DOWNLOAD PDF MAGENTO 2 FRONTEND DEVELOPER GUIDE

Chapter 1 : Buy Magento 2 Developer's Guide - Microsoft Store

Frontend Developer Guide Introduction. This document provides instructions for creating and installing custom storefront themes for a Magento application. It describes Magento's content rendering process and explains the view layer of the system to the extent required to build a theme efficiently.

For backend developers, several topics are covered that will enable you to modify and extend your Magento 2 store. Frontend developers will also find some coverage on how to customize the look of the site in the frontend. What You Will Learn Set up the development and production environment of Magento 2 Understand the new major concepts and conventions used in Magento 2 Build a miniature yet fully-functional module from scratch to manage your e-commerce platform efficiently Write models and collections to manage and search your entity data Dive into backend development such as creating events, observers, cron jobs, logging, profiling, and messaging features Get to the core of frontend development such as blocks, templates, layouts, and the themes of Magento 2 Use token, session, and Oauth token-based authentication via various flavors of API calls, as well as creating your own APIs Get to grips with testing Magento modules and custom themes, which forms an integral part of development In Detail Magento is one of the most exciting, flexible, and customizable e-commerce systems. It offers you an extensive suite of powerful tools for creating and managing an online store. The long-awaited Magento 2 release introduces a whole new e-commerce platform to develop online stores. The all new Magento 2 architecture , Web APIs , and a host of other features are equally challenging to master as much as they are exciting to use. This book will ease the learning curve by offering step-by-step guidance on how to extend the core functionality of your Magento 2 store. We start off with an introduction to the fundamental concepts of Magento to give you a foundation to work from. We then move on to configure the development and basic production environment for Magento. We then delve deeper to get to the core of automated deployments, persisting data, writing data fixture scripts and applying various backend and frontend modifications. As we near the end of the book, you will learn to make API calls and write automated tests. Finally, you will be guided through building a full-blown helpdesk module from scratch. By the end of this book, you will have learned a wide range of techniques to extend and customize your Magento 2 store to fit the requirements of your business. Style and approach This book is a mix of theoretical and step-by-step approaches, explained in a conversational and easy-to-follow style. Topics are explained sequentially, giving detailed explanations of the basic and advanced features to get you working on Magento 2. Table of Contents Chapter 1. Understanding the Platform Architecture Chapter 2. Managing the Environment Chapter 3. Programming Concepts and Conventions Chapter 4. Models and Collections Chapter 5. Using the Dependency Injection Chapter 6.

Chapter 2 : Magento 2 Developers Guide - PDF Free Download - Fox eBook

Magento Developer Documentation. Everything you need to build and manage a customized Magento store.

Creating Your Own Templates 7. Project Structure One of the biggest differences between the two versions of Magento is the way the whole project is structured. In this case, there is no longer a separate skin folder for theme assets. All of your templates and assets are now stored in: You will soon see that, unlike Magento 1. Each one of these folders represents a module in Magento 2 where templates or assets are overridden in this theme. Here is an example of the checkout module in a custom theme. However, instead of having all your sitewide assets and JS in this folder, such items are now organised per module â€" in this case, the Magento Checkout module. Those of you with a keen eye may also have noticed that there is a template folder in the web folder of this module. In this case, this is Catalog: If you need to override Controllers and Blocks, this is done in a different location and normally by a backend developer, which is not covered in this guide. Creating a Theme There are number of improvements to the way themes are managed and set up in Magento 2. The use of the theme. The fallback system in Magento 2 works in a similar way to Magento 1. However, Magento 2 allows you to have unlimited fallback levels and the method by which fallbacks work is much cleaner. You would then create a theme. Once on this page, you should see your new theme! You now need to select your theme. When on this page, click the design tab in the left-hand menu. From here, you can select your new theme, save the config, clear all your caches and your store will use your new theme. Overriding Templates One of the most common things a frontend developer will do when working with Magento 2 is to override a template. Once you know the location of the template you want to override, the same process applies to Magento 2 as it does with Magento 1. Copy the file from the original location and place it in the correct place in your theme so that Magento uses that file over the original. So in this case, we copy the original description. As a general note, you must always copy the original file across to your own theme, as template files are always overridden and not extended as with layout XML files. This basic rule of overriding templates applies throughout the whole of Magento 2. There are some exceptions, especially when it comes to the checkout process, but the basic logic behind even this is still similar. Blocks and Containers One of the biggest changes to the way that templating and layout changes work in Magento 2 vs Magento 1. This and other improvements to the layout system provides considerable added functionality. In most instances using Magento 1. This still works as it did in Magento 1. However, a container also has additional attributes such as htmlTag, which are then used to compile the template HTML when passed through to the Magento rendering engine. In Magento 2, you can simply move containers and blocks around to any other block or container that is defined in the XML! You can still define new layouts such as 1-column, 2-column etc. In Magento 2, layout files now extend files further down the fallback tree. You can simply create a new file of the same name, add in your adjustments, and everything will be compiled into one big XML file, including all the definitions in fallback themes. Moving Elements Around One of the best new features of the new templating and layout system in Magento 2 is the ability to move blocks and containers to different locations within the page. To move a block from one container or block to another, you simply need to use the following syntax: Although correct at the time of writing, the syntax for the move command is currently in review and may change in the future. Creating Your Own Templates Creating your own blocks and templates is more or less the same as in Magento 1. As with Magento 1. If you need to link this to another block type, you just switch it here as with Magento 1. Note that the syntax is slightly different. So, in this instance, it is calling the breadcrumbs. It is beyond the scope of this guide to discuss this at any length, but in general if you wish to override one of these files you can use the same logic as with overriding other templates. However, adding new templates and data is very different and will be discussed in a later guide. Say you wanted to override the registration template, which is shown at the end of the checkout process. This file is originally located here: The file extension used for the checkout template files is. We will publish another tutorial specifically on KnockoutJS at a later date. Conclusion The new templating system is a big improvement over Magento 1. The ability to move blocks and containers around is really a great feature that is now available. In terms of the whole project structure, this is also an improvement over the original Magento. Having templates and assets separated by module makes things much easier to find and modify and is a lot more logical than the Magento 1. Learn more about the Magento Commerce services we offer Developer looking for a new challenge? Check-out our job openings.

