Magento Developer Certification Preparation Study Group

Participant Guide

Contents

Overview

Welcome to the Program!

Welcome to the Magento Developer Certification (MDC) preparation study group program! Thank you for signing up to be a part of a study group to prepare for the Magento Developer Certification exam.

Two Magento Developer Certification Exam Choices

Magento Developer Certification is a mid-advanced level certification geared toward professional developers that have real-world experience with Magento implementations. The exams provide a way for qualified, experienced Magento developers to demonstrate and validate their skills. Developers who pass one of the exams can use the valued Magento Certified Developer credential in marketing their services.

There are two exams in the Magento Certified Developer portfolio.

Magento Certified Developer

Magento Certified Developer Plus

For more information about the Magento Developer Certification program, see the website at www.magentocommerce.com/certification/.

The final section of this guide provides next steps for how to sign up to take the test after you complete your study group meetings.

About This Guide

The purpose of this guide is to give you access to the certification objectives and study group exercises for each meeting. In this guide, the certification objectives are grouped by the associated meeting of the study group.

The certification objectives are the same as those found in the Magento Certified Developer Exam *Study Guide*. You can request a PDF file of the *Study Guide* on the certification resources page at www.magentocommerce.com/certification/.

The questions related to the study guide objectives have been omitted from this guide to avoid confusion with the study group exercises. Please refer to the Magento Certified Developer Exam *Study Guide* to check your knowledge against those questions.

The main value of the study group comes from working through the exercises together with the other participants preparing for the exam.

Group Discussion Tool

The moderator of your study group will choose a group discussion tool for you and the other participants to use during the course of the study group. When you receive your study group information, your moderator should provide you with information about how to access and use the discussion tool. Please use the tool, both to ask questions and to give answers. This is a technique called social learning. The meetings serve as a focal point and to provide moral support and motivation in addition to the group discussions.

If you have any questions regarding the study group itself (in contrast to the technical content), please contact the group moderator. She or he will help, for example with technical support for the group discussion tool.

Code Examples

For some exercises, solutions will be provided during the meetings. Please be aware that the code from the exercise solutions is not production ready. It is distributed to serve as an educational example, but in some cases error checking or something similar might have been neglected.

Meeting 1

Study Guide Objectives

This meeting doesn't cover any technical content, so there are no associated study guide objectives.

Exercises

This meeting doesn't cover any technical content, so there are no exercises.

Meeting 2

Study Guide Objectives

The study guide sections covered in this meeting are **Basics** and **Request Flow**.

Basics

High-level Magento architecture

- Describe Magento code pools
- Describe typical Magento module structure
- Describe location of Magento templates and layout files
- Describe location of Magento skin and JavaScript files
- Identify and explain the main Magento design areas (adminhtml and frontend)
- Explain class naming conventions and their relationship with the autoloader
- Describe methods for resolving module conflicts

Magento configuration

- Explain how Magento loads and manipulates configuration information
- Describe class group configuration and use in factory methods
- Describe the process and configuration of class overrides in Magento
- Register an observer
- Identify the function and proper use of automatically available events, including `*_load_after`, etc.
- Set up a `cron` job

Internationalization

- Describe how to plan for internationalization of a Magento site
- Describe the use of Magento translate classes and translate files
- Describe the advantages and disadvantages of using subdomains and subdirectories in internationalization

Request Flow

Application initialization

- Describe the steps for application initialization
- Describe the role of the system entry point, `index.php`

Front controller

- Describe the role of the front controller
- Identify uses for events fired in the front controller

URL rewrites

- Describe URL structure/processing in Magento
- Describe the URL rewrite process

Request routing

- Describe request routing/request flow in Magento
- Describe how Magento determines which controller to use and how to customize route-to-controller resolution

Module initialization

- Describe the steps needed to create and register a new module
- Describe the effect of module dependencies
- Describe different types of configuration files and their load priorities

Design and layout initialization

- Identify the steps in the request flow in which:
    - Design data is populated
    - Layout configuration files are parsed
    - Layout is compiled
    - Output is rendered

Flushing data (output)

- Describe how and when Magento renders content to the browser
- Describe how and when Magento flushes output variables using the front controller

Exercises

Exercises for the **Basics** and **Request Flow** sections.

