# **Certified Mautic Integrator Curriculum**

## **Course Overview**

This comprehensive curriculum is designed to train technical professionals to become certified Mautic Integrators, covering system administration, deployment, integration, and advanced technical configuration of Mautic platforms.

## **Prerequisites**

* System administration experience (Linux/Unix preferred)
* Basic web server administration knowledge
* Understanding of databases (MySQL/MariaDB)
* Command line interface familiarity
* Basic networking and security concepts
* PHP development knowledge (helpful)
* Git version control basics

## **Module 1: Mautic Configuration (UI and local.php)**

### **Learning Objectives**

* Master both UI-based and file-based configuration methods
* Understand configuration hierarchy and precedence
* Implement advanced configuration strategies

### **Context**

Mautic configuration can be managed through the web interface and directly via configuration files. Understanding both methods and their appropriate use cases is essential for system integrators who need to manage multiple environments and complex deployments.

### **Key Topics**

* Configuration file structure and syntax
* UI vs. file-based configuration management
* Configuration inheritance and environment-specific settings
* Security considerations for configuration files
* Configuration validation and troubleshooting
* Automated configuration deployment strategies

### **Official Reference Links**

* [Configuration Settings](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [System Configuration](https://docs.mautic.org/en/5.2/configuration/settings.html)
* [File Permissions and Security](https://docs.mautic.org/en/5.x/troubleshooting/file_ownership_permissions.html)

## **Module 2: Mautic Configuration Best Practices**

### **Learning Objectives**

* Implement production-ready configuration strategies
* Optimize performance through proper configuration
* Establish security best practices for configuration management

### **Context**

Production Mautic deployments require careful configuration planning to ensure security, performance, and maintainability. This module covers enterprise-level configuration strategies and best practices.

### **Key Topics**

* Environment-specific configuration strategies
* Performance optimization through configuration
* Security hardening and configuration
* Configuration backup and versioning
* Monitoring and alerting for configuration changes
* Configuration documentation and change management

### **Official Reference Links**

* [Production Configuration](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [Security Configuration](https://docs.mautic.org/en/5.x/troubleshooting/file_ownership_permissions.html)
* [Performance Optimization](https://docs.mautic.org/en/5.x/configuration/cron_jobs.html)

## **Module 3: Integrating Mautic with External Systems**

### **Learning Objectives**

* Design and implement complex system integrations
* Understand integration patterns and best practices
* Troubleshoot integration issues and performance problems

### **Context**

Mautic rarely operates in isolation and must integrate with CRM systems, marketing tools, e-commerce platforms, and custom applications. This module covers comprehensive integration strategies and implementation.

### **Key Topics**

* Integration architecture and design patterns
* API-based integration strategies
* Webhook implementation and management
* Data synchronization and mapping
* Error handling and retry mechanisms
* Integration monitoring and maintenance

### **Official Reference Links**

* [API Configuration](https://docs.mautic.org/en/5.2/configuration/settings.html)
* [Webhook Configuration](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [Plugin Resources](https://docs.mautic.org/en/5.2/plugins/plugin_resources.html)

## **Module 4: Customizing Mautic Settings**

### **Learning Objectives**

* Implement advanced customization strategies
* Create custom configuration options
* Manage complex multi-tenant configurations

### **Context**

Advanced Mautic deployments often require customizations that go beyond standard configuration options. This module teaches how to safely customize Mautic while maintaining upgradeability.

### **Key Topics**

* Custom configuration parameters
* Configuration inheritance and overrides
* Multi-tenant configuration strategies
* Custom environment variables
* Configuration templating and automation
* Migration strategies for custom configurations

### **Official Reference Links**

* [Advanced Configuration](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [Environment Configuration](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [System Administration](https://docs.mautic.org/en/5.2/)

## **Module 5: Github, First Glance**

### **Learning Objectives**

* Understand Mautic's development workflow on GitHub
* Navigate the Mautic codebase and repository structure
* Understand contribution processes and community involvement

### **Context**

GitHub is central to Mautic development and community contribution. Understanding the repository structure, development process, and community workflows is essential for integrators who need to track changes, contribute fixes, or manage custom deployments.

