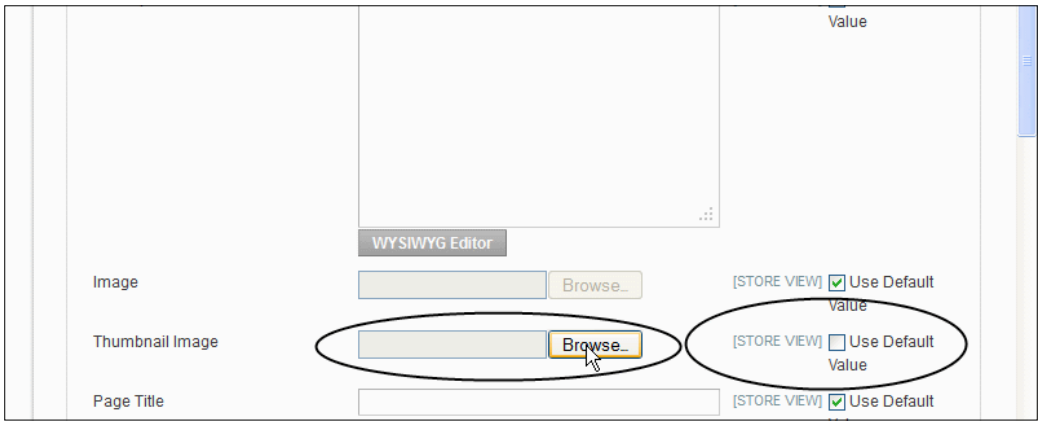

Important: Do not apply any changes on All Store Views, but rather in the separate Store View that we created, in order to avoid messing up the website's settings that are not connected to our mobile application!

3. On the left-hand side of the screen, click on the category that needs to be updated inside the tree.




4. On the right-hand side of the screen, you can see the **General information** tab. There we can see some information about the selected category.

5. Find the field labeled **Thumbnail Image** and uncheck the **Use Default Value** checkbox on the right. This way the **Browse...** button will appear enabled.






If this category already has an image set on some other scope (All Store Views), you may have to check the **Delete Image** checkbox and click on the **Save Category** button in order to make the **Browse...** button appear enabled.

This will not delete the image set in other Store View scopes but just in those selected, which is just what we need.

6. Click on the **Browse...** button in order to upload the prepared image for the selected category.
7. After uploading is finished, just click on the **Save Category** button in the top-right corner of the screen.
8. The category is now saved and a new thumbnail image is assigned to it.
9. Repeat step 3 for each category that is the first child of the **Root Catalog** category.

## How it works

When the thumbnail image is uploaded, it is assigned to the edited category and our mobile application uses that image when the category screen is shown.

It is important to use the exactly defined dimension of images (80px x 80px at 72 dpi) in order for them to be displayed properly in our mobile app.

## There's more...

If we are going to use the same images like we have already defined as the main image for a specific category, it may be easiest to first download the category images from the main website and then to resize and upload them as thumbnail images for each category.

# Preparing your icons, logos, and other images for mobile applications (Should know)

In order to make a complete mobile application, we also need some icons and logos to be prepared before submitting our application to Magento Mobile.

In this recipe we will find out what images we need to prepare and what formats and dimensions they should be in.

## Getting ready

It doesn't matter whether we hire a designer for the Creative Assets preparation or do it by ourselves; it is important to follow some rules in order to do the job right.

First we need to get some image editor software, so we can resize and crop our images, and save them in the proper image format.

There are different image editors out there—depending on the operating system supported. For example, one of the best editors is Adobe Photoshop, which has versions for Windows and Mac OS X, but it is not possible to run it on Linux-based systems and also it is commercial software that requires license.

Luckily, there are plenty of open source, free editors that can be downloaded from the Internet (Paint.NET, just for Windows OS and GIMP, available for either Windows, Mac OSX, Linux, and so on).

Since I am personally a fan of open source software that is available for multiple platforms, I will continue this chapter showing you how to prepare images using the GIMP image editor.

Let's first visit the GIMP website and download and install GIMP on our operating system. Visit `http://www.gimp.org/` and download the newest version of the application for your operating system. Install it on your OS, following the installation instructions found on the website, and then you are ready to start.



The best format to use for our creative assets is `.png` because iOS and Android operating systems are built with included optimization possibilities for the `.png` file format, while other format optimization is not so well supported.

Of course, depending on the targeted OS/mobile devices, we need several dimensions of each image, for each device type (iPhone, iPad, Android-based devices).

For each Magento Mobile application type that we create (iPhone, iPad, Android) we need to prepare eight different Creative Asset types:

- **Large app icon** (this will usually be your logo and brand): A large app icon will be shown in the App Store and also on Google Play (Android apps marketplace) as the main image for our application page. It will not be shown within our mobile application.
- **Custom app icon**: This is our mobile application icon and it is shown in our mobile device menu when we install the application.



It needs to be attractive in order to attract the user's attention while they navigate through the installed apps on the device. With a more attractive icon, there is more chance that the user will start our application again and again and possibly place some new orders in our store.

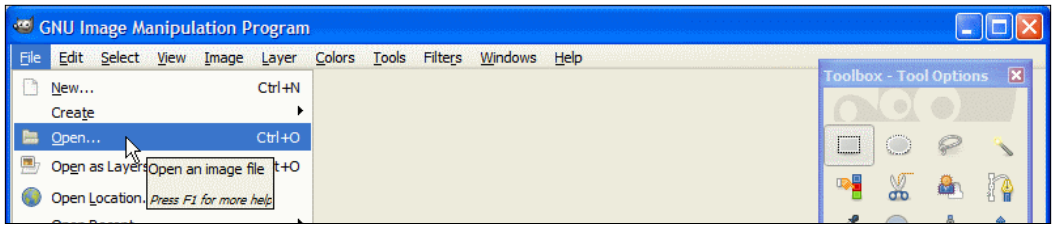
- **Loading splash screen**: It will be shown as full screen for a few seconds, while our mobile application starts.
- **Small header logo**: It is shown inside each screen of our mobile application—on the app title bar.
- **Banner image**: In our application, we have the main home screen. The banner image is shown there. It should also represent our logo/brand.
- **Copyright logo**: As the name says, it is logo for our copyright information. It will be shown on our mobile app's copyright page.
- **App background**: The image  used as background in our app.
- **Top level category thumbnails**: We have already uploaded our category images for our Store View in a previous recipe, but if we are targeting multiple device types and creating separate store views for iPad, we will need to prepare different dimensions for iPad.

In the next steps, I will show you how to prepare a large app icon image for Android and for the rest of the images I am going to provide a table with targeted devices and necessary dimensions for assets, so you will use the same steps to prepare other necessary images.

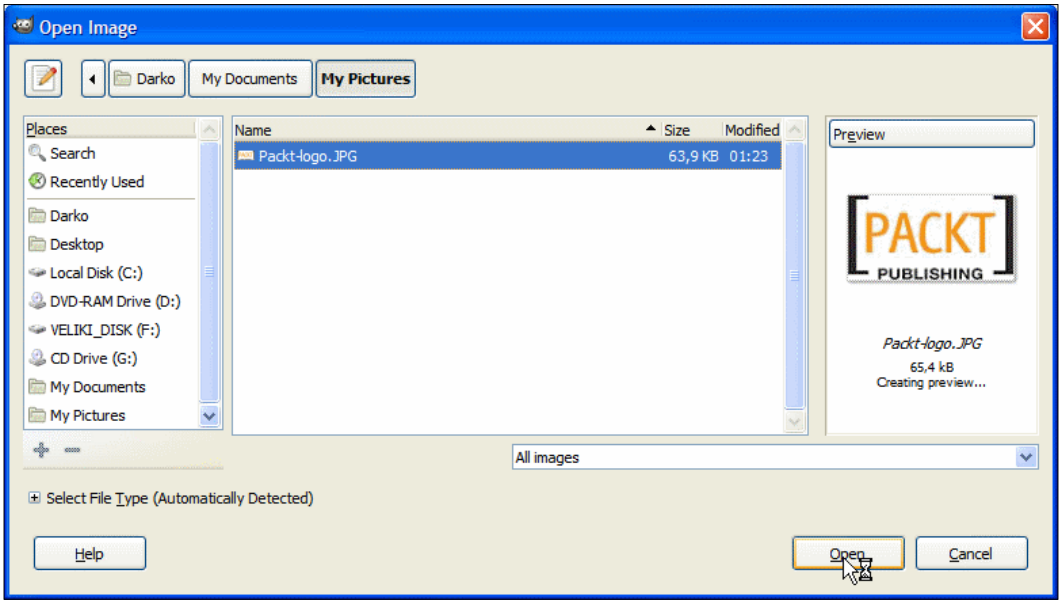
## How to do it...

Follow these steps for creating a large app icon image:

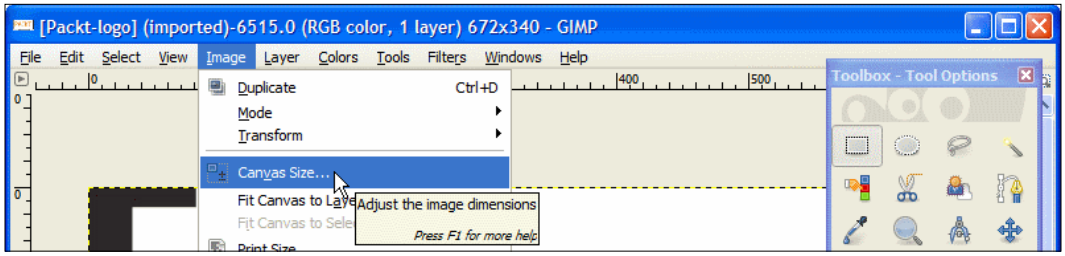
1. Start the GIMP image editor and navigate to the **File | Open** menu action:



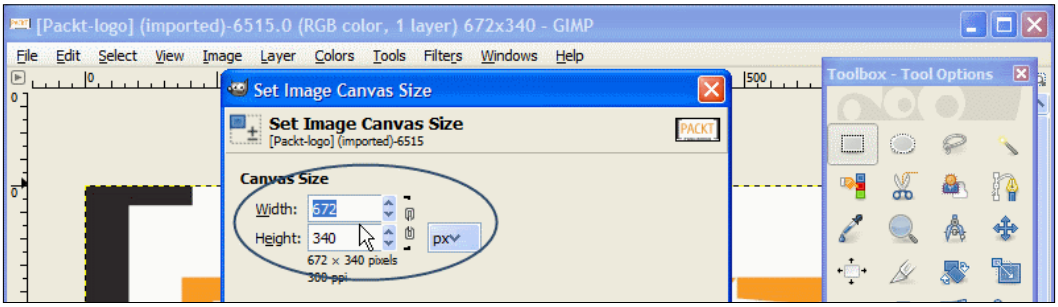

2. Choose the image file for editing from your filesystem and click on the **Open** button. The image will be loaded inside the editor:




3. Navigate to **Image | Canvas Size** to see the current image dimensions:



We can see that our current image is 672px x 340px. For a large app icon, we need the dimensions to be 512px x 512px.



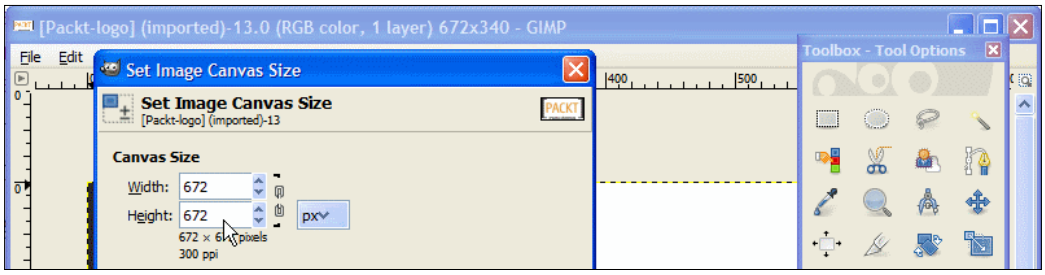

What does that mean to us? Our image height is smaller than needed, so we have two options to make it 512px in height without losing the proportions of the image:

- Set the image canvas to be bigger in height and then proportionally resize the image
- Crop the image to the same width as the current height and then resize proportionally to the necessary dimensions

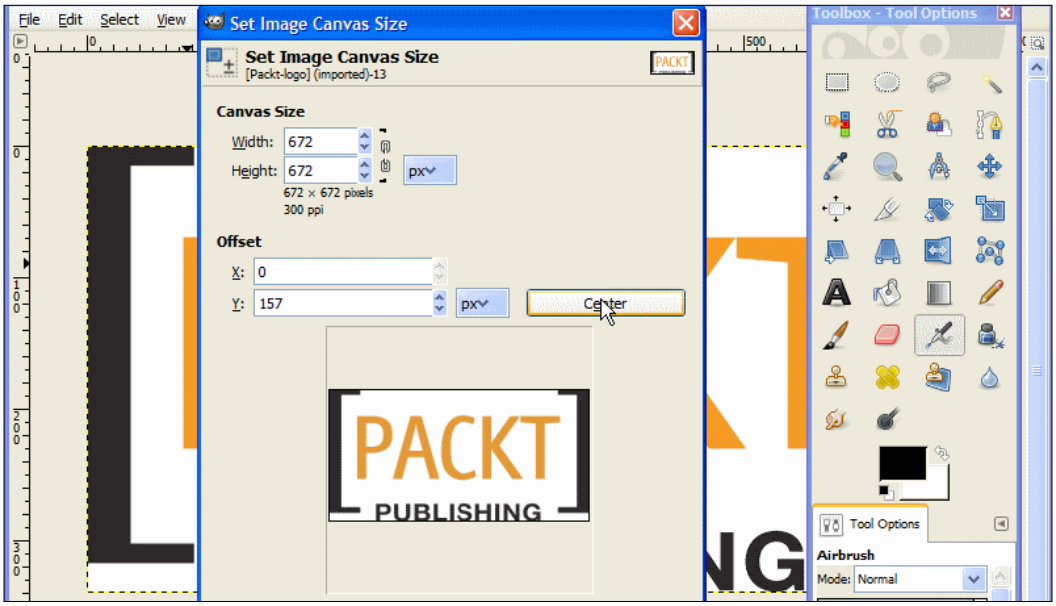


In this particular case, I would strongly suggest setting the image canvas dimension to a bigger value first and then resizing the image, because this way we avoid resizing the image from a smaller dimension to a bigger one and losing the image quality.