- Rewrite the `sales/order` model to add the customer group model as an email template variable in the `sendNewOrderEmail()` method, so the group code can be added to the email using `{{var customer_group.getCode()}}`.
- Create an observer that redirects the visitor to the base URL if the CMS home page URL key is accessed directly (i.e. /home -> /).
- Add a new frontend route and create an index controller and an index action that set the return value of `$this->getFullActionName()` to the response body.
- Rewrite the `Mage_Customer_AccountController::loginAction()` method to set a category view of your choice as the `before_auth_url`.
- Create a dynamic rewrite of the `payment/data` helper only if the version of Magento is older than version 1.4, and if the `ccsave` payment option is enabled for the current store.
- Create a diagram of the important classes involved in a request dispatch.
- Magento can use several ways to specify the current store view for a given request. List the priority of all the different ways.

Meeting 3

Study Guide Objectives

The study guide sections covered in this meeting are **Rendering** and **Widgets**.

Rendering

Themes in Magento

- Define and describe the use of themes in Magento
- Define and describe the use of design packages
- Describe the process of defining template file paths

Blocks

- Describe the programmatic structure of blocks
- Describe the relationship between templates and blocks
- Describe the stages in the lifecycle of a block
- Describe events fired in blocks
- Identify different types of blocks
- Describe block instantiation
- Explain different mechanisms for disabling block output
- Describe how a typical block is rendered

Design layout, XML schema, and CMS content directives

- Describe the elements of Magento's layout XML schema, including the major layout directives
- Register layout XML files
- Create and add code to pages
- Explain how variables can be passed to block instances via layout XML
- Describe various ways to add and customize JavaScript to specific request scopes

Widgets

Frontend widgets

- Create frontend widgets and describe widget architecture

Exercises

Exercises for the **Rendering** and **Widgets** sections.

- Using layout XML, add a custom block with a template that displays the current date in the left column of all pages.
- Using layout XML, remove the checkout link from the top navigation.
- Change the root template to 1column.phtml on the customer account page, using layout XML.
- In a custom action controller, render only the contents of a single custom update handle.
- Rewrite the breadcrumbs block and add a hardcoded crumb at the beginning without modifying the template. **Note:** Do not use the `_toHtml()` method to implement this customization.
- Add an additional hardcoded theme called “primary” that is always first in the fallback order in the configured package. Then add an additional fallback theme base/[configured-default-theme] that is always evaluated just before the base/default fallback. Create a test action that allows you to test all 6 possible template fallbacks at once.
- Create a frontend widget with a JavaScript color picker form element and image upload functionality in the widget form. When rendering, the widget displays the image with a border of the chosen color. You can use the color picker library found in the js/jscolor/ directory in the Magento folder. For documentation, refer to the project home page at http://jscolor.com.
- Add a default instance of the widget to the left column automatically when your module is installed. Use a default image.

Meeting 4

Study Guide Objectives

The study guide sections covered in this meeting are **ORM**, **Database,** and **Setup Scripts**.

ORM and Database

Models, resource models, and collections

- Describe the basic concepts of models, resource models, and collections, and the relationship they have to one another
- Configure a database connection
- Describe how Magento works with database tables
- Describe the load-and-save process for a regular entity
- Describe group save operations
- Describe the role of `Zend_Db_Select` in Magento
- Describe the collection interface (filtering/sorting/grouping)
- Describe the hierarchy of database-related classes in Magento
- Describe the role and hierarchy of setup objects in Magento

Setup Scripts

Install / upgrade scripts

- Describe the install/upgrade workflow
- Write install and upgrade scripts using setup resources
- Identify how to use the DDL class in setup scripts

Exercises

Exercises for the **ORM, Database,** and **Setup** sections.

- Create a custom entity table for comments using a setup script.
- Create a comments entity model, flat table resource model, and a matching resource collection model.
- Create a data upgrade script that uses the comments model classes to populate the database.

- Create a frontend form block that allows visitors to add comments to category and product pages.
- Use the `_beforeSave()` method to automatically set the associated page and an `updated_at` timestamp on the comment records.

Meeting 5

Study Guide Objectives

The study guide section covered in this meeting is **EAV**.

EAV

EAV model concepts

- Define basic EAV concepts and class hierarchy
- Describe the database schema for EAV entities
- Describe the EAV entity structure and how it differs from the standard core resource model
- Describe the EAV load-and-save process and how it differs from the regular load-and-save process

Attribute management

- Identify the purpose of attribute frontend, source, and backend models
- Describe how to implement the interface of attribute frontend, source, and backend models
- Describe how to create and customize attributes

Exercises

Exercises for the **EAV** section.

- Use a setup script to add a `catalog/product` select attribute using the Magento `eav/entity_attribute_source_table`. Also add some options.
- Add a `customer/customer` multiselect attribute using a custom source and backend model. Make it possible to unset all options for that attribute on customers using the Admin interface.
- Add a `catalog/product` select attribute with a custom source model that is indexed by the flat table indexer, and that also works in the layered navigation.
- Create an EAV version of the comments entity implementation from last week's exercise.
- Use a backend model to set the associated page and `store_id` entity properties.