### **Key Topics**

* Mautic repository structure and organization
* Branch strategy and release workflow
* Issue tracking and feature requests
* Community contribution guidelines
* Code review and pull request process
* Release notes and changelog interpretation

### **Official Reference Links**

* [GitHub Repository](https://github.com/mautic/mautic-documentation)
* [Community Contribution](https://docs.mautic.org/en/5.2/)
* [Developer Documentation](https://docs.mautic.org/en/5.2/)

## **Module 6: Installing Mautic (without Composer)**

### **Learning Objectives**

* Perform traditional ZIP-based Mautic installations
* Understand legacy installation methods and use cases
* Troubleshoot traditional installation issues

### **Context**

While Composer-based installations are now preferred, understanding traditional installation methods is important for legacy systems, specific hosting environments, and troubleshooting scenarios.

### **Key Topics**

* ZIP package installation process
* Web-based installer configuration
* Command-line installation options
* File permission and ownership setup
* Database configuration and initialization
* Post-installation security hardening

### **Official Reference Links**

* [Traditional Installation](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [Installation Troubleshooting](https://docs.mautic.org/en/5.2/troubleshooting/troubleshooting.html)
* [File Permissions](https://docs.mautic.org/en/5.x/troubleshooting/file_ownership_permissions.html)

## **Module 7: Composer & Installing Mautic with it (incl. Packagist)**

### **Learning Objectives**

* Master Composer-based Mautic installations and management
* Understand dependency management and package resolution
* Implement advanced Composer workflows for Mautic

### **Context**

Composer has become the standard method for installing and managing Mautic. This module covers comprehensive Composer usage, from basic installation to advanced dependency management and custom package development.

### **Key Topics**

* Composer fundamentals and dependency management
* Mautic Recommended Project structure
* Custom package development for Mautic
* Packagist integration and private repositories
* Composer scripting and automation
* Performance optimization and troubleshooting

### **Official Reference Links**

* [Composer Installation](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [Composer Management](https://docs.mautic.org/en/latest/getting_started/switching_composer.html)
* [Marketplace Integration](https://docs.mautic.org/en/4.x/marketplace/marketplace.html)

## **Module 8: Updating Mautic (incl. Mautic release cycle, and update strategies)**

### **Learning Objectives**

* Implement comprehensive update strategies for various environments
* Understand Mautic's release cycle and version management
* Develop automated update workflows and rollback procedures

### **Context**

Keeping Mautic updated is crucial for security and functionality, but updates can be complex in production environments. This module covers sophisticated update strategies and automation techniques.

### **Key Topics**

* Mautic release cycle and versioning strategy
* Update planning and testing procedures
* Automated update workflows
* Rollback strategies and disaster recovery
* Multi-environment update coordination
* Update monitoring and validation

### **Official Reference Links**

* [Update Process](https://docs.mautic.org/en/latest/getting_started/how_to_update_mautic.html)
* [Composer Updates](https://docs.mautic.org/en/latest/getting_started/how_to_update_mautic.html)
* [Update Troubleshooting](https://docs.mautic.org/en/5.2/troubleshooting/troubleshooting.html)

## **Module 9: Backup / Restore**

### **Learning Objectives**

* Design comprehensive backup strategies for Mautic systems
* Implement automated backup and restore procedures
* Develop disaster recovery plans and testing protocols

### **Context**

Reliable backup and restore capabilities are essential for production Mautic deployments. This module covers enterprise-grade backup strategies, automation, and disaster recovery planning.

### **Key Topics**

* Backup strategy design and implementation
* Database backup and restoration techniques
* File system backup and synchronization
* Automated backup scheduling and monitoring
* Disaster recovery planning and testing
* Cross-environment data migration

### **Official Reference Links**

* [Backup Procedures](https://docs.mautic.org/en/latest/getting_started/how_to_update_mautic.html)
* [Database Management](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [System Administration](https://docs.mautic.org/en/5.2/)

## **Module 10: Understanding Git & Github, incl. PR, Gitpod, Reviewing, Merging**

### **Learning Objectives**

* Master Git workflows for Mautic development and deployment
* Understand GitHub collaboration features and processes
* Implement advanced Git strategies for team environments

### **Context**

Git and GitHub are fundamental tools for Mautic development, contribution, and deployment management. This module provides comprehensive coverage of Git workflows and GitHub collaboration features.