4. Since the large app icon needs to have equal width and height in dimensions, let's make the canvas height the same as the width, so enter `672`px as the canvas's **Height** dimension and press *Enter* on the keyboard:



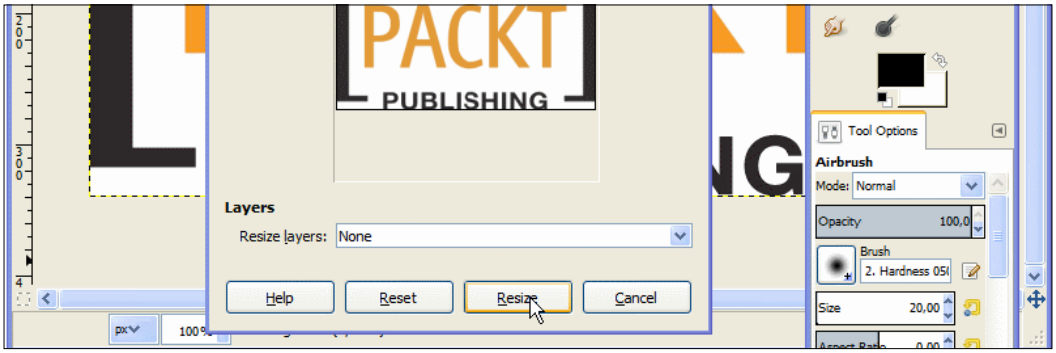

5. Since our canvas height is the now bigger than the height of current edited image, I want to center the image vertically on the canvas. In order to do that, just press the **Center** button as shown in the following screenshot:



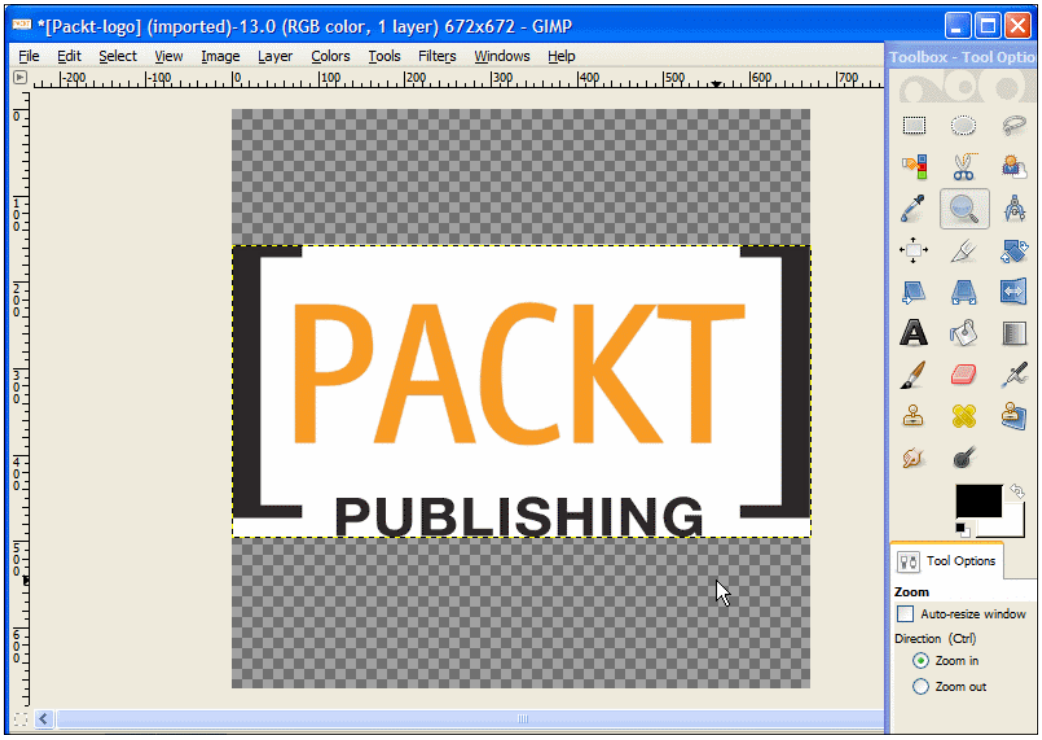



Of course, you have several possibilities to properly position your current image on a bigger canvas. You can manually enter the **X** and **Y** offset for the image position inside the canvas, or you can drag-and-drop the image to set its position inside the preview image box. Feel free to play with the options until you are satisfied with the layout preview.

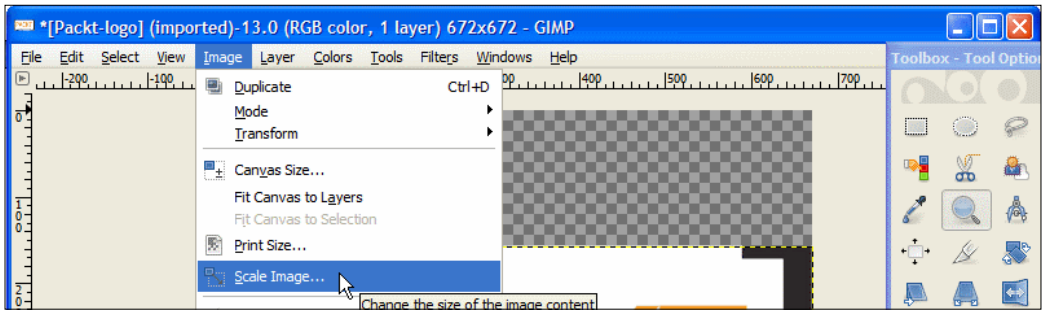
6. When finished positioning, just click on the **Resize** button on the bottom of the form:



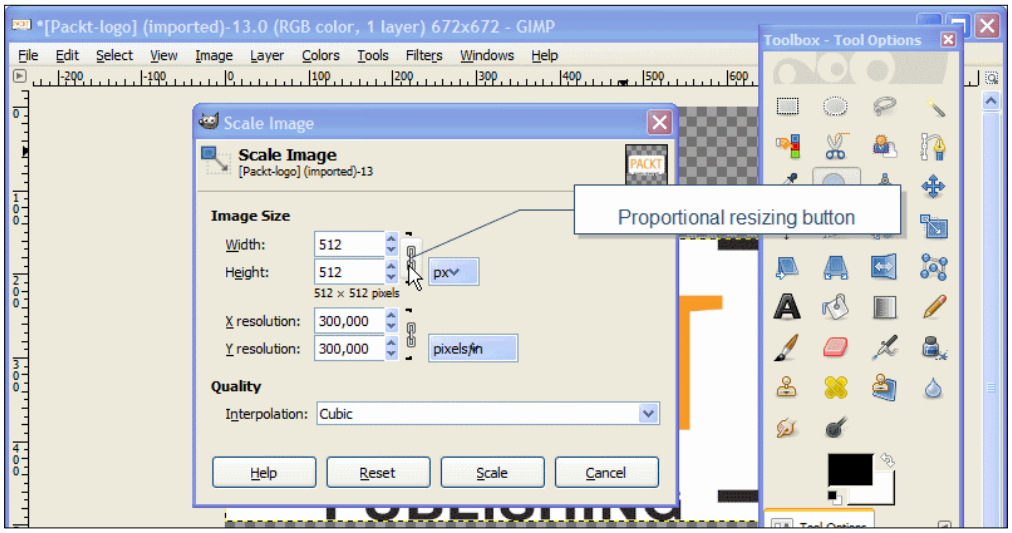
Inside the editor window, our current image should look like the one shown in the following screenshot:



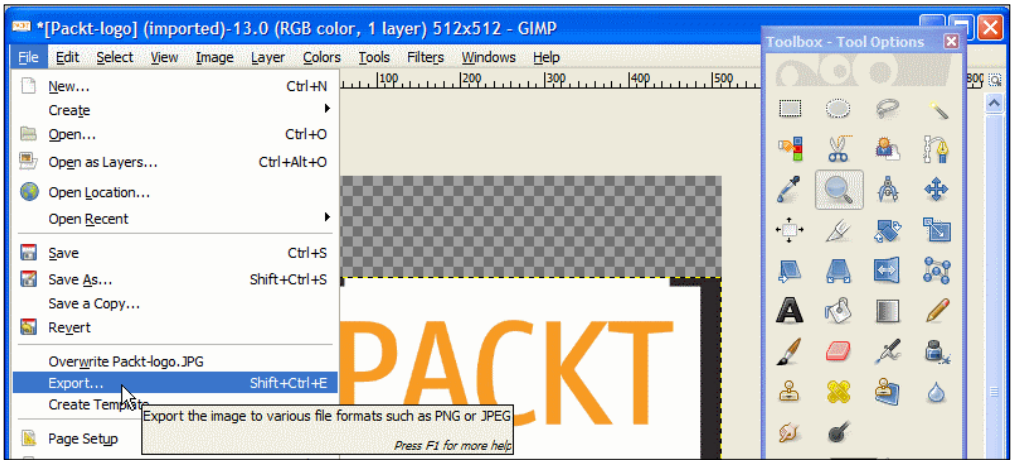

7. Now we need to resize the image to the required dimensions (currently it is 512px x 512px). Navigate to **Image | Scale Image**:



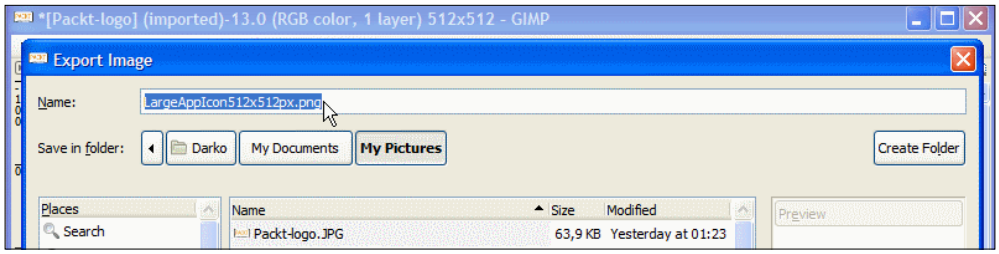

8. First make sure that **Proportional resizing** is turned on, and then insert the `512` value inside the **Width** field. Since we want the image to be optimized for the Web, set the **X** and **Y** resolution values to be 72 pixels/in and click on the **Scale** button to resize them, as shown in the following screenshot:



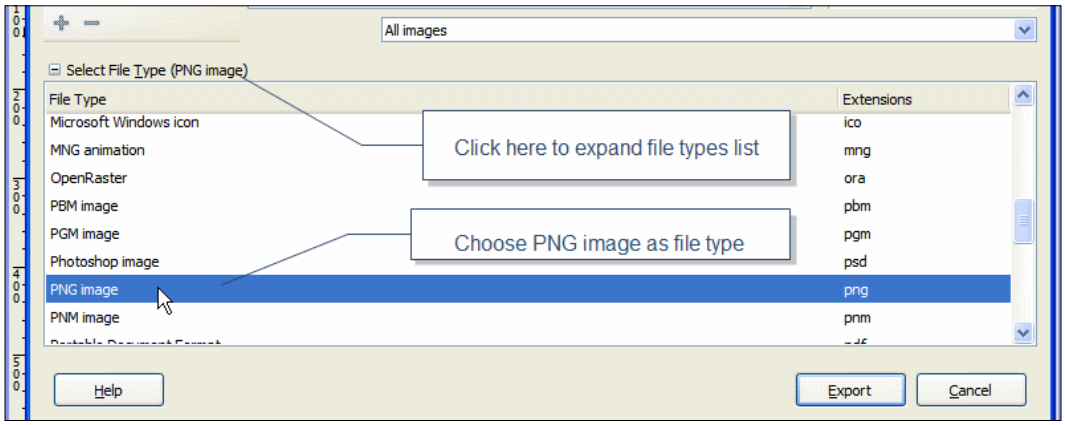

9. Navigate to the **File | Export** menu item:



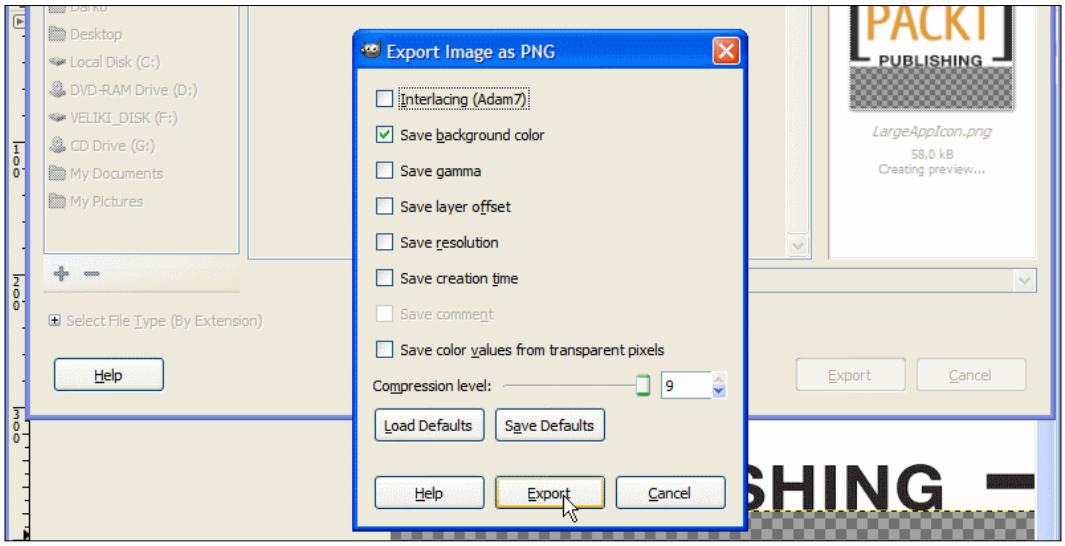

10. Insert the filename for new resized image (for example, `LargeAppIcon512x512px`):



11. Expand the **Select File Type** list, choose **PNG image** from the list, and click on the **Export** button:




12. The **Export Image as PNG** dialog is shown in the following screenshot and we need to set some options for the exported PNG image. Leave all checkboxes unchecked; just check **Save background color** and click on **Export**:






Leaving the options as shown in the preceding screenshot, we are saving the image without transparency, which is in fact just what we need because transparent images will not be accepted by the App Store and Google Play.

