

# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Goals of this certification

The Mautic Developer Certification is designed to validate a developer's skills and knowledge in customizing, extending, and integrating with the Mautic marketing automation platform. This certification demonstrates proficiency in Mautic's architecture, API, plugin development, and theming.

Certified developers can build robust solutions, contribute to the Mautic ecosystem, and enhance Mautic's functionality to meet specific business needs.

The curriculum covers a range of topics, from basic Mautic concepts to advanced development techniques. It includes hands-on exercises and real-world scenarios to ensure developers gain practical experience.

Successful completion of the certification exam signifies a high level of competence in Mautic development.

### Prerequisites

To study for and pass this certification, you'll need to have:

- Basic PHP knowledge (OOP concepts, Symfony framework familiarity is preferred)
- Understanding of web development (HTML, CSS, Javascript)
- Database concepts (MySQL/MariaDB)
- Command line interface experience
- Git version control basics
- Familiarity with Codeception and PHPUnit testing frameworks



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 1: Prerequisites for Mautic

#### Learning objectives

- Understand system requirements for Mautic
- Identify necessary server configurations
- Learn about PHP, database, and web server requirements

#### Context

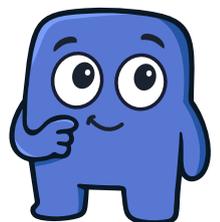
Before diving into Mautic development, it's crucial to understand the technical prerequisites and system requirements. This module covers the foundation needed to run Mautic effectively in development and production environments.

#### Key topics

- PHP version requirements (8.1+)
- Database requirements (MySQL 5.7+/MariaDB 10.2+)
- Web server configuration (Apache/Nginx)
- PHP extensions and configurations
- Memory and performance requirements
- Security considerations

#### Official Reference Links

- [Installation Requirements](#)
- [Mautic Official Website](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 2: Setup and installation using DDEV

#### Learning objectives

- Set up a local development environment using DDEV
- Configure Mautic for development
- Understand the benefits of containerized development

#### Context

DDEV provides a standardized, containerized development environment that ensures consistency across different development setups. This module teaches how to leverage DDEV for efficient Mautic development.

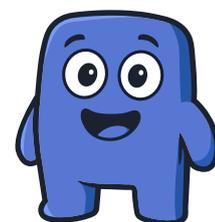
#### Key topics

- DDEV installation and configuration
- Creating a Mautic project with DDEV
- Database setup and management
- Environment configuration
- Debugging setup with DDEV
- Profiling code performance with Xhprof, Blackfire or Xdebug
- Performance optimization in development
- Unit testing setup in Mautic

#### Official Reference Links

- [DDEV Installation Guide](#)
- [Local Development Setup](#)

Note: DDEV is an open source project just like Mautic. If you find their tools valuable, you can find out more about supporting them [here](#).



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 3: Mautic platform walkthrough

#### Learning objectives

- Navigate the Mautic user interface
- Understand core Mautic concepts and terminology
- Identify key features and functionalities

#### Context

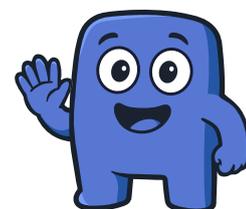
Understanding the Mautic platform from a user perspective is essential for developers. This module provides a comprehensive overview of the platform's capabilities and structure.

#### Key topics

- Dashboard and navigation
- Contacts and segments
- Campaigns and email marketing
- Landing pages and forms
- Reports and analytics
- Configuration and settings
- User management and permissions

#### Official Reference Links

- [Getting Started](#)
- [User Interface Guide](#)
- [Managing Contacts](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 4: Configuring and extending Mautic

#### Learning objectives

- Configure Mautic for different environments
- Understand configuration management
- Learn about extending Mautic's functionality

#### Context

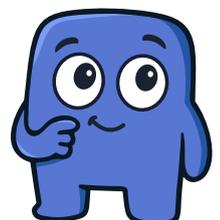
Proper configuration is crucial for Mautic's performance and functionality. This module covers how to configure Mautic for various use cases and how to extend its capabilities.