### **Key Topics**

* Advanced Git workflows and branching strategies
* Pull request creation, review, and management
* Gitpod integration and cloud development
* Code review best practices and tools
* Merge strategies and conflict resolution
* Git automation and hooks

### **Official Reference Links**

* [GitHub Integration](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [Development Workflow](https://docs.mautic.org/en/5.2/)
* [Community Contribution](https://docs.mautic.org/en/5.2/)

## **Module 11: Cron / Console Commands (Integrator level)**

### **Learning Objectives**

* Master advanced cron job configuration and management
* Implement sophisticated command-line automation
* Optimize system performance through proper task scheduling

### **Context**

Cron jobs are essential for Mautic's operation, handling everything from email sending to data processing. This module covers advanced cron management, monitoring, and optimization techniques.

### **Key Topics**

* Advanced cron job configuration and scheduling
* Command-line tool mastery and scripting
* Performance monitoring and optimization
* Error handling and alerting
* Load balancing and distributed processing
* Custom command development

### **Official Reference Links**

* [Cron Jobs Configuration](https://docs.mautic.org/en/5.x/configuration/cron_jobs.html)
* [Command Line Interface](https://docs.mautic.org/en/5.x/configuration/command_line_interface.html)
* [System Optimization](https://docs.mautic.org/en/5.x/configuration/cron_jobs.html)

## **Module 12: Deployment**

### **Learning Objectives**

* Design and implement robust deployment pipelines
* Automate deployment processes for multiple environments
* Implement zero-downtime deployment strategies

### **Context**

Professional Mautic deployments require sophisticated deployment strategies that ensure reliability, minimize downtime, and enable rapid rollbacks when necessary.

### **Key Topics**

* Deployment pipeline design and implementation
* CI/CD integration and automation
* Zero-downtime deployment techniques
* Environment promotion strategies
* Configuration management in deployments
* Deployment monitoring and validation

### **Official Reference Links**

* [Production Deployment](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [Configuration Management](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [System Administration](https://docs.mautic.org/en/5.2/)

## **Module 13: Preparing Mautic for Production**

### **Learning Objectives**

* Implement production-ready configurations and optimizations
* Establish monitoring, logging, and alerting systems
* Optimize performance for high-volume operations

### **Context**

Production Mautic deployments require careful preparation to handle real-world traffic, data volumes, and operational requirements. This module covers all aspects of production readiness.

### **Key Topics**

* Production environment configuration
* Performance optimization and tuning
* Security hardening and compliance
* Monitoring and alerting setup
* Capacity planning and scaling
* Operational procedures and documentation

### **Official Reference Links**

* [Production Configuration](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [Performance Optimization](https://docs.mautic.org/en/5.x/configuration/cron_jobs.html)
* [Security Configuration](https://docs.mautic.org/en/5.x/troubleshooting/file_ownership_permissions.html)

## **Module 14: Deploying Mautic Applications**

### **Learning Objectives**

* Execute production deployments using various strategies
* Manage application lifecycle in production environments
* Implement deployment automation and monitoring

### **Context**

Deploying Mautic applications requires understanding of web application deployment principles, infrastructure management, and operational procedures for maintaining live systems.

### **Key Topics**

* Production deployment execution
* Infrastructure as Code (IaC) implementation
* Container deployment strategies
* Load balancer and reverse proxy configuration
* SSL/TLS certificate management
* Application lifecycle management

### **Official Reference Links**

* [Deployment Guide](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [Configuration Management](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [Troubleshooting Deployment](https://docs.mautic.org/en/5.2/troubleshooting/troubleshooting.html)

## **Module 15: Understanding DDEV**

### **Learning Objectives**

* Master DDEV for Mautic development and testing
* Configure advanced DDEV environments
* Integrate DDEV into development workflows

### **Context**

DDEV provides standardized, containerized development environments for Mautic. Understanding DDEV is essential for consistent development environments and testing procedures.