- Create an uninstall script (which would need to be run manually) for the comments module that leaves Magento in a state as if the module had never been installed.

Meeting 6

Study Guide Objectives

The study guide section covered in this meeting is **Adminhtml**.

Adminhtml

Common structure/architecture

- Describe the similarities and differences between adminhtml and frontend interface and routing
- Describe the components and types of cache clearing using the adminhtml interface

Forms

- Define form structure, form templates, grids in Magento, and grid containers and elements

Grids

- Create a simple form and grid for a custom entity
- Describe how to implement advanced adminhtml grids and forms, including editable cells, mass actions, totals, reports, custom filters and renderers, multiple grids on one page, combining grids with forms, and adding custom JavaScript to an admin form

System configuration

- Define the basic terms, elements, and structure of system configuration XML
- Describe system configuration scopes

Access Control Lists (ACL) and permissions

- Define/identify basic terms and elements of ACL
- Use ACL to:
  - Set up a menu item
  - Create appropriate permissions for users
  - Check for permissions in permissions management tree structures

Working with extensions

- Describe how to enable and configure extensions
- Define Magento extensions and describe the different types of extensions available (Community, Core, Commercial)

Exercises

Exercises for the **Adminhtml** section.

- Create an adminhtml integration for the flat table comments entity that supports all CRUD operations.
- Create the same for the EAV version, utilizing the semi-automatic adminhtml form field generation through the `adminhtml/widget_form` block.
- Create a system configuration option with a new system.xml section that enables/disables the adminhtml option to create new comments.
- On the product edit page, create a new tab containing a grid of all comments associated with this products detail page. Make records removable directly from the grid as a mass action. For this you may want to add a `product_id` attribute to the comment entities.
- In the comments grids, use a renderer to ensure that the backend display is XSS safe.
- In the main comments grid, display the entity type associated with the comments page. Create a custom column renderer and filter for the column display and filter.
- Create a custom system configuration control that allows you to render your own template.
- Create a custom 24-hour and minute time picker control for the comments entity using a custom form element type.

Meeting 7

Study Guide Objectives

The study guide section covered by this meeting is **Catalog**. It also contains exercises for the Magento indexing subsystem.

Catalog

Product types

- Identify and describe standard product types (simple, configurable, bundled, etc.)
- Create custom product types from scratch or modify existing product types
- Identify how custom product types interact with indexing, SQL, and underlying data structures

Price generation

- Identify basic concepts of price generation in Magento
- Modify and adjust price generation for products (for example, during integration of third-party software) – price, special price, group price, tier price etc.

Catalog structure

- Describe the Category Hierarchy Tree Structure implementation (the internal structure inside the database), including:
    - The meaning of parent_id `0`
    - The construction of paths
    - The attributes required to display a new category in the store

Other skills

- Choose optimal catalog structure (EAV vs. flat) for a given implementation
- Implement, troubleshoot, and modify Magento tax rules
- Modify, extend, and troubleshoot the Magento layered (“filter”) navigation
- Troubleshoot and customize Magento indexes
- Describe custom product options in Magento

Exercises

Exercises for the **Catalog** section.

- Rewrite the catalog search full-text indexer to include product tags.
- Create a custom indexer with a custom index for comment to category associations. This index contains references to all the comments associated with a specific category detail page. You might want to add a `category_id` attribute to the comment entities for this.
- Integrate the comment indexer into the adminhtml indexer interface.
- Make the comment indexer process individual entity updates immediately (if not set to manual).
- Set the indexer status to “requires reindex” whenever a comment on the frontend is edited, and update automatically whenever a category is edited in the admin area.
- Create an action controller action method that adds a configurable product to the cart using a default set of options (single click add-to-cart for configurable products).
- Create a product type that lets the customer specify the price he wants pay when adding such a product to the cart, and use that price during checkout. Use the standard product price attribute as a suggested price for the product.

Meeting 8

Study Guide Objectives

The study guide sections covered in this meeting are **Catalog Price Rules**, **Shopping Cart Price Rules,** and **API**.

Price Rules

Catalog price rules

- Identify how catalog price rules are implemented in Magento

Shopping cart price rules

- Describe how shopping cart price rules work and how they can be customized

API

API

- Use the Magento API to implement third-party integrations
- Extend the existing Magento API to allow for deeper integrations into third-party products
- Describe the different Web Service APIs available within the Magento core
- Describe the advantages and disadvantages of the available Web Service APIs in Magento
- Identify the configuration files used for the v2 SOAP API
- Describe the purpose of the configuration files related to the API

Exercises

Exercises for the **Price Rules** and **API** sections.