#### Key topics

- Configuration file structure
- Environment-specific configurations
- Email configuration and sending
- Integration configurations
- Performance and caching settings
- Security configurations
- Multi-tenancy considerations
- Familiarity with production environment configuration

#### Official Reference Links

- [Configuration Settings](#)
- [Email Configuration](#)
- [System Configuration](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 5: Developing with Mautic

#### Learning objectives

- Understand Mautic's architecture and coding standards
- Learn about the Symfony framework integration
- Develop custom functionality within Mautic

#### Context

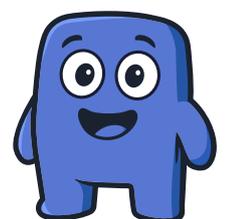
Mautic is built on the Symfony framework, which provides a robust foundation for development. This module introduces developers to Mautic's specific development patterns and best practices.

#### Key topics

- Mautic architecture overview
- Symfony framework integration
- Bundle structure and organization
- Service container and dependency injection
- Event system and hooks
- Form handling and validation
- Security and authentication
- Translations
- Integration with Third-Party authentication systems for fetching data from other services
- Configurations for Bundle
- Familiarity with IntegrationsBundle to add new integrations

#### Official Reference Links

- [Developer Documentation Overview](#)
- [Plugin Development Guide](#)
- [Mautic vs Symfony](#)
- [Integrations](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 6: Setting up a debugging environment

#### Learning objectives

- Configure development debugging tools
- Implement logging and monitoring
- Set up performance profiling

#### Context

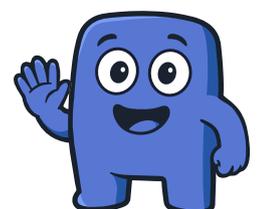
A proper debugging environment is essential for efficient development and troubleshooting. This module should be completed early in the learning path to ensure developers have the right tools before tackling coding tasks.

#### Key topics

- IDE debugging configuration (PHPStorm, VS Code)
- Xdebug setup and configuration
- Logging setup and management
- Performance profiling tools (Xhprof, Blackfire)
- Error tracking and monitoring
- Database query debugging
- Frontend debugging (browser dev tools)
- Remote debugging techniques

#### Official Reference Links

- [Development Environment](#)
- [Troubleshooting](#)
- [Log Management](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 7: Mautic security best practices

#### Learning objectives

- Implement security best practices in Mautic development
- Understand common security vulnerabilities and prevention
- Secure API endpoints and data handling

#### Context

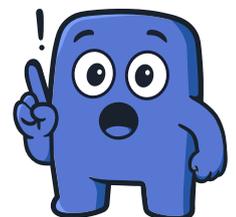
Security is paramount in any application handling customer data. This module covers essential security practices that developers must understand and implement when working with Mautic.

#### Key topics

- Data sanitization and validation
- Preventing XSS (Cross-Site Scripting) attacks
- CSRF (Cross-Site Request Forgery) protection
- SQL injection prevention
- Permissions and role management
- Secure API practices and authentication
- Protecting sensitive configuration data
- Password hashing and storage
- Secure file uploads and handling
- HTTPS and SSL/TLS configuration
- Security headers and configurations

#### Official Reference Links

- [Security Best Practices](#)
- [Configuration Security](#)
- [API Security](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 8: Mautic's API overview

#### Learning objectives

- Understand Mautic's API architecture
- Learn about authentication methods
- Explore available API endpoints

#### Context

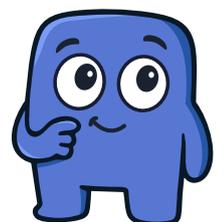
Mautic's API is a powerful tool for integrating with external systems and automating tasks. This module provides an overview of the API's capabilities and structure.

#### Key topics

- REST API fundamentals
- Authentication methods (OAuth, Basic Auth)
- API rate limiting and best practices
- Available endpoints overview
- Response formats and error handling
- API versioning and compatibility

#### Official Reference Links

- [API Configuration](#)
- [API Documentation](#)
- [REST API Reference](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 9: Working with Mautic's API

#### Learning objectives

- Implement API authentication
- Perform CRUD operations via API
- Handle API responses and errors

#### Context

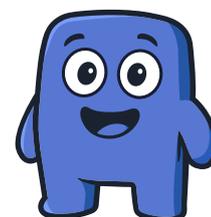
Practical implementation of Mautic's REST API is essential for integrations and custom applications. This module provides hands-on experience with API operations.