### **Key Topics**

* DDEV installation and configuration
* Container orchestration with DDEV
* Custom DDEV configurations for Mautic
* Development workflow integration
* Multi-environment testing with DDEV
* DDEV troubleshooting and optimization

### **Official Reference Links**

* [DDEV Setup](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [Development Environment](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [DDEV Troubleshooting](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)

## **Module 16: Integration Basics (Plugins, Webhooks, APIs, incl. n8n etc., …, also: Postman)**

### **Learning Objectives**

* Master various integration technologies and patterns
* Implement and test integrations using modern tools
* Design scalable integration architectures

### **Context**

Modern Mautic deployments require sophisticated integration capabilities with various external systems. This module covers the full spectrum of integration technologies and tools.

### **Key Topics**

* Plugin architecture and development
* Webhook implementation and management
* REST API integration patterns
* Integration platform usage (n8n, Zapier)
* API testing with Postman and similar tools
* Integration monitoring and troubleshooting

### **Official Reference Links**

* [API Documentation](https://docs.mautic.org/en/5.2/configuration/settings.html)
* [Plugin Development](https://docs.mautic.org/en/5.2/plugins/plugin_resources.html)
* [Webhook Configuration](https://docs.mautic.org/en/5.x/configuration/settings.html)

## **Module 17: Know Important Plugins**

### **Learning Objectives**

* Understand the ecosystem of important Mautic plugins
* Evaluate and select appropriate plugins for various use cases
* Manage plugin lifecycles in production environments

### **Context**

The Mautic plugin ecosystem provides extensive functionality for specific use cases. Understanding key plugins and their capabilities is essential for creating comprehensive solutions.

### **Key Topics**

* Plugin ecosystem overview and evaluation
* Essential plugins for common use cases
* Plugin installation and management
* Plugin security and compliance considerations
* Custom plugin development basics
* Plugin performance impact and optimization

### **Official Reference Links**

* [Plugin Marketplace](https://docs.mautic.org/en/4.x/marketplace/marketplace.html)
* [Plugin Management](https://docs.mautic.org/en/5.2/plugins/plugin_resources.html)
* [Plugin Development](https://docs.mautic.org/en/5.2/plugins/plugin_resources.html)

## **Module 18: First Peek into Mautic Database (& data model)**

### **Learning Objectives**

* Understand Mautic's database architecture and design
* Navigate the database schema and relationships
* Implement database maintenance and optimization strategies

### **Context**

Understanding Mautic's database structure is crucial for advanced integrations, reporting, performance optimization, and troubleshooting complex issues.

### **Key Topics**

* Database schema overview and design principles
* Entity relationships and data flow
* Database performance optimization
* Backup and recovery procedures
* Data migration and synchronization
* Database security and access control

### **Official Reference Links**

* [Database Configuration](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [System Administration](https://docs.mautic.org/en/5.x/segments/manage_segments.html)
* [Data Management](https://docs.mautic.org/en/5.2/)

## **Module 19: CSS Basics (for Forms & Co)**

### **Learning Objectives**

* Implement CSS customizations for Mautic components
* Create responsive and accessible designs
* Integrate custom styling with Mautic themes

### **Context**

While Mautic provides default styling, many implementations require custom CSS for forms, landing pages, and other components to match brand requirements and improve user experience.

### **Key Topics**

* CSS fundamentals for Mautic components
* Responsive design principles and implementation
* Form styling and customization techniques
* Theme integration and customization
* CSS preprocessing and build workflows
* Accessibility considerations in styling

### **Official Reference Links**

* [Theme Customization](https://docs.mautic.org/en/5.2/themes/customizing_themes.html)
* [Form Styling](https://docs.mautic.org/en/5.2/components/forms.html)
* [Email Template Design](https://docs.mautic.org/en/5.x/builders/email_landing_page.html)

## **Module 20: Javascript Basics (for Forms & Co)**

### **Learning Objectives**

* Implement JavaScript enhancements for Mautic components
* Create interactive form behaviors and validations
* Integrate with external JavaScript libraries and frameworks

### **Context**

JavaScript is essential for creating rich, interactive experiences with Mautic forms, landing pages, and integrations. This module covers practical JavaScript implementation for Mautic components.