- Create a coupon programmatically that can be used multiple times on specific product types.
- Automatically add and remove coupons from the cart items.
- Implement an automatic import of coupon codes from a third-party system. Use a simple scenario where there is a file in which codes, discounts, and products are listed.
- Implement a custom condition for sales price rules that activates a sales price rule on a specific weekday.

- Implement a `getOnlineCustomers()` API resource action that returns the number of customers currently online. Make it work in the SOAP v1 and v2 (non WSI-compliant) flavors of the Magento API for authorized users who are assigned to a role that includes this action.
- Implement a `getInstanceVersion()` resource action that returns the Magento version. Make it work in the SOAP v1 and v2 (non WSI-compliant) flavors of the Magento API for authorized users who are assigned to a role that includes this action.

Meeting 9

Study Guide Objectives

The study guide section covered in this meeting is **Checkout**.

Checkout

Checkout components

- Describe how to modify and effectively customize the quote object, the quote item object, and the address object
- Explain the total models architecture

Magento multishipping implementation

- Describe how to extend the Magento multishipping implementation
- Identify limitations of the multishipping implementation

Exercises

Exercises for the **Checkout** section.

- Modify the Onepage-Checkout to skip the shipping step, if only one shipping option is available.
- Implement a module to do total adjustments based on the selected payment option (for example, +4% for credit card payments). Base the adjustment on the quote subtotal. Display the adjustment amount for each payment method beside each method on the select payment step. Implement a complete set of total models to display and add the surcharge. Make sure the additional cost is displayed during checkout, in the frontend order history, in the admin order view, in the admin invoice view, and in the admin printable PDFs.
- Add additional custom product options as labels to quote items on the fly when specific products are added to the cart. Make them work with reorders.
- Create gift functionality: A merchant can mark a product to be added as a gift if some condition is satisfied. For example a condition can be purchasing some other product, or a certain amount of other products. Even though the marked product is added as a free gift, the same product should still be available on the frontend for regular purchase.

The tasks are:

1. Automatically add the gift with a price of zero to the cart.
2. Check for consistency (you can't have the gift if the condition isn't satisfied, no more than one gift, and so on).
3. Process the scenario where, in addition to the gift, there is also another item of the gift product added to the cart.

- Add a new attribute to a product; for example, `supplier_sku`. Pass this attribute to the order. Make sure the attribute is copied to the quote items and to order items. Customize the review step of the checkout and display the new attribute there.
- Implement a custom restriction for certain products during checkout. For example, customers whose billing or shipping address is on a blacklist (or whose country is not supported) won't be able to order this item.
- Implement the possibility of a discount on the order total when shipping to a specific region. Implement this by registering your own total model. Check that the discount also works in case of an admin order.

Meeting 10

Study Guide Objectives

The study guide sections covered in this meeting are **Orders, Invoices, Shipments, Credit Memos,** and **Shipping Methods**.

Sales

Orders, invoices, shipments

- Describe order creation in the admin
- Describe the differences in order creation between the frontend and the admin
- Describe the order shipment structure and process
- Describe the architecture and processing of refunds
- Describe the implementation of the three partial order operations: partial invoice, partial shipping, and partial refund
- Describe cancel operations

Shipping Methods

Shipping methods

- Describe the programmatic structure of shipping methods, how to customize existing methods, and how to implement new methods
- Describe the shipping rates calculation process

Exercises

Exercises for the **Orders, Invoices, Shipments, Credit Memos**, and **Shipping Methods** sections.

- Implement a module that saves all sales CRUD operations (frontend and adminhtml) into a queue table where they could be picked up by an ERP system.
- Implement a shipping module that uses the new product attributes `height`, `width`, and `depth` to calculate the shipping cost. Respect items with free shipping from coupons.
- Automatically attach the invoice PDF with the order confirmation email (backend and frontend).

Meeting 11

Study Guide Objectives

The study guide section covered in this meeting is **Payment**.

Payment

Payment methods

- Describe the programmatic structure of payment methods and how to implement new methods
- Describe card operations (capturing and authorization)

Exercises

Exercises for the **Payment** section.

- Create a dummy payment module that works with the one-page checkout and the multi-address checkout. Have it set the new order into a freely configurable state (with the assigned default status) automatically.
- Create a payment module that uses an external budget through a mock API. Make it require a redirect to a third-party gateway website.

Meeting 12

Study Guide Objectives

The study guide section covered in this meeting is **Customer**.