#### Key topics

- Setting up API credentials
- Making authenticated requests
- Contact management via API
- Campaign and email operations
- Asset and form management
- Batch operations and bulk updates
- Error handling and debugging

#### Official Reference Links

- [REST API Documentation](#)
- [Authentication Guide](#)
- [GitHub Examples](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 10: Building custom API endpoints

#### Learning objectives

- Create custom API endpoints
- Implement proper authentication and authorization
- Design RESTful API interfaces

#### Context

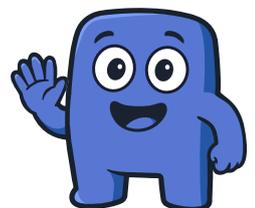
Sometimes the standard API endpoints aren't sufficient for specific use cases. This module teaches how to create custom API endpoints that integrate seamlessly with Mautic's existing API structure.

#### Key topics

- Controller development for APIs
- Route configuration
- Request validation and serialization
- Response formatting
- Authentication integration
- API documentation and testing
- Performance considerations

#### Official Reference Links

- [API Development](#)
- [Plugin Development](#)
- [GitHub Repository](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 11: Introduction to Mautic plugins

#### Learning objectives

- Explore existing plugins and their functionality
- Understand plugin installation and management
- Learn about plugin marketplace and distribution

#### Context

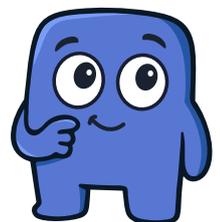
Before creating plugins, it's important to understand the existing plugin ecosystem and how plugins integrate with Mautic's core functionality. This foundational knowledge is essential before diving into plugin development.

#### Key topics

- Plugin marketplace overview
- Installation and activation
- Plugin configuration
- Common plugin patterns
- Plugin dependencies
- Version compatibility
- Community plugins vs. commercial plugins

#### Official Reference Links

- [Plugin Documentation](#)
- [Mautic Marketplace](#)
- [Plugin Resources](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 12: Plugin development

#### Learning objectives

- Understand Mautic's plugin architecture
- Create and structure plugins
- Implement plugin lifecycle management

#### Context

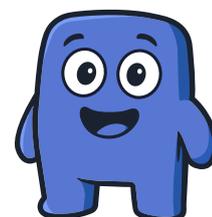
Plugins are the primary way to extend Mautic's functionality. This module covers the fundamentals of plugin development and architecture.

#### Key topics

- Plugin architecture and structure
- Plugin lifecycle and events
- Configuration and settings
- Database integration
- User interface integration
- Security considerations
- Distribution and packaging
- Authentication with third party systems

#### Official Reference Links

- [Plugin Development](#)
- [Plugin Marketplace](#)
- [Plugin Structure](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 13: Setting up a plugin development environment

#### Learning objectives

- Configure development environment for plugin development
- Set up debugging and testing tools
- Implement version control for plugins

#### Context

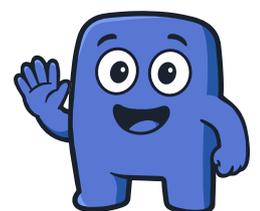
A proper development environment is crucial for efficient plugin development. This module covers the tools and configurations needed for professional plugin development.

#### Key topics

- Development environment setup
- IDE configuration and debugging
- Version control best practices
- Testing framework setup
- Continuous integration
- Documentation tools
- Package management

#### Official Reference Links

- [Development Environment](#)
- [Plugin Dependencies](#)
- [Continuous Integration](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 14: Creating your first Mautic plugin

#### Learning objectives

- Create a basic plugin from scratch
- Implement core plugin functionality
- Test and debug plugin functionality

#### Context

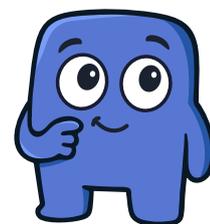
This hands-on module walks through creating a complete plugin, from initial structure to working functionality, providing practical experience with the development process.