### **Key Topics**

* JavaScript fundamentals for Mautic integration
* Form enhancement and validation
* Event handling and user interaction
* API integration with JavaScript
* Performance optimization and best practices
* Security considerations for client-side code

### **Official Reference Links**

* [Form Development](https://docs.mautic.org/en/5.2/components/forms.html)
* [Landing Page Builder](https://docs.mautic.org/en/5.x/builders/email_landing_page.html)
* [Integration Development](https://docs.mautic.org/en/5.2/)

## **Module 21: PHP Outlines (simple elements, role of Symfony as a framework)**

### **Learning Objectives**

* Understand PHP's role in Mautic architecture
* Grasp Symfony framework fundamentals and their application in Mautic
* Implement basic PHP customizations and extensions

### **Context**

Mautic is built on PHP and the Symfony framework. Understanding these foundational technologies is essential for advanced customizations, plugin development, and system administration.

### **Key Topics**

* PHP fundamentals and best practices
* Symfony framework architecture and components
* Mautic's use of Symfony patterns and services
* Object-oriented programming in the context of Mautic
* Dependency injection and service container
* MVC pattern implementation in Mautic

### **Official Reference Links**

* [Developer Documentation](https://docs.mautic.org/en/5.2/)
* [System Architecture](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [Customization Guidelines](https://docs.mautic.org/en/5.2/)

## **Module 22: Webserver Basics**

### **Learning Objectives**

* Configure and optimize web servers for Mautic deployments
* Implement security best practices for web server configuration
* Troubleshoot web server issues and performance problems

### **Context**

Proper web server configuration is critical for Mautic performance, security, and reliability. This module covers Apache, Nginx, and other web server technologies commonly used with Mautic.

### **Key Topics**

* Apache and Nginx configuration for Mautic
* SSL/TLS configuration and certificate management
* Performance optimization and caching strategies
* Security hardening and access control
* Virtual host configuration and management
* Load balancing and reverse proxy setup

### **Official Reference Links**

* [Installation Requirements](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [System Configuration](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [Security Configuration](https://docs.mautic.org/en/5.x/troubleshooting/file_ownership_permissions.html)

## **Module 23: Database Server Basics incl. some tuning, optimize tables, ….**

### **Learning Objectives**

* Configure and optimize database servers for Mautic
* Implement database performance tuning and maintenance
* Design backup and recovery strategies for database systems

### **Context**

Database performance directly impacts Mautic's overall performance and user experience. This module covers database administration, optimization, and maintenance specific to Mautic deployments.

### **Key Topics**

* MySQL/MariaDB configuration and optimization
* Database performance tuning and monitoring
* Index optimization and query performance
* Database maintenance and cleanup procedures
* Backup and recovery strategies
* Database security and access control

### **Official Reference Links**

* [Database Configuration](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)
* [Performance Optimization](https://docs.mautic.org/en/5.x/configuration/cron_jobs.html)
* [System Requirements](https://docs.mautic.org/en/5.x/getting_started/how_to_install_mautic.html)

## **Module 24: Shell (e.g. chown, scp, … AND the risks when working here) and Shell scripting**

### **Learning Objectives**

* Master essential shell commands for Mautic administration
* Develop shell scripts for automation and maintenance
* Understand security risks and best practices for shell operations

### **Context**

Shell access is essential for Mautic system administration, but it also presents security risks. This module covers safe and effective shell usage for Mautic management.

### **Key Topics**

* Essential shell commands for file and permission management
* Shell scripting for automation and maintenance
* Security considerations and risk mitigation
* Remote access and file transfer techniques
* Process management and monitoring
* Scripting best practices and error handling

### **Official Reference Links**

* [File Permissions](https://docs.mautic.org/en/5.x/troubleshooting/file_ownership_permissions.html)
* [Command Line Interface](https://docs.mautic.org/en/5.x/configuration/command_line_interface.html)
* [System Administration](https://docs.mautic.org/en/5.2/)

## **Module 25: Email Templating Challenges**

### **Learning Objectives**

* Overcome common email templating challenges
* Implement cross-client compatible email designs
* Optimize email templates for deliverability and engagement

### **Context**

Email templating presents unique challenges due to inconsistent client support and rendering differences. This module addresses these challenges with practical solutions.