We learned how to prepare a large app icon, and for other necessary images just repeat the previous steps, using the dimensions provided in the following table:

| | iPhone 3 | iPhone 4 | iPad | Android based devices |
|---|---|---|---|---|
| **Large app icon** | 512px x 512px | 512px x 512px | 512px x 512px | 512px x 512px |
| **Custom app icon** | 57px x 57px | 114px x 114px | 72px x 72px | 48px x 48px |
| **Loading splash screen** | 320px x 460px | 640px x 920px | 768px x 1004px | 320px x 455px |
| **Small header logo** | 35px x 35px | 70px x 70px | 35px x 35px | 35px x 35px |
| **Banner image** | 320px x 230px | 640px x 460px | 768px x 294px | 320px x 258px |
| **Copyright logo** | 100px x 100px | 200px x 200px | 100px x 100px | 100px x 100px |
| **App background** | 320px x 367px | 640px x 734px | Landscape: 1024px x 704px, Portrait: 768px x 960px | 320px x 455px |
| **Top-level category thumbnails** | 80px x 80px | 80px x 80px | 243px x 243px | 80px x 80px |

## How it works...

We will have to upload some of those images inside our Magento Mobile Admin Panel as required; some of them we will upload as category thumbnail images, and the rest of them we will have to send to Magento Connect by e-mail so that they can prepare our application for submission on App Store/Google play.

## There's more...

We don't need to prepare all dimensions of images at once, but rather prepare images for currently targeted devices. As we have already opened our image editor, we should start with preparing the largest dimension of the image and then resize it to smaller dimensions and just save each version, so we can use them later for other targeted devices.

# Updating an application's copyright information (Must know)

As with any other software product, website, or application, our app also has to have copyright information included. We will learn how to update copyright information that will be visible inside our app.

## How to do it...

1. Log in to your Magento Backend.
2. Navigate to **System | Configuration** in the main menu.
3. Next, we have to select the proper Store View—the one that we have already defined for our mobile application.
4. In the top-left corner of the screen, locate the **Current Configuration Scope** list and make sure that our earlier defined Mobile Store View is selected.



When configuring anything in Magento, always look at the **Current Configuration Scope** list and switch to the appropriate one!

If we define our copyright information on some other Store View or on Default Scope, we will mess up our Website Copyright information!

5. Click on the **Design** tab on the left side of the screen.
6. In the middle of the screen, click on the **Footer** row to expand it.
7. Locate the **Copyright** input box and enter some short copyright information (for example, **Copyright by My Company, 2012**).
8. Make sure to uncheck the **Use Website** checkbox on the right, or Magento will use the default setting instead of ours.
9. Click on the **Save Config** button in the top-right corner of the screen and we are done.

## How it works

When our mobile application loads on our customer's devices, it connects to our main website and reads the configuration from there, and anytime when we update the copyright information for the mobile Store View, it will also be updated inside all our mobile applications connected to our site.

## There's more...

After changing any of the settings in Magento, make sure to refresh website's cache inside the admin area. Navigate to the **System | Configuration | Cache Management** section in order to be sure that the configuration is properly updated.

From this screen, we have several options available for Cache Management:

- **Refresh** (single or multiple cache types): This operation will delete and recreate the cache files.
- **Flush Magento Cache**: This operation will delete the Magento cache files from the `var/cache` folder.
- **Flush Cache Storage**: This operation will do the same as the previous one, with the difference that it will also clear your `optcode` cache.

In Magento we have several different cache types:

- Configuration
- Layouts
- Blocks HTML output
- Translations
- Collections Data
- EAV types and attributes
- Web services configuration

You can find more about those cache types on the official Magento wiki page at `http://www.magentocommerce.com/wiki/modules_reference/english/mage_adminhtml/system_cache/index`.

For the purpose of copyright information, it is enough to **Refresh** the cache for **Blocks HTML output**, but there is no issue if we refresh all of these cache types.

# Creating and configuring a basic mobile application (Must know)

Finally, with everything configured and prepared, it is time to start exploring Magento Mobile Admin Panel in order to create our first Magento Mobile application.

## How to do it...

Follow these steps:

1. Inside your **Magento Admin Panel**, navigate to **Mobile | Manage Apps** on the main menu.
2. Click on the **Add App** button in the top-right corner. The **New App** screen will be shown.
3. Since we have to create a separate application for each mobile device type, let's choose our first targeted platform.

   Under the **Device Type** list, we can choose **iPad**, **iPhone**, or **Android**.

   For the purpose of this recipe, since the procedure is almost the same for all device types, I will choose **Android**.
4. After choosing the desired **Device Type**, click on the **Continue** button, and click on the **General** tab under **Manage Mobile App**.
5. First we have to fill in the box named **App Name**. Choose an appropriate name for your mobile application and insert it there.
6. Under the **Store View** list, make sure to choose our earlier defined Store View with updated mobile theme exceptions, our mobile copyright information, and category thumbnail images.
7. Set the **Catalog Only App** option to **No**.
8. Click on the **Save and Continue Edit** button in the top-right corner of the screen.
9. Now you will notice a warning message from Magento that says something like the following:

   **Please upload an image for "Logo in Header" field from Design Tab.**

   **Please upload an image for "Banner on Home Screen" field from Design Tab.**

   Don't worry, Magento expects us to add some basic images that we prepared for our mobile app. So let's add them.
10. Click on the **Design** tab on the left-hand side of the screen.
11. Locate the **Logo in Header** label and click on the **Browse...** button on the right to upload the prepared small header logo image. Make sure to upload the image with proper dimensions for the selected device type (**iPhone**, **iPad**, or **Android**).
12. In the same way, click on the **Browse...** button on the right of the **Banner on Home Screen** label and choose the appropriate prepared and resized banner image.
13. Now, let's click on the **Save and Continue Edit** button in order to save our settings.

## How it works

For each device type, we will have to create a new Magento Mobile application in our Magento Mobile Admin Panel.

When we once select **Device Type** and click on the **Save** button, we are unable to change **Device Type** later for that application.

If we have chosen the wrong **Device Type**, the only solution is to delete this app and to create a new one with the proper settings.

The same applies with our chosen Store View when configuring new app.

## There's more...

When our configuration is saved for the first time, auto-generated App Code will appear on the screen and that will be the code which will uniquely identify our Device Type— the assigned application to be properly recognized with Magento Mobile.

For example, defand1 means that this application is the first defined application for the default Store View targeted on android (def = default store view, and=android).

### How to use mobile application as catalog only

Under step 7 we set **Catalog Only App** to **No**, but sometimes, if we don't need checkout and payment in our mobile app, but we want to use it just as catalog to show products to our mobile customers, we just need to set the **Catalog Only** option to **Yes**.

# Styling your mobile application (Must know)

To define the look and feel of our mobile app, it is necessary to take some time to configure design area of our application. We are going to learn how to use all available styling options inside our Magento Mobile Admin Panel in order to design the basic interface of our application.

## Getting ready

Log in to your Magento Admin Panel, if not done already, and navigate to **Mobile | Manage Apps** in the main menu.

Click on our previously created application in order to open the **Edit App** screen.

Navigate to the **Design** tab on the left-hand side of the screen where we already uploaded logo in header and banner on home screen images.

You will notice the **Color Themes** tab in the middle of the screen. Expand it if not already expanded.

On the right side of the screen, we have some kind of preview box for our mobile application design.

## How to do it...

Follow these steps:

1. Choose the most appropriate **Preset Theme**:

   Under the **Color Themes** tab, navigate to the **Preset Theme** list. Here are some default predefined design themes for a mobile application. Click on the list to expand and preview the available themes. Choose a theme from the list.

   After choosing the theme, click on the **Update Preview** button in order to see the changes we made.

2. Define the custom interface and font colors:

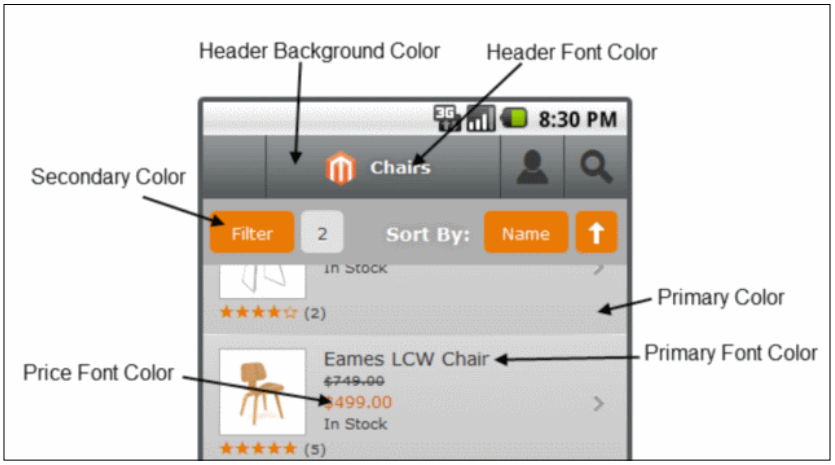
   Of course, if we want to make some changes in the selected default theme, we can do it, so let's inspect color options that are available to us for stylizing our app.

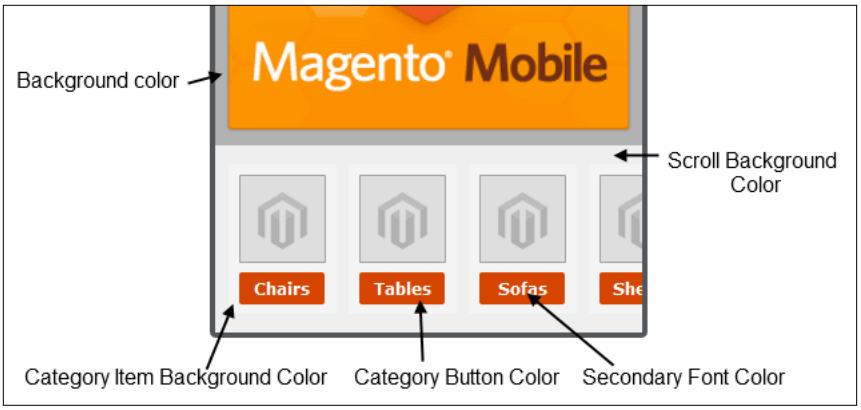
   Click on some of the color options and choose the custom color from the palette. To see the changes in our future mobile application, just click on the **Update Preview** button on the right-hand side of the screen.

   We can see that the new selected color appears somewhere on the **Preview Image** on the right.

   For the preview screen we can also choose screen 1 or screen 2 to see the changes applied to the mobile app interface.

   Since it is much easier to see something than explain it with words, let's list color options with images to see where each color appears:






3. Set up advanced settings:

   Click on and choose colors for each **Advanced Settings** option and click on the **Update Preview** button to see the changes.

> Advanced settings in most cases can remain the default and it is not necessary to change them, but feel free to play with options until the moment you are satisfied with your mobile application preview.

4. Define the mobile app title bar buttons:

   On the application header (title bar) there are some optional buttons that can be removed. For example, if we are using our application as catalog only, we don't need to display the **Account** button. Also, if we don't want to enable search in our mobile app, we can choose not to display the **Search** button on the header.

   Locate the **Tabs** row below the **Color Themes** tab and expand it if not already expanded, in order to see the available options.

   Under the **Title Bar** options we have the **Home**, **Search**, and **Account** buttons displayed and for **Search** and **Account** we have an option to disable them—not including them in our mobile application.

   Try to click on the **Make Inactive** link under the **Search** button and click on the **Update Preview** button to see the changes.

   Notice that the **Search** button disappears from our mobile application header. Of course, we can enable it again if we want; just click on the **Activate** link under the **Search** button and it will become active again.

   The same options are available for the **Account** button.

   The **Home** button is not possible to deactivate, because we must have it in our application, so the users can return to our app's home page from any screen inside.

5. Define the sort order of option menu items in our app:

   Under the **Options Menu Items** list, we have three items listed in the order they will appear inside our application's options menu:

   - **Shop**
   - **Cart**
   - **More info**

   Basically, they are just shortcuts to some of the screens in our mobile application. The **Shop** menu item is the shortcut to the *Categories* list, **Cart** is shortcut to the content of the user's *Shopping Cart,* and **More info** is the shortcut to our application's *copyright information* (**About** screen).

   Click on the left and right arrow icons under each menu item in order to change the sort order in which they will appear inside our app's options menu.

   We can also deactivate each of them by clicking on the **Make Inactive** link under each item and also reactivate by clicking on the **Activate** link.

6. Save our configuration:

   Once we are satisfied with the colors, fonts, and buttons in the mobile app, we just need to click on the **Save** or **Save and Continue Edit** button in order to save our configuration.

## How it works

When we define options and styles for our mobile application, and submit our app to Magento Mobile, the mobile application loads the configuration that we defined and sets up the look and feel by using defined options.

We can always change these options, but when our application is already submitted to App Store/Google Play, if we make some changes, it will be necessary to resubmit the application as a new one and that can cause some additional costs.