#### Key topics

- Plugin scaffolding and structure
- Configuration files and metadata
- Basic functionality implementation
- Event handling and hooks
- User interface integration
- Testing and validation
- Installation and activation
- Console commands for the plugin
- Familiarity with DTO (Data Transfer Objects) pattern

#### Official Reference Links

- [Plugin Getting Started](#)
- [Plugin Structure](#)
- [Plugin Config](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 15: Handling plugin updates and maintenance

#### Learning objectives

- Implement plugin versioning and updates
- Handle database migrations
- Manage plugin lifecycle events

#### Context

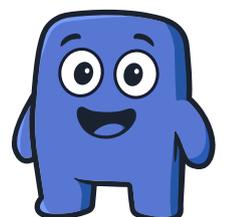
Long-term plugin maintenance requires proper update mechanisms and lifecycle management. This module covers strategies for maintaining plugins over time.

#### Key topics

- Version management and semantic versioning
- Database schema updates
- Configuration migrations
- Backward compatibility
- Update notifications and processes
- Error handling during updates
- Rollback strategies

#### Official Reference Links

- [Plugin Migration 4 to 5](#)
- [Marketplace Best Practices](#)
- [Plugin Installation](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 16: Mautic's Continuous Integration and code quality

#### Learning objectives

- Understand Mautic's CI/CD pipeline
- Implement code quality tools
- Ensure code standards compliance

#### Context

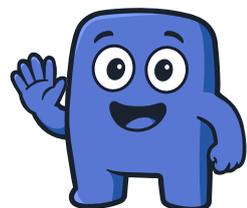
Maintaining code quality is essential for contributing to Mautic or developing professional plugins. This module covers the tools and practices used in Mautic's development workflow.

#### Key topics

- PHPStan for static analysis
- PHP CS Fixer for code style
- Rector for automated refactoring
- PHPUnit for testing
- GitHub Actions and CI pipelines
- Code coverage analysis
- Pre-commit hooks
- Automated code reviews
- Quality gates and standards

#### Official Reference Links

- [Continuous Integration](#)
- [Code Standards](#)
- [Testing Guide](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 17: Mautic contribution guide

#### Learning objectives

- Understand the contribution process
- Learn best practices for contributing
- Navigate the Mautic community

#### Context

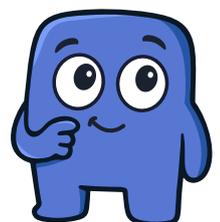
Contributing to Mautic benefits the entire community and helps developers grow their skills. This module covers how to effectively contribute to the Mautic project.

#### Key topics

- Contribution workflow and guidelines
- Fork, branch, and pull request process
- Issue tracking and bug reporting
- Feature proposal process
- Documentation contributions
- Code review process
- Community communication channels
- Recognition and credits

#### Official Reference Links

- [Contributing to Mautic](#)
- [Mautic Community](#)
- [GitHub Repository](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 17: Mautic contribution guide

#### Learning objectives

- Understand the contribution process
- Learn best practices for contributing
- Navigate the Mautic community

#### Context

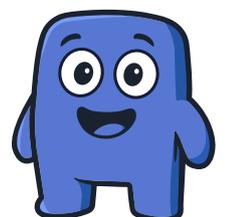
Contributing to Mautic benefits the entire community and helps developers grow their skills. This module covers how to effectively contribute to the Mautic project.

#### Key topics

- Contribution workflow and guidelines
- Fork, branch, and pull request process
- Issue tracking and bug reporting
- Feature proposal process
- Documentation contributions
- Code review process
- Community communication channels
- Recognition and credits

#### Official Reference Links

- [Contributing to Mautic](#)
- [Mautic Community](#)
- [GitHub Repository](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 18: Customising the user interface

#### Learning objectives

- Modify Mautic's user interface
- Implement custom UI components
- Integrate with Mautic's design system

#### Context

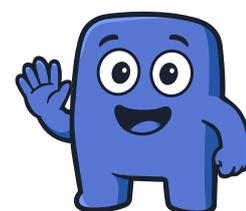
User interface customization allows developers to create tailored experiences within Mautic. This module covers techniques for modifying and extending the UI.

#### Key topics

- UI architecture and components
- Theme customization
- Custom form elements
- Dashboard widgets
- Menu and navigation customization
- Responsive design considerations
- Accessibility requirements

#### Official Reference Links

- [Themes Getting Started](#)
- [Theme Forms](#)
- [Design Permissions](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 19: Mautic's Twig templating engine

#### Learning objectives

- Master Twig templating in Mautic
- Create custom templates and layouts
- Implement template inheritance and blocks

#### Context

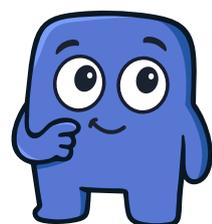
Twig is the templating engine used throughout Mautic for rendering views and emails. Understanding Twig is essential for customizing the user interface and email templates.