Chapter 3 : Magento 2 tutorial: whatâ€™s new for frontend developers? | Inviqa

Themes overview Themes overview. A theme is a component of Magento application which provides a consistent look and feel (visual design) for entire application area (for example, storefront or Magento admin) using a combination of custom templates, layouts, styles or images.

Default Themes Below is a screenshot of the default blank theme. The product filter on the category page on smaller device widths has been included underneath the products, which puts products higher in content priority, but could become annoying for users wishing to filter products if they have to scroll through all the products on the page first. This is where you can define the theme, but also define a fallback parent theme. Below is the Magento Blank example. In Magento 2, there are unlimited levels of parent themes you can inherit from, and they will fall back sequentially until there are no more parents defined. I used the Magento Blank theme as a parent and have titled it, which is essentially all you need to begin building your own theme. You have to visit this page before you can start using your theme and make it available for selection for your store. Image sizes on category, product, and basket pages can all be defined within this xml file by width, height, and the ratio see a short example below. Composer Another handy new feature is the ability to package multiple themes in Composer. Everything is contained within this theme or fallback themes , which is handy to keep everything together â€” and makes it much easier to exploit your theme, should you wish to. Below is an example of a module. In practice you will be creating your own Module folders, in your own theme. So for example to override: You can also have images here which are specific to the module. If you want to completely change the layout structure, you can override it with an override folder. That folder would then include all the layout information for that module, which can be modified to override any parent XML information. However, in Magento 2 this would be rarely used as layouts now inherit each other, unlike in Magento 1. Magento has instead chosen to have all of the XML within the module. If you want to modify XML for a module, you will have to make a new XML file with the same name in your equivalent theme directory and make the modifications there remember that all of the attributes and definitions from the original module will be inherited into this file. Another great feature of the new XML in Magento 2 is the the move action. Previously if you wanted to move a block somewhere, you had to unset the child and then insert it back in by referencing it further down in the layout XML. In Magento 2, you just need one line. Layout handles Usually, you would have a checkout XML with the different parts of the checkout included with the layout handles. In Magento 2, these are separate files. The layout instructions are much more logical now, which I will explain in more detail later in this guide. Magento currently uses structural and content blocks where the structural blocks act like placeholders. Content blocks then actually contain the content. In Magento 2, this has been changed to containers and blocks, which makes a bit more sense. Containers and blocks A container is a structure without content which holds other blocks and containers. By going back to extend and override layouts instead of making changes inlocal. However, there are numerous advantages of having it in XML vs a template. If you are just extending one of the inbuilt themes such as the blank theme , to customise the CSS you simply override the. So for example if you wanted to override the checkout CSS, originally defined in the following file: So, it would look something like this: If you still want to keep all of these or some of the original files, you need to copy the contents of the original over to your override file. It also has a built-in fallback mechanism so it will search for the LESS files based on the hierarchy you have set up in the theme. You can also add your own custom CSS files if you wish. However, there are various methods of doing this which will be discussed in later guides. Browser support Magento 2 supports IE9 and upwards, which is great. All the JavaScript loading is now handled by requireJS, meaning you no longer have to worry about including the script tags yourself, or checking if a file is loaded in the correct order, as RequireJS makes sure all your dependencies are loaded before your code is executed. Documentation Magento recommends using the base theme as the parent of your custom theme, but from reading feedback from the community so far, it seems a lot of people have preferred to ignore that and just create a whole theme from scratch. Conclusion Overall, Magento 2 is definitely a big improvement over the original Magento 1. The improvements in the JavaScript are one of the biggest changes as you can now easily add and extend new and existing JS to meet your needs, without having to worry about loading all the dependencies and files yourself. Related reading When to upgrade to Magento 2 Got questions about Magento ecommerce development? Article update This article was originally published under Session Digital, which unified with Inviqa in June Commerce Technical tutorials Categories.

Chapter 4 : Magento 2 Developer's Guide Download

We just published our third annotated study guide: for Magento's new Frontend Developer test. It contains pages with detailed answers to the.

Chapter 5 : Magento 2 tutorial: how to use the new frontend templating system | Inviqa

The SwiftOtter Frontend Developer Exam Preparation Guide has pages detailing literally every aspect of customizing Magento's frontend. The goal of the study guide is to fill in knowledge voids. This not only makes you a better developer, but will help you pass the test.

Chapter 6 : Magento 2 Block Template Ultimate Guides - Devdocs â€“ Mageplaza

Check this video and watch how to create magento 2 theme. It is always good to work with your custom theme.

Chapter 7 : Magento 2 Developer's Guide | PACKT Books

The Magento 2 Certified Professional Front End Developer exam, a primarily scenario-based exam, is designed to validate the skills and knowledge needed to understand Magento's theming components and the ability to modify the user interface according to best practices.

Chapter 8 : Magento 2 Certified Frontend Developer Study Guide - Magento Forums

I wanted to share the completion of our latest project for Magento developers: the frontend developer study guide. It's over pages with concentrated information on customizing the Magento frontend. While it was originally intended as a study book for the Frontend Developer test, it is one of the.