## There's more...

If we are unsatisfied with our custom color configuration, we can click on the **Reset theme to default** button to restore the default theme settings. Of course, not all of the screens of our application are visible from Preview Image in Magento Mobile Admin Panel. All the colors, depending on screen type, will be visible in a Preview application that we will use to preview the final mobile application before submitting it to Magento Mobile.



There is also the possibility to test our app with real devices, using Magento Mobile Previewer App for iOS and Android, which is covered in later sections of this book.

# Adding static content (Must know)

In a Magento Mobile app, we can define some static pages that will be visible under the **More info** screen in our mobile application. For example the **About us** page, **Contact** page, and so on.

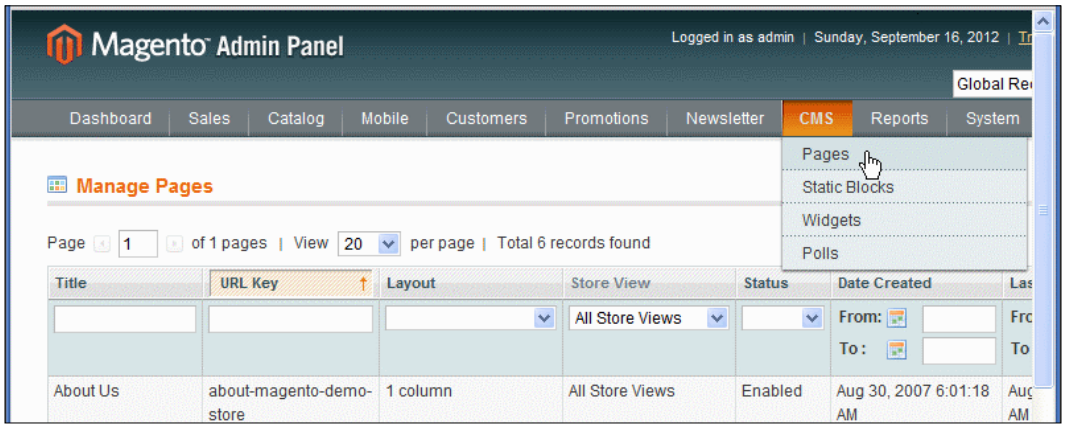
To do this properly, we will first define the static page content inside the **Magento Admin** area, and then we will assign that content (static page) as part of our mobile application.

Static content could be text/html with images and links, because inside a mobile application, it is loaded and displayed as formatted HTML inside the WebView component.

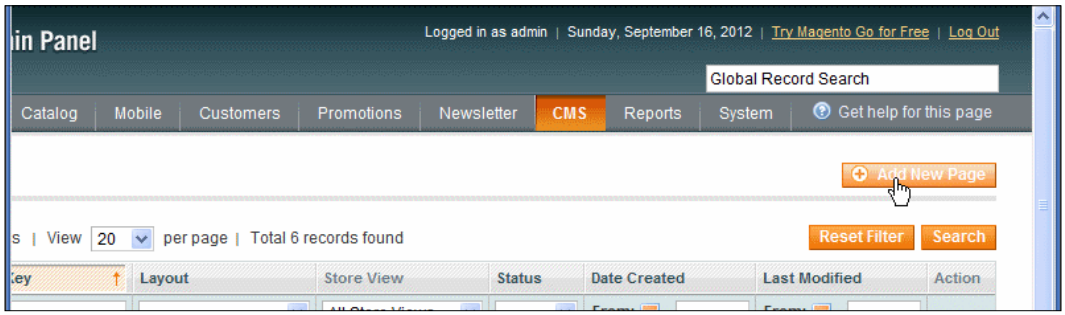
## How to do it...

1. Log in to your **Magento Admin** area, and navigate to **CMS** | **Pages** in the main menu. We can see the list of previously defined pages that can be displayed on our Magento website.

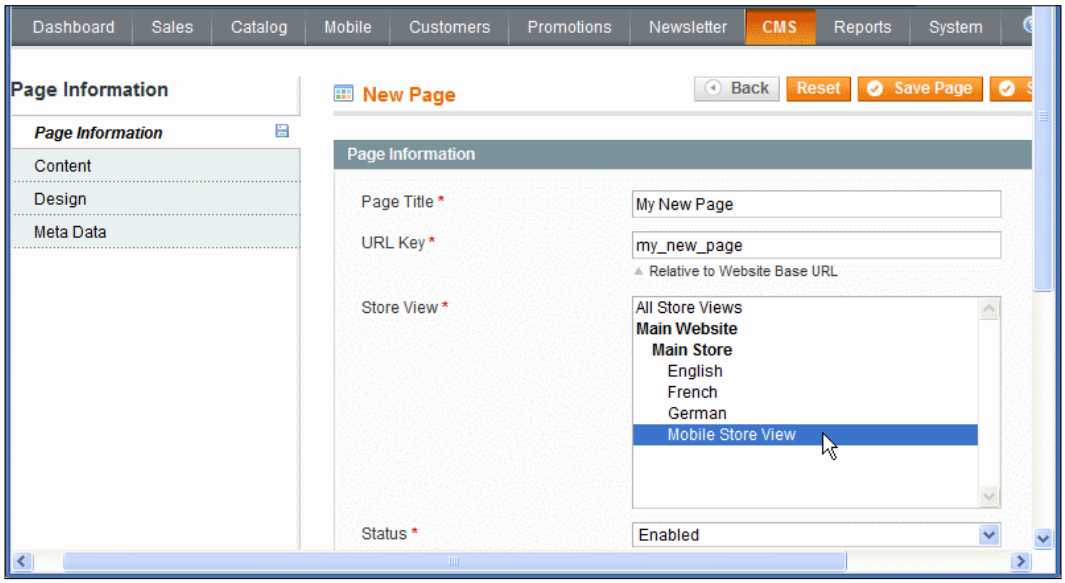
   On the **Manage Pages** screen, we have a grid with columns where we can see the columns, named **Title**, **URL Key**, **Layout**, **Store View**, **Status**, and so on. This is shown in the following screenshot:



2. Click on the **Add New Page** button in the top-right corner, in order to create new pages for our mobile application. The **New Page** screen is shown as follows:

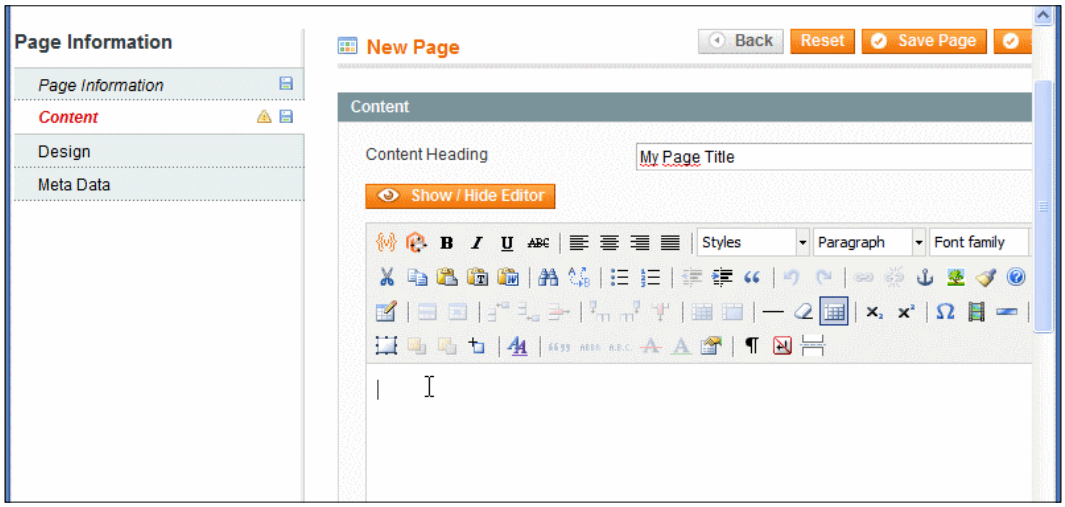



3. Locate the **Page Title** field and enter the title for a new page, for example, **My New Page**:



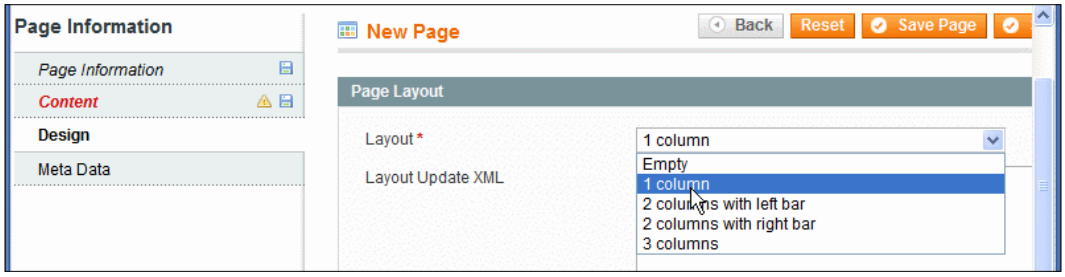

4. In the **URL Key** field, enter a string value that represents the title of our new page, but make sure it is in *lowercase letters* and one word or more words are connected with the underscore character (for example: **my_new_page**), as it will became a part of the *URL* that leads to this page.
5. Next, choose the **Store View** value we defined for our mobile application from the list.
6. Set the **Status** field to **Enabled** in order to make the page visible in our mobile application.

7. Click on the **Save and Continue Edit** button to save the current changes. The **Content** tab is now opened, as shown in the following screenshot:

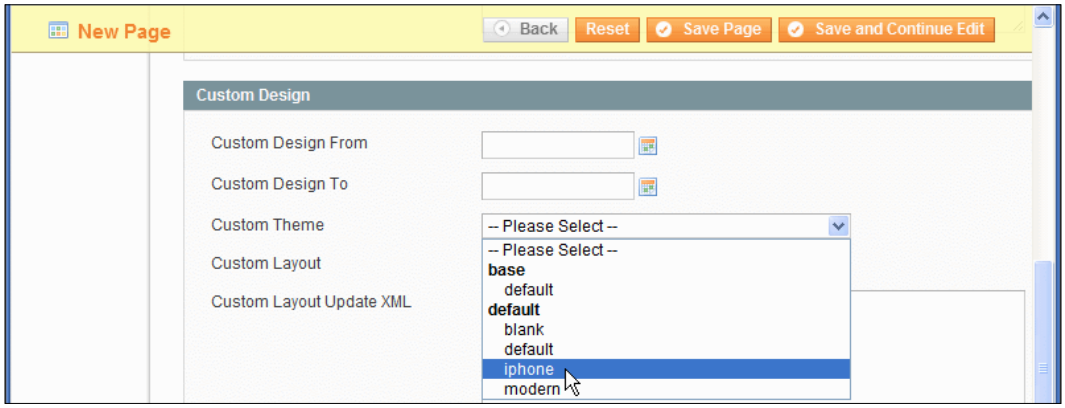



8. Under the **Content Heading** field, enter the title of the page as you want it to appear in the mobile application.
9. Below the **Content Heading** field, a web based HTML editor is shown. Here we need to enter the main page content. We are free to use the available HTML formatting and styles inside the editor to stylize our page content as we want. After we are done with the page content, click on the **Save and Continue Edit** button again to save the changes.
10. Locate the **Design** tab on the left-hand side of the screen and click on it to see the layout options. Since we are preparing a page for mobile devices, it is best to choose the **1 column** layout value under the **Page Layout** row.

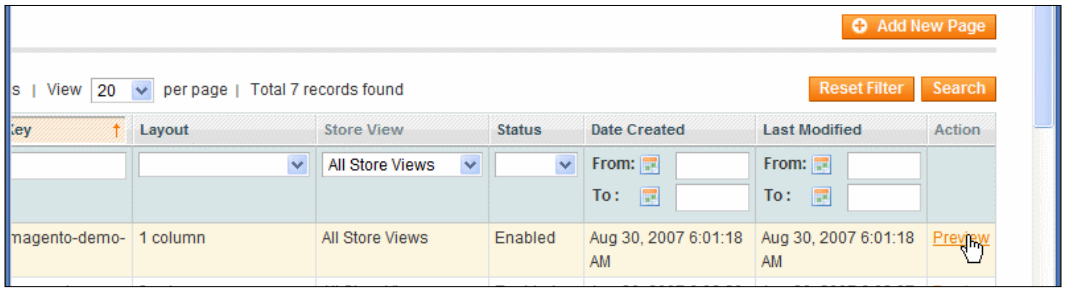
    **Layout Update XML** is something that we don't regularly need for our page and it is way beyond the scope of this book, so leave it empty.



11. Locate and expand the row with the label, **Custom Design**, and from the **Custom Theme** list, choose **iphone** since this is the theme we had set for our mobile application. Also set **Custom Layout** to **1 column**, because it is the best value for displaying on mobile devices:




12. Once you click on **Save Page**, the **Manage Pages** screen is shown and we can use the **Preview** link, on the right-hand side of each page listing to quickly preview our page in a web browser. This is shown in the following screenshot:






We don't need to insert any metadata for our mobile page, if it is going to be visible in our mobile application only, because metadata is used for web pages SEO (Search Engines Optimization), and our page is going to be integrated inside the mobile application, where search engines are not relevant.

13. Navigate to **Mobile** | **Manage Apps** in the main menu and click on our application that is listed under the **Manage Apps** screen, to open it for editing.
14. Next, navigate to the **Content** tab option on the left-hand side of the screen and there the **Pages** row is shown.
15. Click on the **Add Page** button to add a CMS page to our mobile application and under the **Label** field, enter a label for your newly created page.
16. Under the **Get Content from CMS Page** list, choose the page we just created and click on the **Save** button. The selected page has just become a part of our mobile application.

## How it works

Except the *Product Catalog* pages, Magento offers an option to add **Content Management System** (**CMS**) pages that are basically the static HTML pages, but on these pages we can add some information about the website/store (contact, about us, and so on).