#### Key topics

- Twig syntax and fundamentals
- Template inheritance and blocks
- Custom Twig functions and filters
- Template debugging
- Performance optimization
- Security considerations
- Integration with Mautic's data
- Translations and pluralization

#### Official Reference Links

- [Themes Getting Started](#)
- [Twig Documentation](#)
- [Theme System](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 20: Customizing landing page templates

#### Learning objectives

- Develop responsive landing page templates
- Implement template variables and personalization
- Optimize landing pages for conversion

#### Context

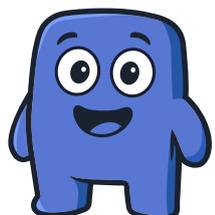
Landing pages are critical for lead generation and conversion. This module focuses specifically on creating effective landing page templates.

#### Key topics

- Landing page template structure
- Responsive design for landing pages
- Form integration in templates
- Template variables and tokens
- Personalization techniques
- A/B testing setup
- Performance optimization
- SEO considerations

#### Official Reference Links

- [Themes Getting Started](#)
- [Landing Pages](#)
- [Theme System](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 21: Email template development with MJML

#### Learning objectives

- Master MJML for email development
- Create responsive email templates
- Implement dynamic content and personalization

#### Context

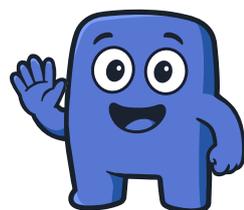
MJML is a powerful framework for creating responsive emails. This dedicated module covers email template development in depth, including MJML usage.

#### Key topics

- Introduction to MJML framework
- MJML syntax and components
- Converting MJML to HTML
- Responsive email design principles
- Email client compatibility
- Dynamic content integration
- Personalization tokens and variables
- Email deliverability optimization
- Testing across email clients
- Inline CSS and styling
- Dark mode support

#### Official Reference Links

- [Email Templates](#)
- [GrapesJS Builder](#)
- [MJML Documentation](#)
- [Themes Getting Started](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 22: Adding custom JavaScript and CSS

#### Learning objectives

- Integrate custom JavaScript and CSS
- Implement frontend build processes
- Optimize performance and loading

#### Context

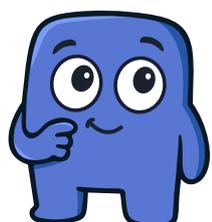
Custom JavaScript and CSS allow for enhanced user experiences and specialized functionality. This module covers best practices for frontend development in Mautic.

#### Key topics

- Asset management and organization
- JavaScript event handling
- CSS preprocessing and optimization
- Build tools and workflows
- Performance considerations
- Security and XSS prevention
- Third-party library integration

#### Official Reference Links

- [Themes Getting Started](#)
- [Theme System](#)
- [Frontend Assets](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 23: Database and entities

#### Learning objectives

- Understand Mautic's database architecture
- Work with entity relationships
- Implement database best practices

#### Context

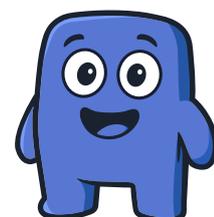
Database knowledge is fundamental for Mautic development. This module covers the database structure and how to work with it effectively.

#### Key topics

- Database schema overview
- Entity relationships and associations
- Query optimization
- Indexing strategies
- Data integrity and constraints
- Performance monitoring
- Backup and recovery considerations

#### Official Reference Links

- [Plugin Data](#)
- [Components Database](#)
- [Installation Database](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 24: Understanding Mautic's database schema

#### Learning objectives

- Navigate Mautic's database structure
- Understand key tables and relationships
- Implement proper database queries

#### Context

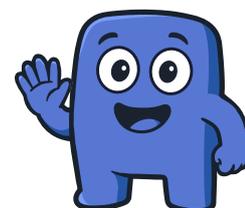
A deep understanding of Mautic's database schema is essential for effective development and troubleshooting. This module provides a comprehensive overview of the database structure.