Customer

Customer

- Describe the architecture of the customer module
- Describe the role of customer addresses
- Describe how to add, modify, and display customer attributes

Exercises

Exercises for the **Customer** section.

- Add a Boolean select customer address attribute `personal_delivery` that appears on all customer address forms, frontend and backend.
- Add a customer profile picture attribute and an option to set/unset it in the frontend and backend on a separate tab.

Next Steps: Magento Developer Certification Test

The sections below provide more information about the two Magento Certified Developer tests you can take, and instructions for how to register for the test. For more information about the Magento Certified Developer program, see the website at www.magentocommerce.com/certification/.

The Ultimate Magento Credential

Magento Developer Certification is a mid-advanced level certification geared toward professional developers that have real-world experience with Magento implementations. The exams provide a way for qualified, experienced Magento developers to demonstrate and validate their skills. Developers who pass one of the exams can use the valued Magento Certified Developer credential in marketing their services. There are two exams in the Magento Certified Developer portfolio.

Magento Certified Developer

A Magento Certified Developer can skillfully use all business processes in Magento, such as:

- Structure of catalog, indexes, promotions, price generation logic
- Architecture of checkout, payment/shipment methods, sales/order processing
- Advanced core knowledge — forms/grids full functionality, API, widgets, etc.
- A Certified Magento Developer can make design decisions on the code level, including how to parse data files, steps of import, data verification, logging (etc.)

About the Exam

- 70 multiple-choice questions
- 90 minutes to complete the exam
- Based on Magento Community Edition version 1.7.0.2
- No hard-copy or online materials may be referenced during the exam

- No prerequisites; we recommend taking Fundamentals of Magento Development course as a first step
- *Study Guide* available

Magento Certified Developer Plus

The Magento Certified Developer Plus is skillful in the use of Magento Enterprise Edition and has delved into details of the structure of Magento that many have not often explored, but which can give someone an advantage when implementing a site using Magento.

About the Exam

- 85 multiple-choice questions
- 120 minutes to complete the exam
- Questions relating to the Magento Community Edition are based on v.1.7.0.2, and questions relating to Magento Enterprise Edition are based on v.1.12.0.2
- No hard-copy or online materials may be referenced during the exam
- No prerequisites; we recommend taking Fundamentals of Magento Development course as a first step
- *Study Guide* available

Purchase Your Voucher and Register for the Exam

You can choose which exam to take, and purchase your exam voucher, at www.magentocommerce.com/certification/.

After you purchase your voucher, you register to take the exam.

Magento Certification exams are delivered through a network of Prometric testing centers worldwide. There are three ways to schedule a test appointment:

- Online: http://www.register.prometric.com/Index.asp?tcl=0&cookie%5Ftest=1
- Onsite at your local Prometric test center.
- Call Prometric's Regional Registration Center: 866.664.9501

Information Needed to Schedule Your Exam Appointment

Please be prepared to provide the following information when scheduling your appointment:

- **The name you use when scheduling your appointment must exactly match the name on your identification,** which you will be asked to present on the day of your exam.
- At a **minimum**, your identification must be a valid government-issued ID that shows your name in the English alphabet, your signature, and your photograph (for example, driver license, passport, and so on).
- Your Social Security, Social Insurance, or Prometric Testing I.D. (SP#) Number.
- Contact phone number(s) – If there is a problem, we will use these number(s) to reach you.
- Mailing address – Please provide the address to which you would like your certification certificate mailed.
- Exam number and title:
    - M70-101: Magento Certified Developer
    - M70-201: Magento Certified Developer Plus
- Email address – Please use the email address associated with your Magento account so that your testing record can be properly matched.
- Magento Account ID Number – Your Magento Account ID Number is MAG000xxxxx. This information can be accessed by clicking on the My Account link, which appears on the top right of any Magento website page once you have logged in to your Magento Account.
- Method of payment – You are prompted to enter your voucher number.

What to Bring With You to Your Exam Appointment

You must have two valid forms of ID, one with a photo and both with signature. Forms of primary ID must be original, non-expired (valid), photo-bearing, signature-bearing, and government-issued. Visit www.prometric.com/TestTakers/FAQs for more details.

If you need to reschedule or cancel your exam appointment please go to the Prometric website and review the information there.

Your Exam Results

Upon completion of the exam, your test results will be displayed on the screen and then printed and provided to you as you leave the testing center.

While we hope that you will pass the exam the first time, if you do not, Magento offers a certification retake discount. For information on the discount and our retake policy, please see www.magentocommerce.com/certification.

Questions? Comments? Contact us at certification@magento.com

Thank you for participating in the study group, and good luck!