Also we have an easy way to define the basic layout of these pages on the site (**1 column**, **2 columns**, **3 columns**, ...) and most of this we can do through the **CMS** | **Pages** screen accessible from the main menu.

## There's more...

Our mobile page can use any of the CMS pages defined in Magento, but because of the limited screen resolutions on mobile devices, it is always best to define them to use our mobile (iPhone) theme, which is to be displayed in a user-friendly way inside our mobile app.

We can have as many pages as we need inside our mobile application by adding them to our app, by clicking on the button, **Add Page**, in our application's content editing screen.

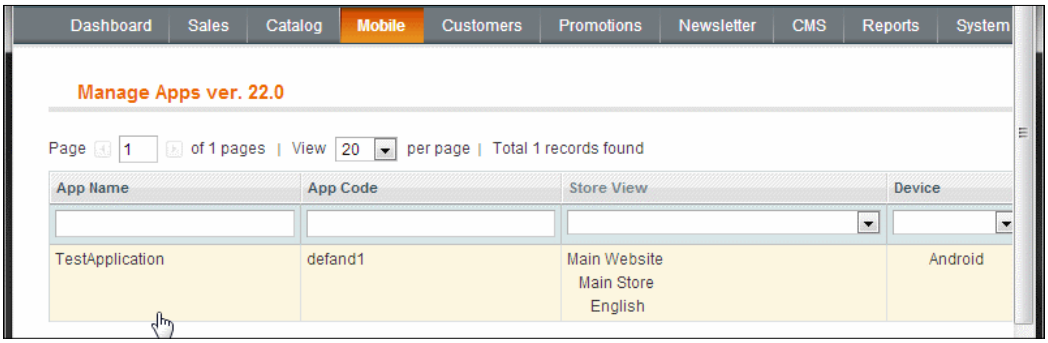
# Configuring payment methods (Must know)

As on the website store, we also need to configure the payment methods for our mobile application. In the following steps, we will see how easy it is.

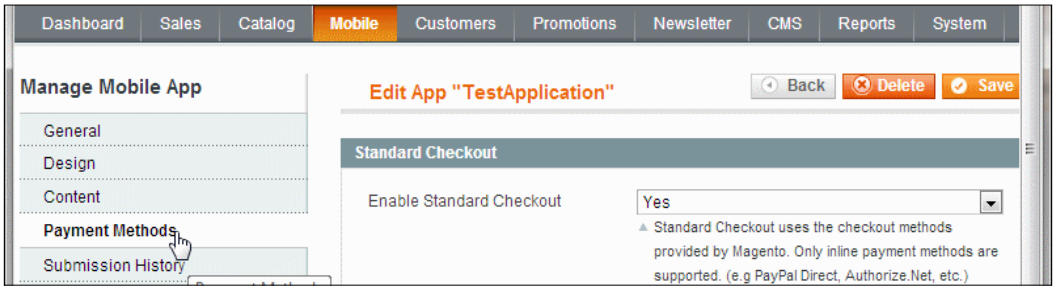
In fact, mobile payment methods are based on some of Magento's standard payment methods, and we basically just need to enable them through **Magento Admin Panel**.

## How to do it...

1. Log in to your **Magento Admin** area.
2. Navigate to **Mobile** | **Manage apps** in the main menu. A screen with listed mobile apps is shown, as illustrated in the following screenshot:



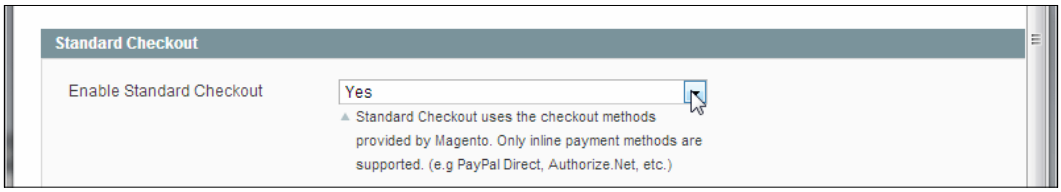

3. Click on our mobile application in the list to open the **Edit App** screen.
4. Navigate to the **Payment Methods** tab on the left-hand side to open the payments configuration screen:



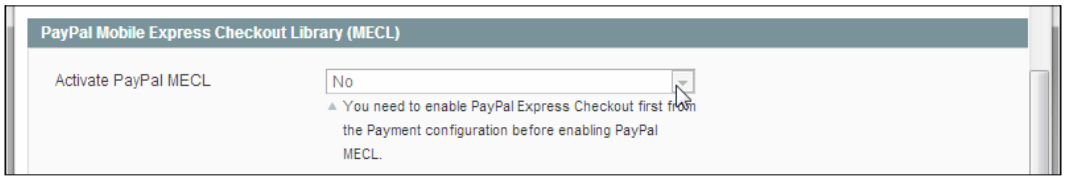

5. Choose the **Standard Checkout** option:

   With this option enabled, some of Magento's default payment methods (which are enabled through the **System** | **Configuration** | **Payment Methods** area) will also become available in our mobile app.

   For example, PayPal Direct, Autorize.Net, Check/Money Order, and so on.



6. Choose the **PayPal Mobile Express Checkout Library (MECL)** option:

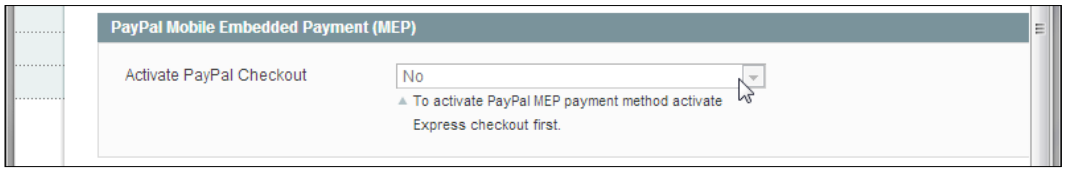





The MECL method is currently available for Android applications only and this option will not be shown for iPhone or iPad.

In order to enable it, the PayPal payment settings have to be pre-configured inside the **System** | **Configuration** area under the **PayPal** section.

7. Choose the **PayPal Mobile Embedded Payment (MEP)** option:






MEP is currently available for iPhone and iPad apps. Android does not currently support it.

Also the main PayPal settings have to be previously configured in **System** | **Configuration** under the **PayPal** area.

## How it works...

Before we configure any of the third-party payment methods, we need to configure them inside the **System** | **Configuration** area of our Magento Store. Also be aware that it may be necessary to make some contracts and agreements with payment provider companies in order to be able to use them in our store/mobile application. For example, if we are going to use any of the available PayPal services for mobile, we first need to configure the main PayPal settings for our Magento installation from the **System** | **Configuration** | **Paypal** screen, and then activate the desired mobile payment method based on it.

# Configuring social networks integration (Should know)

If you want to promote the products on your web store, there are many ways to do that and one of them is marketing through social networks.

Magento Mobile gives us the possibility to enable the users of our mobile app to share the product links from our website through their private profiles on Facebook, Twitter and LinkedIn, and that way we can get more visits on our web store and possibly increase the sales.

## Getting ready

We can choose if we are going to enable link sharing for all available social network options inside the Magento Mobile app (Facebook, Twitter, LinkedIn) or just some of them. In order to do that, we need to create an account on each social media site and configure it for our mobile app.

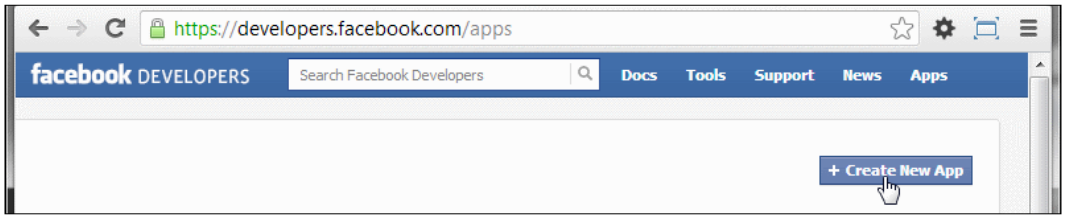
## How to do it...

1. Set up a Facebook account for Magento Mobile integration:

   First of all, sign-up at Facebook and activate an account if you haven't one already. It is probably best to create a new account with the name of your web store rather than a private account.

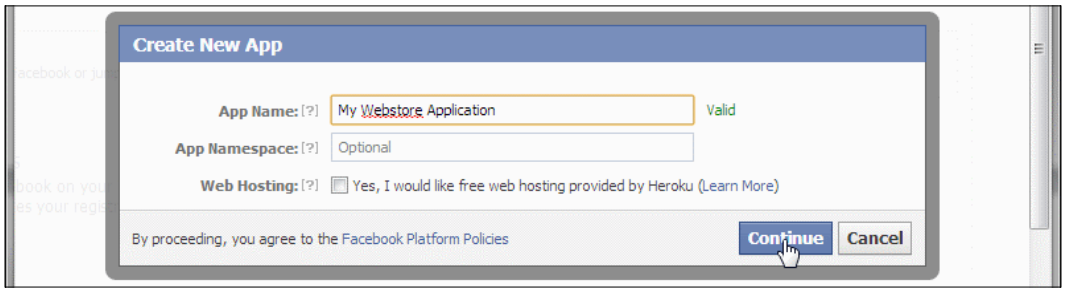
   Log in on Facebook with your credentials and visit the next link in order to create a Facebook application that will communicate with your Mobile app: `https://developers.facebook.com/apps`.

   Click on the **Create New App** button in the top-right corner of the screen:

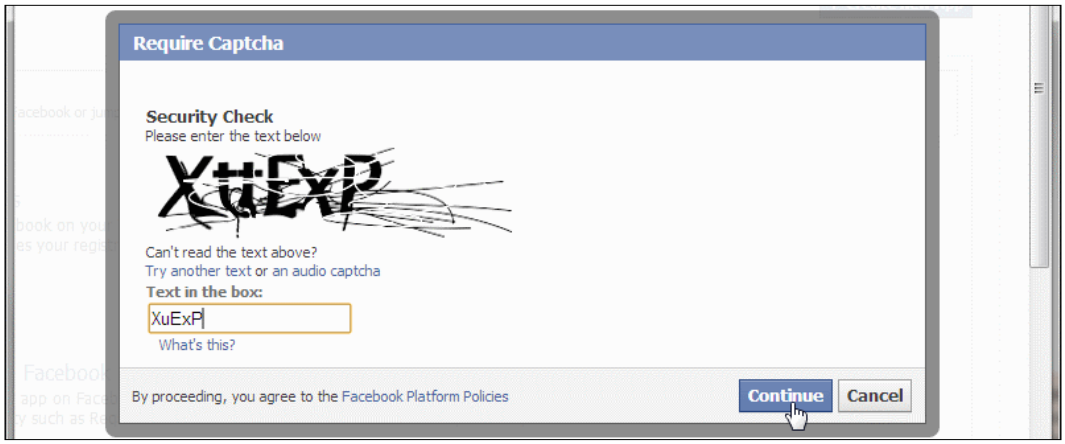



   The **Create New App** dialog is shown. Just enter the name of your app in the **App Name** field, for example, **My Webstore Application**.

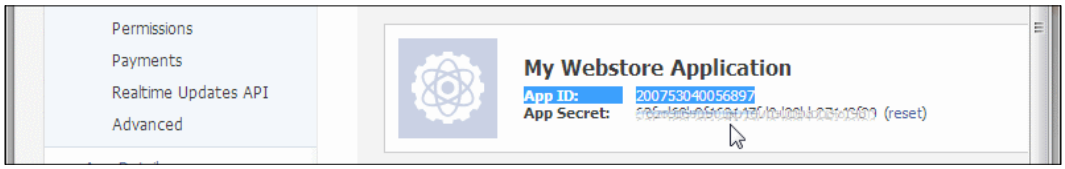
Leave the other fields as they are and press the **Continue** button, as shown in the following screenshot:




You need to enter a security check code in order to continue, so in the next dialog, just enter the characters into the text field as they appear in screenshot shown and press the **Continue** button.



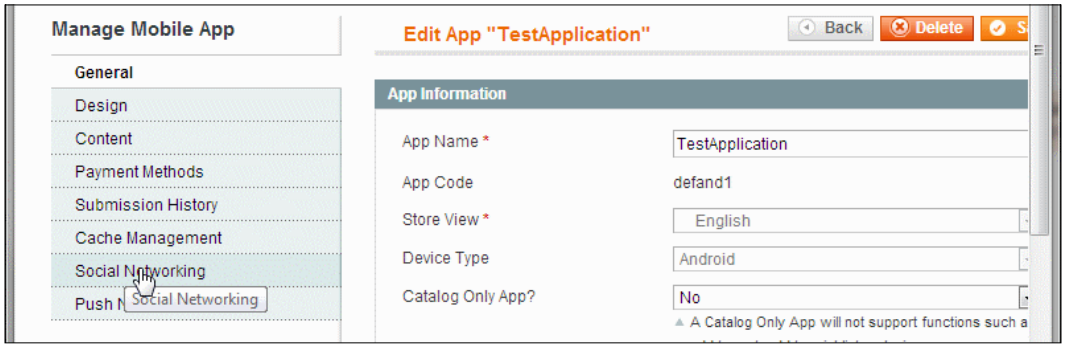

Congratulations, you have just created your Facebook App. With the App screen shown as follows, find the text value **App ID** somewhere on the top-center of the screen, near the **App Name** value, and copy the App ID value in memory, or write it down.



