

Restructuring the MODX Documentation

At <https://docs.modx.org/> you can find a preview of new MODX documentation. All the content has been synchronised with the official docs.modx.com on March 5th 2019, which is available in Markdown at <https://github.com/Mark-H/Docs>

To make the docs better, we also need to re-think the structure. What are the questions people have at different points of working with MODX, and at different skill levels, and how do we help them get there the fastest? What documentation needs to be rewritten and what is missing?

On the next few pages, you'll find a proposed navigation structure/sitemap. [Text between brackets is explanatory.] Please add comments if you have any suggestions.

Proposed Navigation Structure

- Getting Started [What is MODX overview, installation, upgrades, introduction to template tags and elements]
 - What's MODX? [Explain CMS/CMF functionality, what creative freedom means, who it is for]
 - Server Requirements
 - Installation [Standard installation guide]
 - Advanced installation
 - CLI installation [running setup from the CLI with the XML configuration]
 - Installing from git [both a direct git clone and the composer create-project approach]
 - Troubleshooting [common installation problems and how to interpret error messages]
 - Modsecurity
 - Getting started [tutorial explaining a total newbie where to look for things, creating a template and some resources]
 - Friendly URLs [explaining the modx side of the setup, settings, alias_path, etc)
 - Setup for apache [htaccess snippet]
 - Setup for nginx [config]
 - Setup for lighttpd [config]
 - Maintenance
 - Securing MODX [hardening guide, but cleaned up and simplified]
 - Moving a MODX site [explaining how to move a site in its entirety]
 - Upgrading MODX
 - Upgrading to 2.7 [highlights and things to consider when upgrading to this release; one page per minor version]
 - Upgrading to 2.6
 - Upgrading to ...
 - Glossary
 - Directory structure & key files
- Building Sites [Site structuring, detailed tag reference, using extras, TVs, detailed elements]
 - Tag Syntax
 - Output Filters/Modifiers
 - Mosquito Syntax
 - Caching
 - Date formats
 - Resources [documents]
 - Symlinks

- Weblinks
 - Static Resources
 - Content Types [html/css/js etc, impact on url]
 - Custom resource types [explaining what they are and what they can do, providing references to e.g. Collections and MoreGallery, plus link to development section on how to build them]
- Elements
 - Templates [including media source and tv assignments]
 - Template Variables [including media sources and template assignments]
 - Input Types [include link to TV section in Extending MODX, and detailed information about each available core input type, perhaps a section on popular third party types too]
 - Output Types
 - Bindings [FILE, SELECT, etc]
 - Chunks
 - Snippets
 - Plugins
 - Static Elements
- Contexts
 - Routing contexts [referring to extras, custom plugin, and index.php solutions]
- Media Sources [explaining abstraction layer, concepts, available media source types, file manager]
- Extras/Add-ons [where to get and how to install, package providers]
- Internationalisation [lexicons, multiple contexts, relevant extras]
- Integrating Templates [building templates, chunks, using snippets, best practices, etc - maybe this should be in the form of a guide/tutorial]
- Managing Content [discussing ways to change how content is managed]
 - Form Customization
 - Manager themes?
 - Dashboards [dashboard widgets for user groups]
- Properties & Property Sets
- Settings [explaining setting inheritance from system > context > usergroup > user, where to change settings]
 - ... list of settings, perhaps per area to not have a hundred pages each 3 lines long...
- Tutorials
 - Creating a blog
 - Loading Pages in the Front-End via AJAX and jQuery Tabs
 -
- Extending MODX [Writing snippets, writing plugins, CMPs, transport packages,]
 - The MODX object [\$modx object and what is commonly used in what way]
 - Loading \$modx externally

- Requests & Responses [sendRedirect, sendForward, sendError, request/response objects]
 - Element/templating utilities [getChunk, setPlaceholder]
 - Logging [setLogLevel, log]
 - Session handling [\$ _SESSION, configuration]
- Snippets
 - Templating [no markup in snippets, use chunks]
 - Registering CSS and JS
- Plugins & Events
 - Available Events
 - Priorities
 - Triggering custom events
- Setting up a development environment [Setting up a project directory outside the MODX root to use with an IDE/git, relevant settings and other aspects]
- xPDO
 - Getting started [explaining how to load data from existing models, how to change values and save that]
 - Retrieving objects [xPDOQuery, getObject/getCollection/getIterator]
 - Complex queries [xPDOQuery, xPDOPCriteria, where, andCollection, orCollection]
 - Graphs [getObjectGraph, getCollectionGraph]
 - Related objects [getOne, getMany, direct accessors]
 - Lazy loading
 - Using PDO [xPDO->query/fetch modes]
 - Setting object fields [set, save, updateCollection]
 - Using Transactions [beginTransaction/commit]
 - Removing objects [remove, cascading remove for composites, removeCollection]
 - Custom Models
 - Define a schema [reverse engineer from a table or write by hand, indexes, relations]
 - Field types [reference of available phptypes/dbtypes and how xPDO transforms them automatically]
 - Indexes
 - Relations
 - Validators
 - Schema versions [1.0, 1.1, differences and how to recognise/upgrade]
 - Inheritance [basic object extending, plus class_key based single inheritance]
 - Generate the model and map files [explaining difference between models and maps]
 - [3.x] Autoloading models

- Loading your package
 - Handling migrations [xPDOManager to add/alter/remove fields/tables/indexes]
 - Multiple Database Connections [primary/secondary/failover, mutable]
- Caching [modCacheManager]
 - Custom caching [expirations, partitions]
 - Using memcached [configuration/code]
 - Using redis [configuration/code]
- Namespaces
- Internationalization
- Menus [include some background on actions too]
- Custom Manager Pages
 - Step-by-step Tutorial [2.3+ style only]
 - ExtJS [adding a tabbed panel with some grids and a window]
 - modExt reference
 - Tutorials
 - Basic HTML [ParsedManagerController]
- Custom TV Types [2.2+ style only]
- Custom Resource Types [tutorial]
- Custom User types [explain when to use and how]
 - Tutorial
- Processors
 - Using runProcessor [\$modx->runProcessor]
 - From AJAX requests [connectors]
- Consuming (REST) APIs [modRest]
 - Migrating from modRestClient
- Building a REST API [modRestServer]
- Transport Packages [explain concept, vehicles, validators, resolvers“shipping containers”]
 - Vehicles
 - Validators
 - Resolves
 - Dependencies
 - Setup options
 - Custom build script tutorial
 - MyComponent?
 - Teleport?
- Core Model Reference [detailed reference/explanation of core models that people commonly interact with, plus information on how to get details on other models]
 - modResource [explaining process(), getTVValue/setTVValue)
 - modUser/modUserProfile [password handling/verification, blocking, get/setSessionContext, reloading session info]

- modSystemSetting/modContextSetting/modUserSetting [primarily explain the caching that's involved when changing these, and how setOption is temporary]
- User Guide [Explaining MODX to non-technical site editors]
- Contribute [Explaining how to contribute to MODX and the docs]
 - Reporting issues on GitHub
 - Triaging issues
 - Building the Core [technically how to help build the core]
 - Coding standards
 - Using git & GitHub
 - Running the unit tests

Unsure where to place in the tree:

- ACLs/users/user groups - does that fall under "Building Sites"?
 - User groups
 - Roles
 - Policies
 - Policy templates
 - Permissions
 - Permission groups (from 3.x)
 - Resource groups
 - Anonymous usergroup/users
 - Super users
 - Guides
 - Limited manager users
 - Restricting elements
 -
-