### **Key Topics**

* Email client compatibility and rendering differences
* HTML email best practices and limitations
* CSS support and workarounds for email
* Image optimization and inline techniques
* Template testing and validation procedures
* Deliverability considerations in template design

### **Official Reference Links**

* [Email Builder](https://docs.mautic.org/en/5.x/builders/email_landing_page.html)
* [Email Configuration](https://docs.mautic.org/en/5.x/channels/emails.html)
* [Template Development](https://docs.mautic.org/en/5.x/builders/email_landing_page.html)

## **Module 26: MJML Basics**

### **Learning Objectives**

* Master MJML for responsive email template development
* Integrate MJML workflows with Mautic development
* Optimize MJML templates for performance and compatibility

### **Context**

MJML provides a powerful framework for creating responsive email templates that work across all email clients. Understanding MJML is essential for professional email template development.

### **Key Topics**

* MJML syntax and component library
* Responsive email design with MJML
* MJML compilation and build workflows
* Custom component development
* Integration with Mautic templating system
* MJML best practices and optimization

### **Official Reference Links**

* [MJML Email Templates](https://docs.mautic.org/en/5.x/builders/email_landing_page.html)
* [Email Template Development](https://docs.mautic.org/en/5.x/builders/email_landing_page.html)
* [Theme Integration](https://docs.mautic.org/en/5.2/themes/customizing_themes.html)

## **Module 27: Working with Themes**

### **Learning Objectives**

* Develop and customize Mautic themes
* Implement theme management and deployment strategies
* Create responsive and accessible theme designs

### **Context**

Themes control the visual presentation of Mautic's user-facing components. Understanding theme development and management is essential for creating branded, professional-looking Mautic implementations.

### **Key Topics**

* Theme architecture and development workflow
* Custom theme creation and modification
* Theme asset management and optimization
* Responsive design implementation in themes
* Theme deployment and version management
* Theme compatibility and upgrade strategies

### **Official Reference Links**

* [Theme Management](https://docs.mautic.org/en/5.x/themes/manage_themes.html)
* [Theme Customization](https://docs.mautic.org/en/5.2/themes/customizing_themes.html)
* [Theme Development](https://docs.mautic.org/en/5.2/themes/customizing_themes.html)

## **Module 28: Email Service Providers**

### **Learning Objectives**

* Configure and optimize various email service providers
* Implement advanced email delivery strategies
* Monitor and troubleshoot email delivery issues

### **Context**

Email delivery is critical to Mautic's effectiveness. Understanding various email service providers and their configuration options enables optimal email delivery performance.

### **Key Topics**

* Email service provider comparison and selection
* SMTP and API-based email delivery configuration
* Email authentication (SPF, DKIM, DMARC) setup
* Delivery optimization and throttling strategies
* Bounce handling and reputation management
* Multi-provider strategies and failover

### **Official Reference Links**

* [Email Configuration](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [Email Delivery](https://docs.mautic.org/en/5.x/channels/emails.html)
* [Transport Configuration](https://docs.mautic.org/en/5.x/configuration/settings.html)

## **Module 29: Email Deliverability (Avoid & Test being considered Spam, understand requirements, Google Postmaster Tools, ….)**

### **Learning Objectives**

* Implement comprehensive email deliverability strategies
* Monitor and improve sender reputation
* Utilize deliverability tools and analytics for optimization

### **Context**

Email deliverability is crucial for successful marketing automation. This module covers advanced deliverability techniques, monitoring, and optimization strategies.