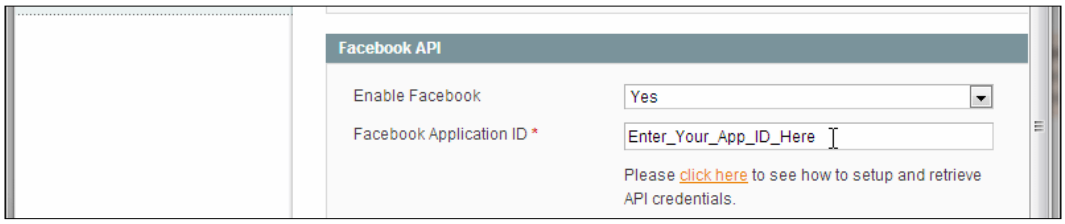
2. Enable Facebook integration through **Magento Admin Panel**:

   Log in to your **Magento Admin** area and navigate to **Mobile** | **Manage Apps**. Click on your Magento Mobile application created earlier to open the **Edit App** screen.

   On the **Edit App** screen, navigate to **Social Networking**, as shown in the following screenshot:




   Locate the **Facebook API** tab in the middle of the screen and set the **Enable Facebook** value to **Yes** and inside the **Facebook Application ID** field, paste the App ID copied from your Facebook app page.




   Click on the **Save and Continue Edit** button to save new settings.

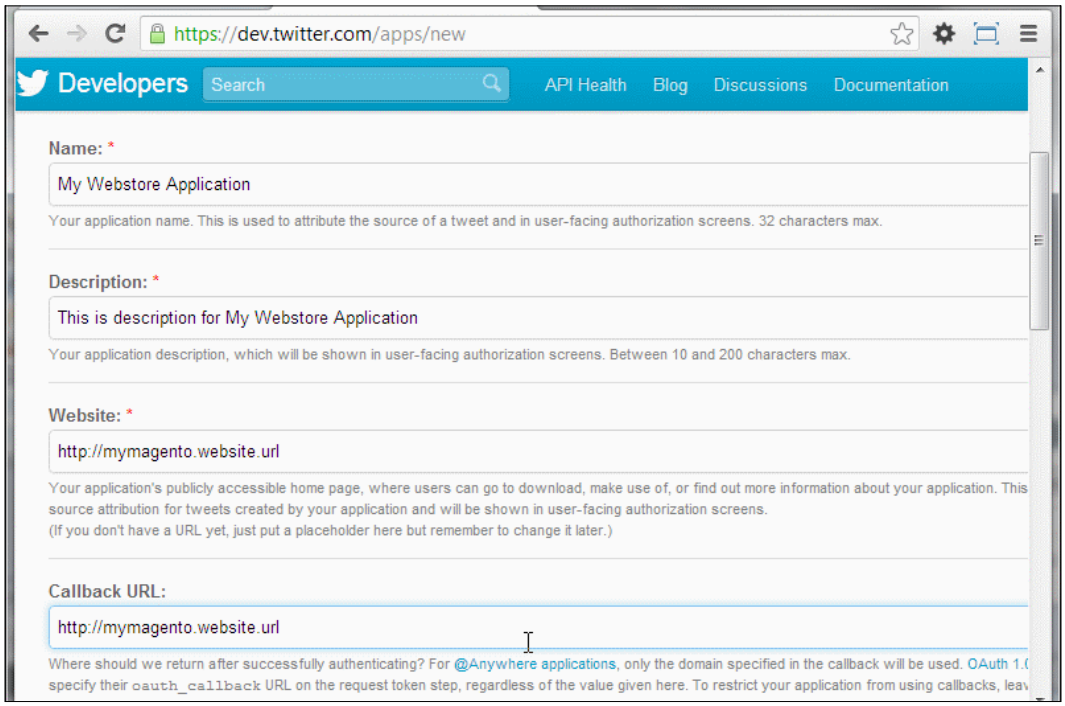
3. Set up a Twitter account for the Magento Mobile integration:

   First log in to your created Twitter account.

   Visit the following link in your browser in order to create the Twitter application that will be connected to your Magento Mobile app:

   `https://dev.twitter.com/apps/new`

Fill in the **Name** field with the custom chosen app name and add a short description to the **Description** field, as shown in the following screenshot:






A description will be shown to the user when the **Share on Twitter** functionality is used for the first time from your mobile App, so make sure that it is short and clear.

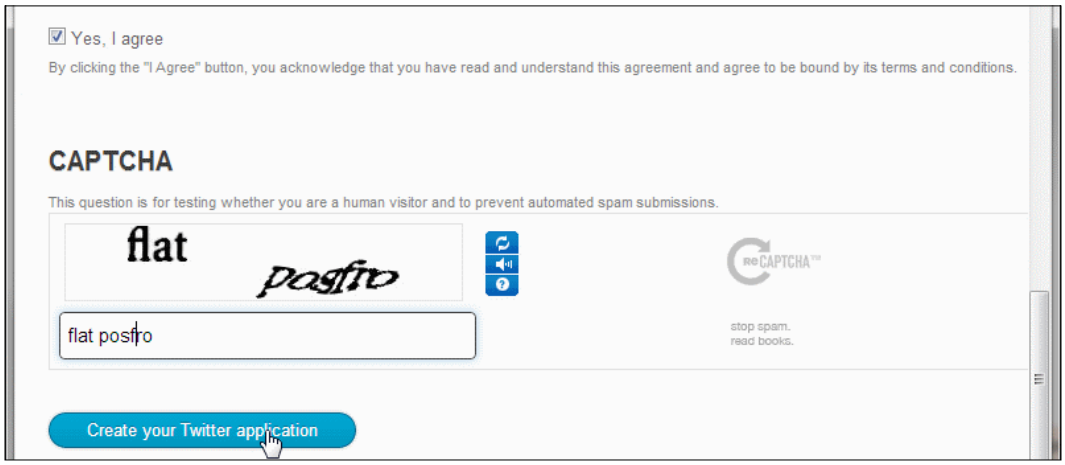
Under the **Website** field, enter the URL of your web store.



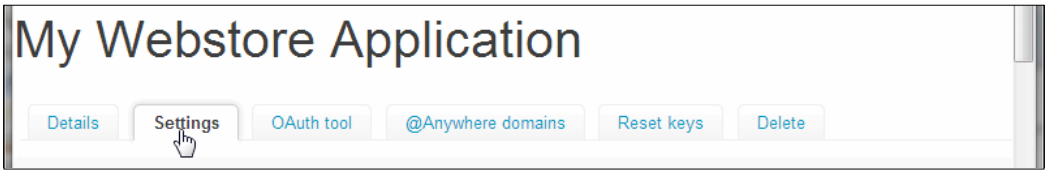
Enter your site URL as well, under the **Callback URL** field; otherwise the entire thing won't work.

Also you have to agree with **Twitter Developer Rules** by checking the **Yes, I agree** checkbox.

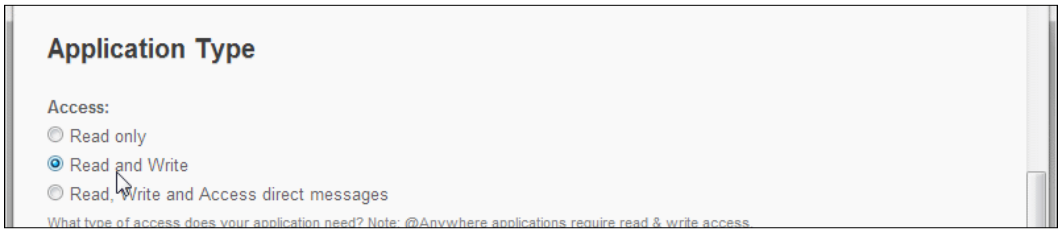
Fill in the provided **CAPTCHA** code and click on the **Create your Twitter application** button:




The application edit screen is shown as follows:

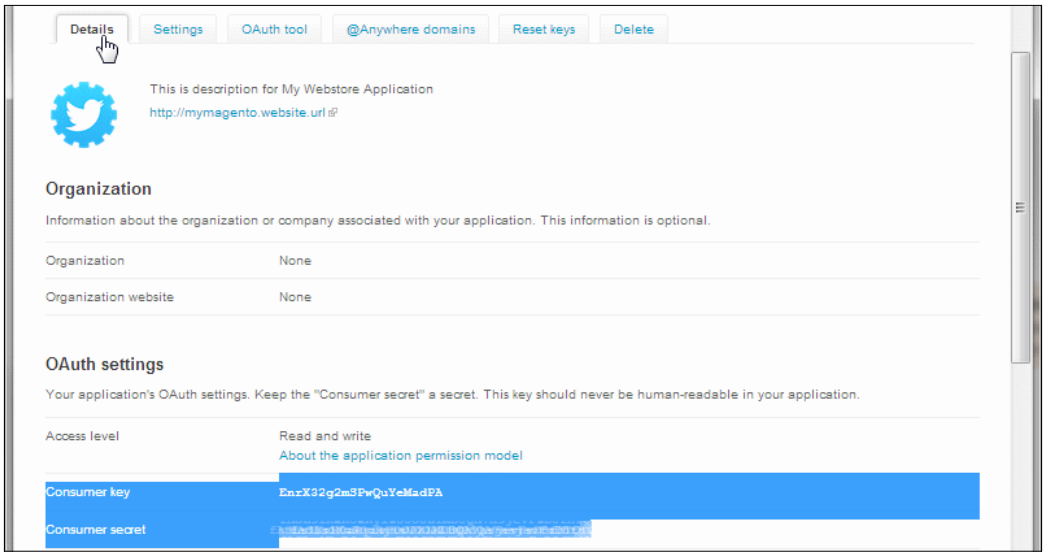



Navigate to the Settings tab and in the Application Type area, choose the **Read and Write** access type.




Click on the **Update this Twitter application's settings** button to save your app.

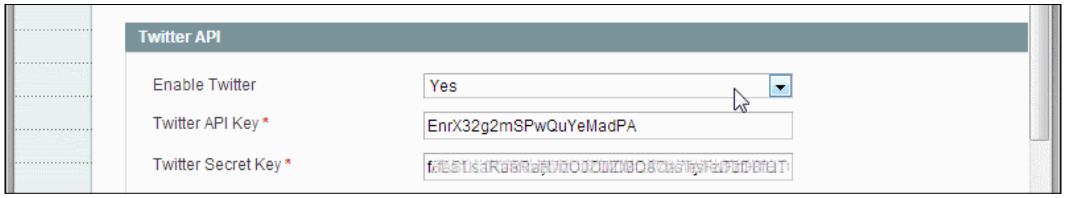
Navigate to the **Details** tab and locate **Consumer Key** and **Consumer Secret**, which have to be provided to **Magento Admin Panel**, and write them down.




4. Enable Twitter integration through **Magento Admin Panel**:

   Inside the earlier opened **Social Networking** screen of **Magento Admin Panel**, locate the **Twitter API** section.

   Set the **Enable Twitter** field value to **Yes** and copy the necessary values from the Twitter app page into your Magento Mobile app **Twitter API** fields. Copy the **Consumer Key** value from the Twitter app to the field named **Twitter API Key**, and also the **Secret Key** value from the Twitter app to a field named **Twitter Secret Key**.



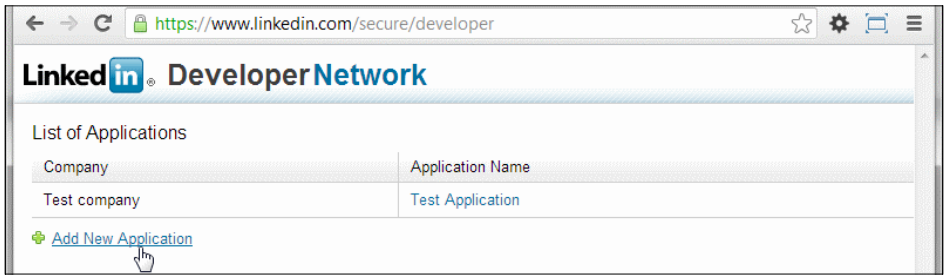

   Click on the **Save and Continue Edit** button to save and leave this page open for one more integration, that is, LinkedIn.

5. Set up a LinkedIn account for the Magento Mobile integration:

   Log in to your LinkedIn account.

   Visit the following URL to create a new LinkedIn application: `https://www.linkedin.com/secure/developer`.

Click on the **Add New Application** link, as shown in the following screenshot:

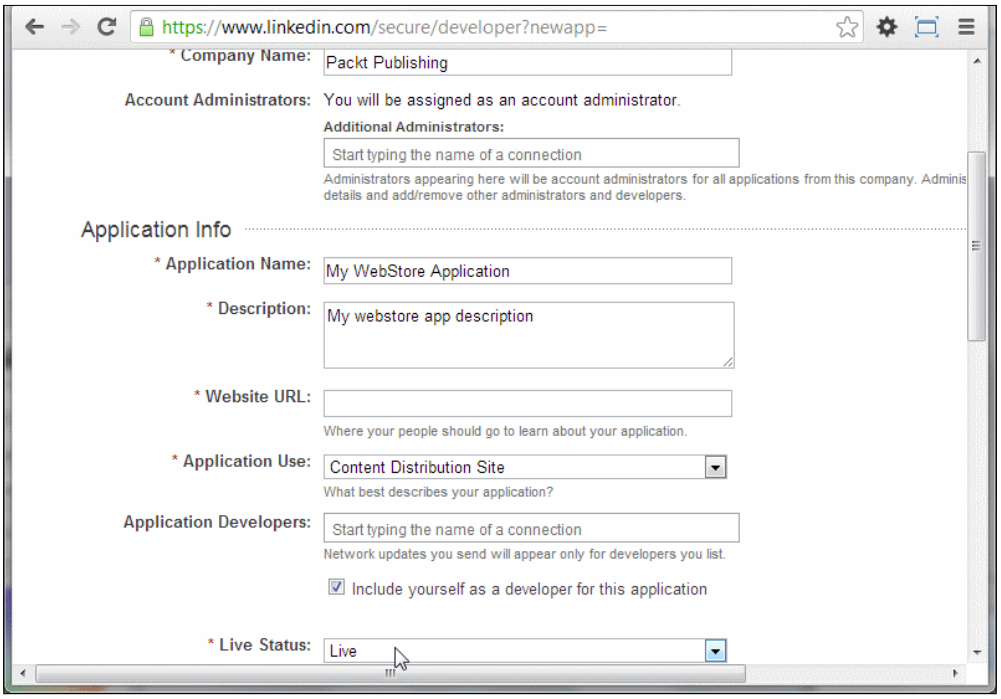



Fill in the required fields:

- **Company name**
- **Application Name**
- **Description**

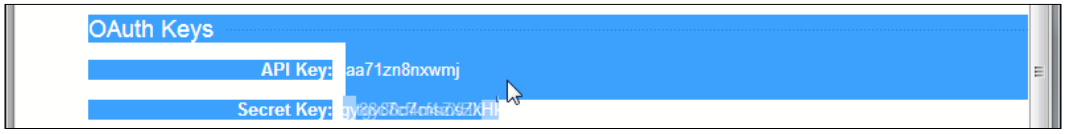
Choose the following settings from the list:

- **Application Type**: **Web Application**
- **Application Use**: **Content Distribution Site**
- **Live Status**: **Live**



Next, fill in the required contact information and confirm with **Agree** in the **Terms of Service** section.