#### Key topics

- Core entity tables
- Relationship mappings
- Audit and tracking tables
- Configuration and settings storage
- Queue and job tables
- Cache and session management
- Data archiving and cleanup

#### Official Reference Links

- [Contact Management](#)
- [Plugin Data](#)
- [Components Database](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 25: Working with Doctrine ORM in Mautic

#### Learning objectives

- Master Doctrine ORM concepts
- Implement entity mapping and relationships
- Optimize database queries

#### Context

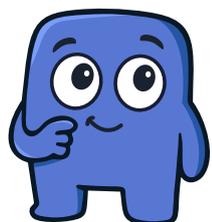
Doctrine ORM is the database abstraction layer used in Mautic. Understanding Doctrine is crucial for effective database operations and entity management.

#### Key topics

- Entity mapping and annotations
- Repository patterns
- Query builder and DQL
- Relationship management
- Lazy loading and performance
- Caching strategies
- Transaction management

#### Official Reference Links

- [Plugin Data](#)
- [Components Database](#)
- [Doctrine Documentation](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 26: Creating custom entities

#### Learning objectives

- Design and implement custom entities
- Set up entity relationships
- Implement entity validation and constraints
- Extending Mautic's core entities

#### Context

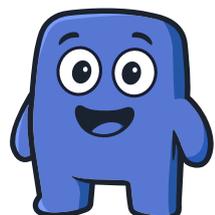
Custom entities allow developers to extend Mautic's data model for specific use cases. This module covers the complete process of creating and managing custom entities.

#### Key topics

- Entity design and planning
- Annotation and mapping configuration
- Repository implementation
- Form integration
- API integration
- Validation and constraints
- Performance considerations

#### Official Reference Links

- [Plugin Data](#)
- [Components Database](#)
- [Plugin Structure](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 27: Database migrations and schema updates

#### Learning objectives

- Create and manage database migrations
- Handle schema updates safely
- Implement rollback strategies

#### Context

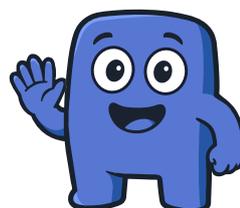
Database migrations are essential for maintaining database schema consistency across environments. This module covers migration strategies and best practices.

#### Key topics

- Migration file structure and naming
- Schema modification techniques
- Data migration strategies
- Version control integration
- Rollback and recovery
- Testing migrations
- Production deployment considerations

#### Official Reference Links

- [Plugin Data](#)
- [Plugin Installation](#)
- [Components Database](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 28: Command line interface and automation

#### Learning objectives

- Understand Mautic's CLI architecture
- Execute built-in commands
- Configure cron jobs and automation
- Create custom CLI commands

#### Context

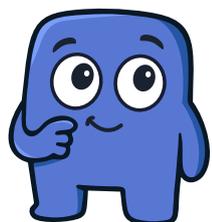
Mautic's CLI provides powerful tools for administration, maintenance, and automation. This consolidated module covers both using existing commands and creating custom ones.

#### Key topics

- CLI architecture and structure
- Built-in command overview, command structure and syntax
- Cron job configuration and scheduling
- Queue processing automation
- Email sending automation
- Cache management
- Database operations
- Creating custom CLI commands
- Command class structure, argument and option handling
- Input validation and output formatting
- Progress indicators
- Command registration and testing
- Debugging and logging

#### Official Reference Links

- [Cron Jobs](#)
- [Command Line](#)
- [Queue Processing](#)
- [Components Commands](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 29: Event listeners and subscribers

#### Learning objectives

- Understand Mautic's event system
- Distinguish between listeners and subscribers
- Implement event-driven architecture

#### Context

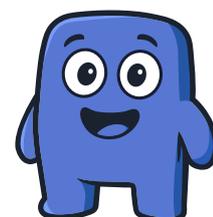
Mautic's event system provides a powerful way to extend functionality without modifying core code. This module covers the fundamentals of event-driven development.