### **Key Topics**

* Email deliverability fundamentals and best practices
* Sender reputation management and monitoring
* SPF, DKIM, and DMARC implementation and optimization
* Google Postmaster Tools and other deliverability platforms
* List hygiene and engagement optimization
* Deliverability testing and validation procedures

### **Official Reference Links**

* [Email Deliverability](https://docs.mautic.org/en/5.x/channels/emails.html)
* [Email Configuration](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [System Optimization](https://docs.mautic.org/en/5.x/configuration/cron_jobs.html)

## **Module 30: Secure Setup and Maintenance**

### **Learning Objectives**

* Implement comprehensive security measures for Mautic installations
* Establish security monitoring and incident response procedures
* Maintain security through updates and ongoing maintenance

### **Context**

Security is paramount in marketing automation systems that handle sensitive customer data. This module covers enterprise-level security implementation and maintenance.

### **Key Topics**

* Security architecture and threat modeling
* Access control and authentication systems
* Data encryption and protection strategies
* Security monitoring and logging
* Incident response and recovery procedures
* Security compliance and audit requirements

### **Official Reference Links**

* [Security Configuration](https://docs.mautic.org/en/5.x/troubleshooting/file_ownership_permissions.html)
* [File Permissions](https://docs.mautic.org/en/5.x/troubleshooting/file_ownership_permissions.html)
* [System Security](https://docs.mautic.org/en/5.x/configuration/settings.html)

## **Module 31: Mautic Translation (Transifex)**

### **Learning Objectives**

* Understand Mautic's translation and localization system
* Contribute to translation efforts through Transifex
* Implement multi-language Mautic deployments

### **Context**

Mautic supports multiple languages through community translation efforts coordinated via Transifex. Understanding the translation system is important for global deployments and community contribution.

### **Key Topics**

* Mautic localization architecture and implementation
* Transifex platform usage and workflow
* Translation contribution and review processes
* Multi-language deployment strategies
* Custom translation and localization development
* Language pack management and updates

### **Official Reference Links**

* [Localization Support](https://docs.mautic.org/en/5.x/configuration/settings.html)
* [Community Contribution](https://docs.mautic.org/en/5.2/)
* [International Configuration](https://docs.mautic.org/en/5.x/configuration/settings.html)

## **Module 32: Mautic REST API – Basic Understanding**

### **Learning Objectives**

* Master Mautic's REST API fundamentals
* Implement API authentication and security
* Develop basic API integrations and automations

### **Context**

Mautic's REST API enables powerful integrations and automations. Understanding API fundamentals is essential for integrators who need to connect Mautic with external systems.

### **Key Topics**

* REST API principles and Mautic implementation
* Authentication methods and security considerations
* API endpoints and resource management
* Request/response handling and error management
* Rate limiting and performance optimization
* API testing and debugging techniques

### **Official Reference Links**

* [API Configuration](https://docs.mautic.org/en/5.2/configuration/settings.html)
* [API Documentation](https://docs.mautic.org/en/5.2/)
* [Developer Resources](https://docs.mautic.org/en/5.2/)

## **Module 33: Mautic Community and Contribution (incl. Understanding Open Source)**

### **Learning Objectives**

* Understand open source principles and the Mautic community
* Contribute effectively to the Mautic project
* Engage with the community for support and collaboration

### **Context**

Mautic is an open source project with a vibrant community. Understanding how to engage with the community, contribute to the project, and leverage community resources is essential for long-term success with Mautic.

### **Key Topics**

* Open source principles and licensing
* Mautic community structure and governance
* Contribution guidelines and processes
* Code contribution and pull request workflow
* Community support channels and resources
* Event participation and networking

### **Official Reference Links**

* [Community Guidelines](https://docs.mautic.org/en/5.2/)
* [Contribution Process](https://docs.mautic.org/en/5.2/)
* [Developer Community](https://docs.mautic.org/en/5.2/)

# Q & A Samples

## **Category 1: System Administration and Configuration (25% - 20 Questions)**

### **Question 1**

Which file contains the main configuration parameters for a Mautic installation?

A) config/config.yml B) app/config/local.php C) .env D) config/parameters.ini

**Correct Answer: B) app/config/local.php**

### **Question 2**

What is the correct file permission setting for Mautic configuration files in a production environment?