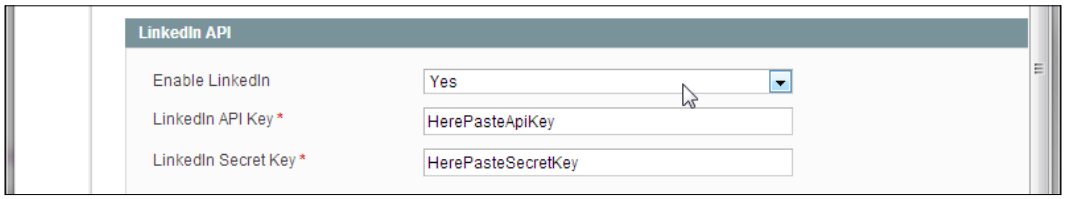
Click on the **Add Application** button. You will notice the **ApiKey** and **SecretKey** values listed on the screen. Leave this screen open, in order to be able to copy the provided values inside your **Magento Admin Panel**.




6. Enable the LinkedIn integration through **Magento Admin Panel**.

   Inside the open **Social Networking** screen of **Magento Admin Panel**, locate the **LinkedIn API** section.

   Set the **Enable LinkedIn** value to **Yes** and paste the **LinkedIn API Key** from your previously created LinkedIn application. Also copy and paste the LinkedIn secret key here.




7. Click on the **Save** button and we are done.

## How it works...

In order to be able to publish updates on a particular social network, we first need to create an application on it that will do the publishing job for us. When a user browses through products in our mobile store, a long press made on any specific product should give the user an option to share that product over his/her social network account.

Users can choose from social networks that we configured for our application inside **Magento Admin Panel**.

When a user clicks on the **Share** link, our mobile app will trigger our social network application, recognized by the corresponding API key. Then, the user will be asked to log in to the social network with his credentials and grant permissions to our application to publish updates in his account.

## There's more...

When a user confirms for the first time that our application is allowed to publish under his account, the user will not be asked the next time he/she wants to share some product under his/her account.

Social networks integration is something that should be taken very seriously and we should make sure that it's included inside our mobile application, because it can bring more profit to our store. It should be considered as a free source of marketing possibilities for our store.

# Configuring the Push notification feature (Become an expert)

The Push notification feature allows us to send notification messages to all customers that have our Magento Mobile application installed on their devices, even if the application is not currently turned on.

It can be useful if we want to send some kind of marketing messages, or some other messages to our app users to draw attention to some new product at the store, and so on.

The Push notifications feature for Magento Mobile is an extra service provided by a third-party company: *Urban Airship* in co-operation with Magento Mobile.

It is an optional feature with additional costs for us as store owners, but it can bring some benefits to our sales marketing.

It is up to you to decide if you want to use it or not. Anyway, it is not necessary to configure it right away, because it can be implemented later on in the next re-submission to App Store/ Google Play.

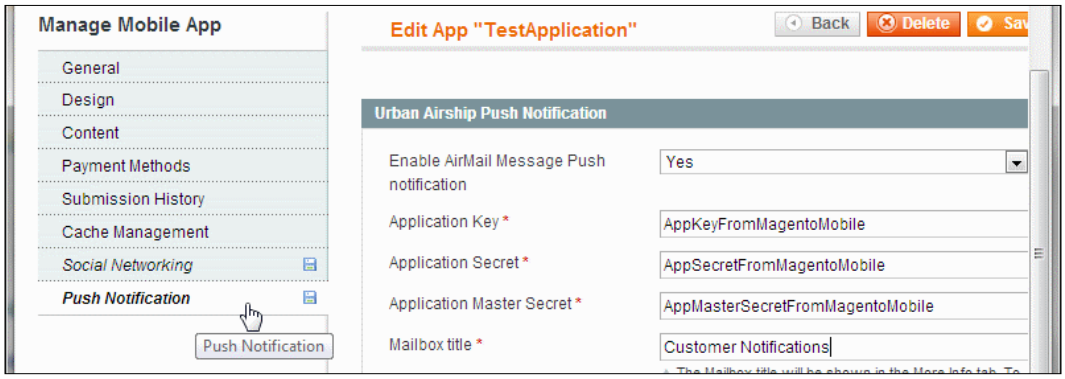
## Getting ready

In order to be able to successfully set up the Push notification feature, it will be necessary for our mobile app to be already submitted to Magento Mobile and has to be done before our app is published to App Store/Google Play.

## How to do it...

1. After the app submission, contact `magentomobile@magento.com` to receive the **Application Key**, **Application Secret**, and **Application Master Secret** of our mobile app that are necessary for configuration.
2. Log in to your **Magento Admin Panel** and navigate to **Mobile** | **Manage Apps**. Click on your mobile application from the list in order to see your app editing screen.

3. Navigate to the **Push Notification** tab from the left-hand side menu.
4. Set the **Enable AirMail Message Push notification** value to **Yes**.




5. Fill in the **Application Key**, **Application Secret**, and **Application Master Secret** values with the information received from Magento Mobile. Add some custom name in the **Mailbox Title** field, for example, **Customer Notifications**. The mailbox title will be shown inside the **More** screen of our mobile app, and in it our push notifications sent to customers will be listed.
6. Click on the **Save** button in order to save the new configuration.

## How it works...

The Push notification feature is implemented inside your Magento Mobile app and when the user installs your mobile application. The user's device sends a unique device token that is used by the `Urban Airship` API to pass your messages to that user's device.

You can log in with your **App Key** and **App Master secret** values to *Urban Airship* on this URL: `https://go.urbanairship.com/accounts/login/` and explore more options available for the Push notification service.

## There's more...

Except the basic Push notifications, which are used to send some plain text messages to the users, there is also an option to use a **Rich Push** notifications service for your Magento mobile app.

The Rich Push notifications give us the possibility to add formatted HTML and other rich media components into push notification messages. A Rich Push message takes the user into the application where the message is displayed in a customizable WebKit view.

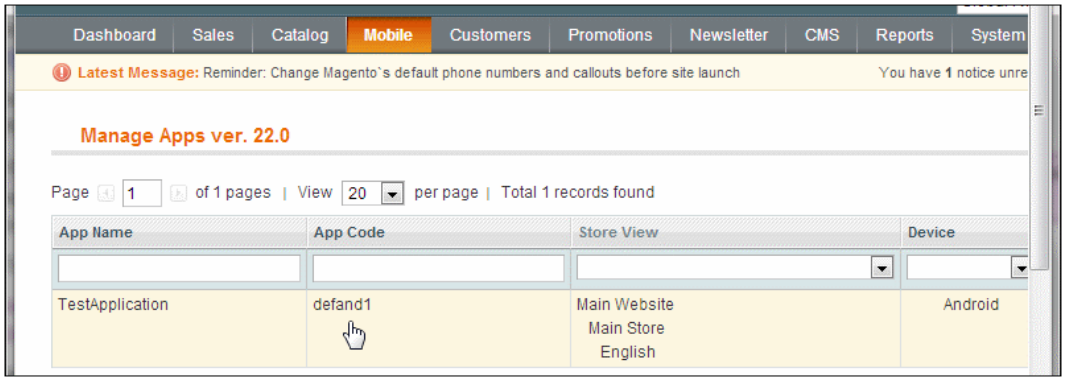
# Installing and using the Magento Mobile previewer app (Should know)

Even before publishing our mobile application, it is possible to test it on mobile devices using the **Magento Store Mobile** application provided by the Magento company, which is available for free download on App Store/Google Play.

## How to do it...

1. Open your App Store/Google Play application from your mobile device and search for the **Magento Store Mobile** application developed by *Magento Inc.*, and download it to your mobile device.
2. Start the **Magento Store Mobile** app and the main login screen will be shown.
3. In the first textbox, enter the URL or your Magento website, for example: (`http://yourwebsite.com/`).
4. In the second textbox, enter the auto generated **App Code** from **Magento Admin Panel** (for example, **defand1**).

   You will find that the code by logging in inside the Magento admin area with your browser, and navigating to the **Mobile** | **Manage Apps** screen. **App Code** is shown in the following screenshot in one of the columns on the grid:




5. Touch the **Continue** button on the device to connect with your store.

   The home screen will be shown with the listed categories we assigned to our mobile Store View.

6. Navigate through the *categories*, *info*, *user account*, *products list*, and *product detail* screens on device to see how your defined application will look like.
7. Log in to the **Magento store admin** area and through **Mobile** | **Manage apps**, open the application edit screen.

   Try to change some **Design**, **Color** theme settings or add/remove some navigation tabs to/from your app. After saving the changes, you will have to close and open the **Previewer** app again on the mobile device to see the changes.

## How it works...

The Magento Store Mobile app is made exactly the same way as your application will look, with the difference being that it has a login screen, so that you can preview any defined application on any URL. Your application will not have the login screen and your store will be immediately shown when the user opens the app on a mobile device.

Every time the Magento Mobile Preview application is opened, it loads configuration data from our website and it is displayed accordingly to our configuration.

## There's more...

You can test social networks integration along with some other options from inside the preview application by navigating to the product list inside the mobile app. Just touch and hold a finger on some specific product and a popup menu with several options will appear. If the social networks integration is properly configured, you will see the **Share It** link on the popup menu, and will be able to share that specific product on chosen social networks.

# Enrolling into the iOS Developer Program (Should know)

In order to successfully submit our iPhone, iPad mobile application to the App Store, we need to enroll into the **iOS Developer Program** and give our developer credentials to Magento Mobile when submitting our finished app, and Magento Mobile will do the rest of the job for the setup and deployment on the App Store under our developer account.

## Getting Ready

One year's subscription to iOS Developer Program has some additional annual costs (currently it is $99/year), so be prepared to use your credit card a little bit more.

If you are enrolling as an *individual*, there's much less complicated documentation that we need to provide to Apple is far less complicated, than enrolling as a *company*. If you are enrolling as company, please prepare the additional data that will be necessary to provide to Apple:

1. The legal authority to bind your company/organization to the Apple Developer Program legal agreements.
2. An address of the company's principal place of business or corporate headquarters.
3. A D-U-N-S® Number assigned to a legal entity.



More info about the data can be found in Apple Developer Program's FAQ: `https://developer.apple.com/support/D-U-N-S/`

## How to do it...

1. Navigate your browser to `https://developer.apple.com/programs/ios/`.
2. Once you click on the **Enroll Now** button, the enrolling info screen is shown.
3. After reading the information on this screen, just click on the **Continue** button.
4. Choose **I need to create a new account ...** and click on **Continue**.
5. Click on either the **Individual** or **Company** button depending on how you would like to enroll.
6. Fill in the required fields on the registration form (*Apple ID information*, *Security information*, and *Personal information*) and click on **Continue**.
7. Follow further directions to confirm your e-mail address.
8. Accept the Apple License agreement.
9. Continue with purchasing the yearly subscription and finally activate the program by following the directions on the Apple Developer Program pages.

## How it works...

After confirming and purchasing the Apple Developer Program yearly subscription, you can submit as many applications for iPhone/iPad you developed on the App Store while the yearly subscription is active.

## There's more...

If you're enrolling as a company, it is possible that you will have to provide some additional documents to Apple in order to activate your subscription.

Apple will notify you if any more information is required and the way you can provide any such documents to them (by e-mail, fax, and by other means).

# Enrolling into the Android Developer program (Should know)

Enrolling into the Android Developer program is far less complicated than Apple's. Also, we don't have the yearly subscription but a one time Developer Registration Fee (currently $25).

One more advantage is that we don't have the application reviewing process as in the App Store, but our application is available on Google Play immediately after submission.

## Getting ready

Make sure that you have a previously registered Gmail account in order to continue.

## How to do it...

1. Navigate with your browser to `http://play.google.com/apps/publish`.
2. Sign in with your existing Gmail credentials.
3. Fill in your basic developer account information (Developer Name, E-mail address, Website URL, Phone number) and click on the **Continue** link.
4. Agree to the Android Market Developer Distribution Agreement and click on the **I agree**, **Continue** link.
5. Click on **Continue** again and the **Google Checkout** screen will open with the Android-Developer Registration Fee inside the shopping cart.
6. Fill in the required credit card payment information and confirm payment.
7. After confirming payment, your developer account is ready to use.

## How it works...

After confirming payment, we are able to publish applications to Google Play. With our credentials, if we are building Android app, Magento Mobile will publish our application to Google Play for us.

## There's more...

If we are developing an application that is not free, but paid on Google play, we will need to get a **Google Checkout Merchant** account in order to receive payment for selling our application.



In a Magento Mobile case, it is quite logical that our application will be free for download because we want as many customers to get our app to use it for buying on our web store, so we don't have to worry about the Google Checkout Merchant account.

# Application submission to Magento Mobile (Must know)

After all that hard work, finally it is time to submit our mobile app to Magento Mobile. There are just a few more steps, so let's get started.