#### Key topics

- Event system architecture
- Event listener vs. subscriber patterns
- Event priority and ordering
- Custom event creation
- Performance considerations
- Debugging events
- Best practices

#### Official Reference Links

- [Event Listeners](#)
- [Components Events](#)
- [Plugin Structure](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 30: Understanding Mautic events

#### Learning objectives

- Identify available events in Mautic
- Understand event data and context
- Implement event handling strategies

#### Context

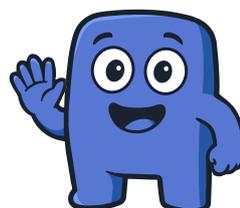
Mautic dispatches numerous events throughout its operation. Understanding these events is crucial for creating responsive and integrated extensions.

#### Key topics

- Core event types and categories
- Event lifecycle and timing
- Event data and context
- Event documentation
- Event debugging tools
- Performance impact
- Security considerations

#### Official Reference Links

- [Components Events](#)
- [Webhook Events](#)
- [Event Listeners](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 31: Creating event listeners

#### Learning objectives

- Implement event listeners
- Handle event data and responses
- Manage listener lifecycle

#### Context

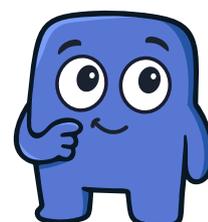
Event listeners provide a simple way to respond to specific events in Mautic. This module covers the creation and management of event listeners.

#### Key topics

- Listener class structure
- Event subscription
- Event handling methods
- Response modification
- Error handling
- Listener registration
- Testing listeners

#### Official Reference Links

- [Event Listeners](#)
- [Components Events](#)
- [Plugin Structure](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 32: Using event subscribers

#### Learning objectives

- Implement event subscribers
- Handle multiple events in a single class
- Optimize subscriber performance

#### Context

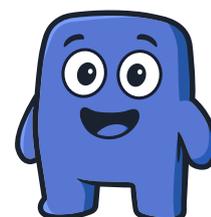
Event subscribers provide a more organized way to handle multiple related events. This module covers subscriber patterns and best practices.

#### Key topics

- Subscriber class structure
- Multiple event handling
- Event priority management
- Conditional event handling
- Subscriber registration
- Performance optimization
- Testing strategies

#### Official Reference Links

- [Event Listeners](#)
- [Components Events](#)
- [Plugin Getting Started](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 33: Email and campaign development

#### Learning objectives

- Develop custom email functionality
- Create advanced campaign logic
- Implement email tracking and analytics

#### Context

Email and campaigns are core features of Mautic. This module covers advanced development techniques for creating sophisticated email and campaign functionality.

#### Key topics

- Email service architecture
- Campaign workflow development
- Email template engines
- Personalization and segmentation
- Delivery optimization
- Analytics and tracking
- A/B testing implementation

#### Official Reference Links

- [Campaigns](#)
- [Email Channels](#)
- [Components Campaigns](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 34: Developing campaign logic

#### Learning objectives

- Create complex campaign workflows
- Implement conditional logic
- Optimize campaign performance

#### Context

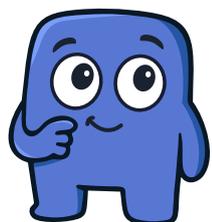
Advanced campaign development requires understanding of workflow logic and decision trees. This module covers creating sophisticated automated campaigns.

#### Key topics

- Campaign workflow architecture
- Decision node implementation
- Conditional logic and branching
- Timing and scheduling
- Performance optimization
- Testing and debugging
- Analytics integration

#### Official Reference Links

- [Creating Campaigns](#)
- [Campaign Events](#)
- [Components Campaigns](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 35: Advanced email and campaign customization

#### Learning objectives

- Implement advanced customization techniques
- Create custom campaign actions
- Develop sophisticated email features

#### Context

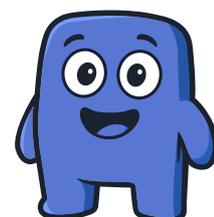
This module covers advanced techniques for customizing email and campaign functionality beyond standard configuration options.

#### Key topics

- Custom campaign actions
- Advanced email features
- Integration with external services
- Custom tracking and analytics
- Advanced personalization
- Performance optimization
- Security considerations

#### Official Reference Links

- [Components Campaigns](#)
- [Event Listeners](#)
- [Email Configuration](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 36: Using and extending reports

#### Learning objectives

- Understand Mautic's reporting system
- Create custom reports
- Extend existing report functionality

#### Context

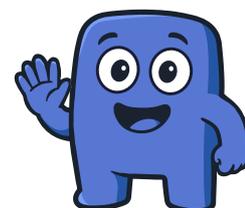
Reports provide valuable insights into marketing performance. This module covers how to work with Mautic's reporting system and create custom reports for specific needs.