## Getting ready

Prepare your credit card to purchase your Magento Mobile activation key.

## How to do it...

1. Navigate to `http://www.magentocommerce.com/product/mobile#mobile-tabs` with your browser.
2. Choose your app targeted platform(s) and pricing plan from the `Plans` and `Pricing` table.
3. Check the **I agree to the terms and conditions** checkbox and click on the **Add to Cart** button.
4. If you are satisfied with the content of your shopping cart, just go to **Checkout** and complete your purchase of the Magento Mobile activation key.
5. When you get the key, log in to your **Magento Admin Panel** and navigate to **Mobile** | **Manage Apps**.
6. Click on the app for which you bought the key and if your application is ready for submission, just click on the **Save and Submit App** button in the top-right corner. The **App Submission** screen will be shown.
7. Paste your app's activation key in the required textbox.
8. Enter the title of your application, which will appear on Google Play/App Store.
9. Enter the short description of your application in the **Description** field.
10. Enter your contact e-mail address, which will be used both for Google Play/App Store and also by Magento Mobile to contact you for all necessary contact requests.

11. Choose the countries from where your application is going to be downloadable on App Store/Google Play.
12. Enter some short copyright information.
13. Choose the requested appropriate icons for your application.
14. Take the time once more to preview your application on the preview screen.
15. Click on the **Submit App** button to finish your submission.
16. Send your App Store/Google Play credentials from your contact e-mail address (provided here in the point #10) to the Magento Mobile e-mail address: `mobileinfo@magento.com`.

## How it works...

Once the required information is provided and our application is submitted to Magento Mobile, our preparation is done. When the application is submitted, the rest of the work is done by Magento Mobile. They will contact you for any changes and additional information if necessary.

## There's more...

After submitting our application, we need to wait a few days before our application appears on Google Play/App Store.

On the App Store, the application reviewing process may take up to 10 days, but in Google Play, there is no reviewing process, and we will be notified when our application is published and available on App Store/Google Play.

# Application resubmission and monitoring progress (Should know)

After first submission, in case we need to change some of our app core settings (app icon, app description, and so on), it may be necessary for us to re-submit it in order for our changes to become available in the published version of our app.

Also, we can see the status of the specific app from our **Magento Admin** area.

## How to do it...

1. Log in to your **Magento Admin Panel**.
2. Navigate to **Mobile** | **Manage Apps**. Our application list is shown and in the right-hand side column labeled **Status**, we can see if our application is successfully submitted to Magento Mobile.

3. For eventually re-submitting our application, we need to purchase the re-submission application key from Magento Mobile, in the same way we purchased the first submission key (the re-submission key will become available on Magento Mobile for us when the first submission key is purchased).
4. From the already submitted application edit screen inside our **Magento Admin Panel**, just click on the **Save and Submit app** button with the new re-submission key inserted in the requested field.
5. In order to see the submission history for any of our applications, we need to navigate to **Mobile** | **Submission History**.

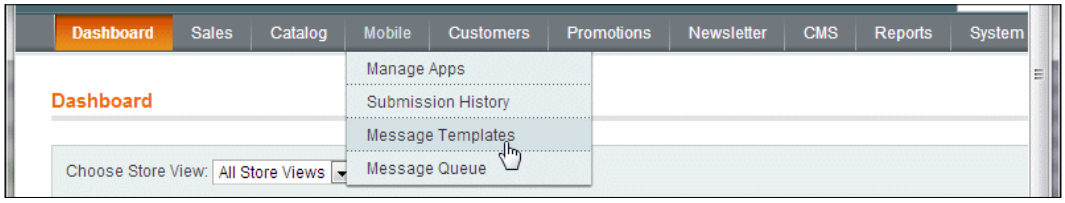
# Push notification messages administration (Become an expert)

If we enable and set up push notifications for our mobile app, when it is published and available on App Store/Google Play, we can send the messages to users of our app to inform them about new products, new application versions, and so on (if we change something and re-submit the app).

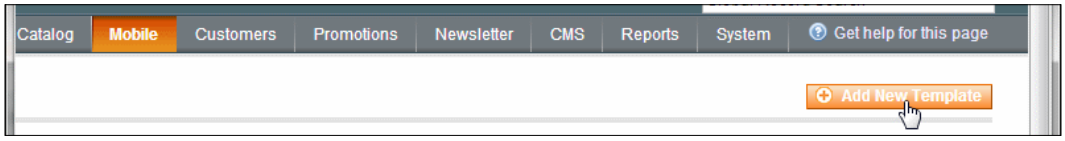
Here we are going to learn how to create a new template for messages and how to track our sent messages queue.

## How to do it...

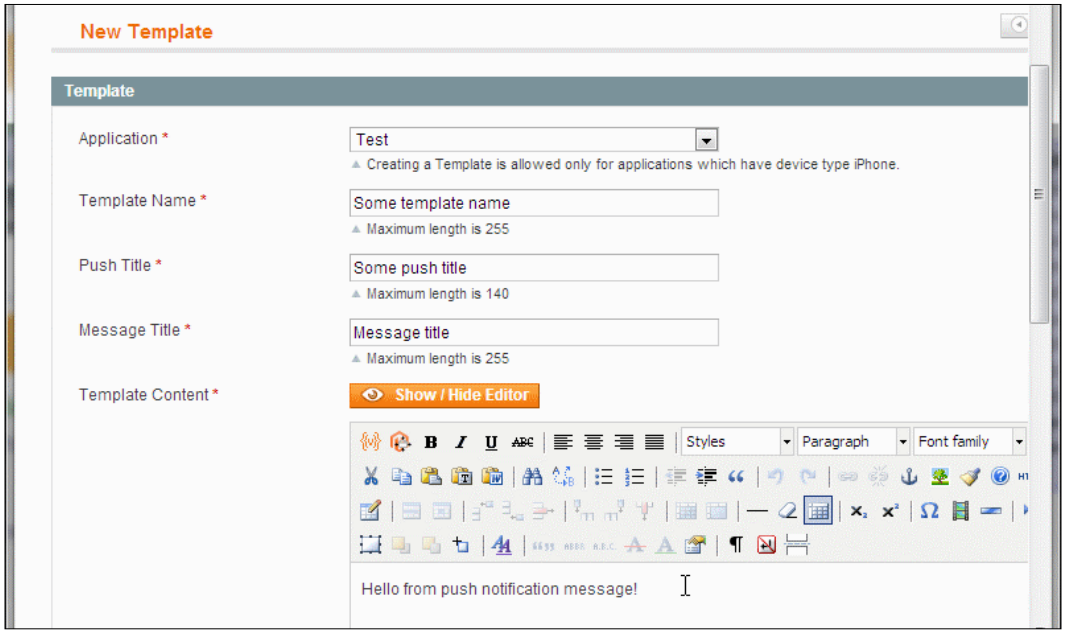
1. Log in to your **Magento Admin** area.
2. Navigate to **Mobile** | **Message Templates**:




3. Click on the **Add New Template** button in the top-right corner of the **AirMail Templates** screen. The **New Template** screen is shown as follows:

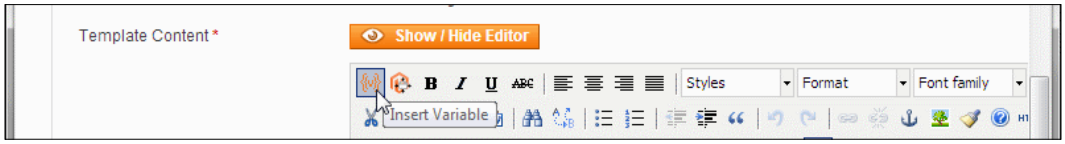


4. From the **Application** select box, choose the appropriate app for which you want to create a template.
5. Enter some custom **Template Name** in the next field.
6. In the **Push Title** textbox, enter the title of the message that will appear to the users when the message is received.
7. In the **Message Title** textbox, enter the title of the message that will be displayed when the user opens the message on mobile device to see its details.




8. Inside the **Template content** field, enter the main content of the message. Here we have options to click on the **Show/Hide Editor** button in order for the editor to display plain text/HTML formatted text of our message content.

   Here we also have an option to insert some of Magento's default or custom defined variables by pressing the **Insert Variable** button and choosing one from the list:



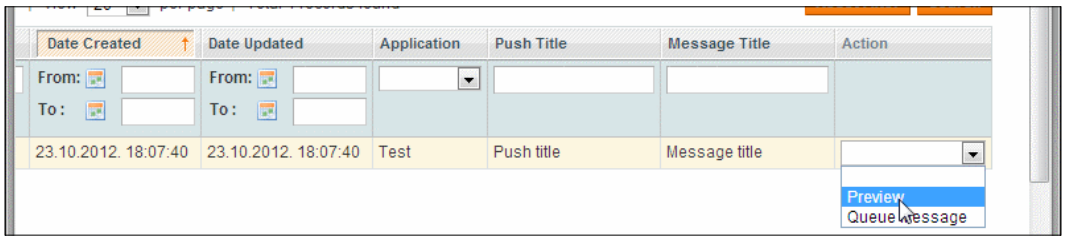


Please note that if we didn't subscribe to the additional **Rich Push** feature with Airship, our messages would not be sent as HTML formatted text but rather in plain text, so in that case, it is good to hide the editor when entering the text of the message. Otherwise, we are free to use any of the HTML formatting options that the displayed editor provides, in order to create our message.

9. After we are satisfied with the content of our message, we just need to click on the **Save** button in the top-right corner of the screen.

   When we save our template, the **AirMail Templates** grid screen is shown with the excerpt filtering on different columns; from here we have an option to preview our template and also to send it to our mobile app users.

10. For the purpose of previewing the template again, under the **Action** column on your template row, choose the **Preview** value, and a pop-up window will be shown with a preview of your message being displayed:




   If you are unsatisfied with the way it looks, you can always click on the template row in order to edit the template.

11. To send the message, just choose **Queue Message** under the **Action** column. The message is now queued and will be sent.

12. Let's navigate to **Mobile** | **Message Queue** to see and manage the progress of your messages. The **AirMail Messages Queue** screen is shown with multiple filtering options to display the queued messages.

    Under the **Status** column, we can see the status of our messages: **Canceled**, **In Queue**, or **Complete**. Successfully sent messages will have the **Complete** status.

## How it works...

When we define the template for our message, the same can be used for re-sending the same message multiple times. The template content is saved in our Magento database and it can be re-used as many times as needed.

When we put a message in the queue, the message is submitted to our pre-defined *UrbanShip* account in order to be sent by the *UrbanShip AirMail* service to our users.

## There's more...

While we are on the **Message Queue** screen, if our message has not been sent yet, we can still cancel or delete it by choosing the appropriate action (**Cancel** or **Delete**) from the **Action** select box and clicking on the **Submit** button to perform the desired action for the selected message.



Thank you for buying
# Magento Mobile How-to

## About Packt Publishing

Packt, pronounced 'packed', published its first book "*Mastering phpMyAdmin for Effective MySQL Management*" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

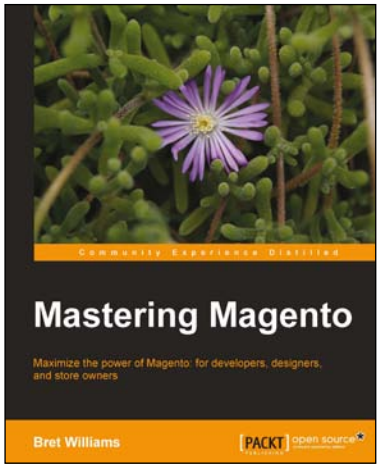
Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: `www.packtpub.com`.

## Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to `author@packtpub.com`. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.

[PACKT] PUBLISHING




## Mastering Magento

ISBN: 978-1-84951-694-5 Paperback: 300 pages

Maximize the power of Magento: for developers, designers, and store owners

1. Learn how to customize your Magento store for maximum performance
2. Exploit little known techniques for extending and tuning your Magento installation.
3. Step-by-step guides for making your store run faster, better and more productively.

---




## Magento 1.4 Themes Design

ISBN: 978-1-84951-480-4 Paperback: 292 pages

Customize the appearance of your Magento 1.4 e-commerce store with Magenoto's powerful theming engine

1. Install and configure Magento 1.4 and learn the fundamental principles behind Magento themes
2. Customize the appearance of your Magento 1.4 e-commerce store with Magento's powerful theming engine by changing Magento templates, skin files and layout files
3. Change the basics of your Magento theme from the logo of your store to the color scheme of your theme

Please check **www.PacktPub.com** for information on our titles

[PACKT] PUBLISHING



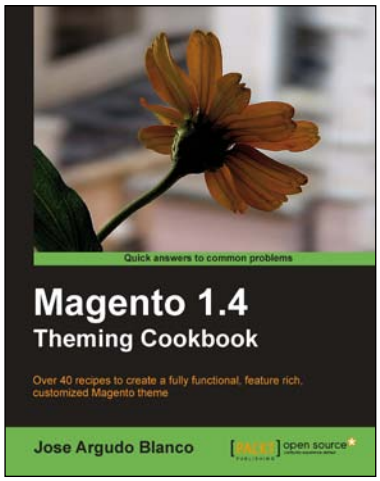

## Magento: Beginner's Guide

ISBN: 978-1-84719-594-4 Paperback: 300 pages

Create a dynamic, fully featured, online store with the most powerful open source e-commerce software

1. Step-by-step guide to building your own online store
2. Focuses on the key features of Magento that you must know to get your store up and running
3. Customize the store's appearance to make it uniquely yours
4. Clearly illustrated with screenshots and a working example

---




## Magento 1.4 Theming Cookbook

ISBN: 978-1-84951-424-8 Paperback: 200 pages

Over 40 recipes to create a fully functional, feature rich, customized Magento theme

1. Create rich, fully featured themes in easy to follow steps
2. Customize and localize your themes to make them sellable
3. Step-by-step recipes to help you solve problems related to Magento theming