#### Key topics

- Report architecture and structure
- Built-in reports overview
- Creating custom report types
- Query building for reports
- Data aggregation and calculation
- Custom report widgets
- Exporting and scheduling reports
- Performance optimization for large datasets
- Report permissions and access control
- Integrating custom data sources

#### Official Reference Links

- [Reports](#)
- [Components Reports](#)
- [Plugin Development](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 37: Testing and debugging

#### Learning objectives

- Set up comprehensive testing environments
- Implement debugging strategies
- Create automated test suites, and distinguish between unit and functional tests

#### Context

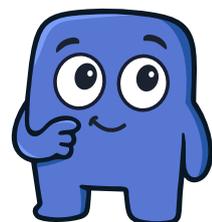
Testing and debugging are crucial for maintaining high-quality Mautic installations and custom development. This module covers tools and techniques for effective testing, including unit and functional testing approaches.

#### Key topics

- Testing environment setup, debugging tools and techniques
- Unit vs Functional testing concepts
- Unit testing with PHPUnit
  - Testing individual classes and methods
  - Mocking dependencies
  - Isolated test cases
  - Fast execution
- Functional testing with Symfony
  - Testing complete workflows
  - Database interactions
  - API endpoint testing and browser automation testing
- Integration testing, test data fixtures and factories
- Performance testing, security testing and Continuous integration
- Code coverage analysis

#### Official Reference Links

- [Development Environment](#)
- [Continuous Integration](#)
- [Troubleshooting](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Module 38: Testing plugins

#### Learning objectives

- Implement comprehensive plugin testing strategies
- Create unit and functional tests for plugins
- Set up automated testing pipelines

#### Context

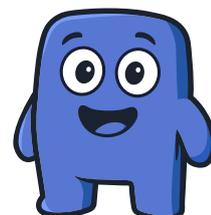
Plugin testing requires specialized approaches to ensure functionality works correctly within the Mautic ecosystem. This module focuses specifically on testing plugin code.

#### Key topics

- Plugin testing environment setup
- Unit testing plugin components
  - Testing plugin services and event listeners/subscribers
  - Testing custom entities
  - Mocking Mautic dependencies
- Functional testing plugin features
  - Testing complete plugin workflows
  - Testing UI components and database operations
- Integration testing with Mautic core
- Test fixtures for plugin data
- Automated testing pipelines
- Test coverage for plugins
- Testing plugin updates and migrations
- Performance testing for plugins

#### Official Reference Links

- [Continuous Integration](#)
- [Plugin Development](#)
- [Testing Best Practices](#)



# MAUTIC DEVELOPER CERTIFICATION

## Curriculum overview

### Certification requirements

A decorative graphic consisting of a yellow arrow pointing right, which is partially overlapped by a teal arrow pointing right.

The Mautic Developer Certification exam is a comprehensive assessment of your Mautic development skills. It covers all the key topics outlined in the curriculum.

The exam consists of multiple-choice questions and scenario-based problems.

To earn the certification, candidates must:

1. Pay for the examination entry fee
2. Verify identity in the online examination platform
3. Sit and pass the certification exam
4. Achieve a score of 80% or higher

To prepare for the exam, we recommend the following:

- Review the Mautic documentation thoroughly.
- Complete the hands-on exercises and sample projects.
- Participate in Mautic community forums and discussions.
- Gain practical experience by working on Mautic development projects.
- Contribute to Mautic by fixing bugs, adding or improving features, and reviewing contributions.

Further information including details on how to book your examination and fees is available online via the Mautic website.

This document provides an overview of the Mautic Developer Certification curriculum, highlighting the key topics and providing guidance on how to prepare for the certification exam. This certification is a valuable credential for developers looking to demonstrate their expertise in Mautic and contribute to the Mautic community. Certifications do not expire, however a new certification is offered when each major version of Mautic enters Long Term Support. Existing certified developers will have an opportunity to sit the exam to gain certification in the new version.