A) 777 (read, write, execute for all) B) 644 (read/write for owner, read-only for group and others) C) 755 (read/write/execute for owner, read/execute for group and others) D) 600 (read/write for owner only)

**Correct Answer: B) 644 (read/write for owner, read-only for group and others)**

### **Question 3**

Which cron job is essential for processing queued emails in Mautic?

A) mautic:segments:update B) mautic:campaigns:trigger C) messenger:consume email D) mautic:social:monitoring

**Correct Answer: C) messenger:consume email**

### **Question 4**

What is the recommended approach for managing environment-specific configurations?

A) Hardcode all settings in local.php B) Use environment variables and configuration templates C) Store everything in the database D) Use separate Mautic installations for each environment

**Correct Answer: B) Use environment variables and configuration templates**

### **Question 5**

Which command checks for available Mautic updates via CLI?

A) bin/console mautic:update:check B) bin/console mautic:version:check C) bin/console mautic:update:find D) bin/console mautic:check:updates

**Correct Answer: A) bin/console mautic:update:check**

### **Question 6**

What is the purpose of the site\_url configuration parameter?

A) To set the admin panel URL B) To define the base URL for email links and tracking C) To configure the database connection D) To set the API endpoint URL

**Correct Answer: B) To define the base URL for email links and tracking**

### **Question 7**

Which database engines are officially supported by Mautic?

A) MySQL and PostgreSQL only B) MySQL, MariaDB, and PostgreSQL C) MySQL and MariaDB only D) Any database supported by Doctrine ORM

**Correct Answer: C) MySQL and MariaDB only**

### **Question 8**

What is the recommended way to configure email transport settings for production?

A) Use SMTP with authentication B) Use the local sendmail binary C) Configure via API-based email service providers D) Both A and C depending on requirements

**Correct Answer: D) Both A and C depending on requirements**

### **Question 9**

Which cron job parameter helps prevent memory issues during long-running processes?

A) --time-limit B) --memory-limit C) --batch-limit D) All of the above

**Correct Answer: D) All of the above**

### **Question 10**

What is the correct format for configuring a database DSN in Mautic?

A) mysql://user:password@host:port/database B) pdo\_mysql://user:password@host:port/database C) Configured via individual parameters in local.php D) Both A and C are valid approaches

**Correct Answer: C) Configured via individual parameters in local.php**

### **Question 11**

Which setting controls whether Mautic uses Composer for plugin management?

A) composer\_enabled B) marketplace\_enabled C) update\_method D) composer\_updates

**Correct Answer: D) composer\_updates**

### **Question 12**

What is the primary purpose of the tracking code configuration?

A) To enable Google Analytics integration B) To track website visitors and page views C) To monitor email opens D) To track API usage

**Correct Answer: B) To track website visitors and page views**

### **Question 13**

Which command rebuilds Mautic segments?

A) bin/console mautic:segments:rebuild B) bin/console mautic:segments:update C) bin/console mautic:segments:refresh D) bin/console mautic:update:segments

**Correct Answer: B) bin/console mautic:segments:update**

### **Question 14**

What is the recommended approach for handling large CSV imports?

A) Import everything through the web interface B) Use background processing with cron jobs C) Split files into smaller chunks manually D) Increase PHP memory limits indefinitely

**Correct Answer: B) Use background processing with cron jobs**

### **Question 15**

Which configuration option enables queue-based email processing?

A) email\_delivery\_method = queue B) mailer\_spool\_type = memory C) Configure email to use spool directory D) Set messenger transport for email

**Correct Answer: D) Set messenger transport for email**

### **Question 16**

What is the purpose of the 'do\_not\_track\_ips' configuration?

A) To prevent certain IPs from being tracked B) To block spam traffic C) To exclude internal team members from analytics D) All of the above

**Correct Answer: D) All of the above**

### **Question 17**

Which file should be excluded from version control in a Mautic installation?

A) app/config/local.php B) .htaccess C) index.php D) composer.json

**Correct Answer: A) app/config/local.php**

### **Question 18**

What is the correct way to configure CORS settings for Mautic?

A) Edit .htaccess file only B) Configure in System Settings > CORS Settings C) Set PHP headers manually D) CORS configuration is not available in Mautic

**Correct Answer: B) Configure in System Settings > CORS Settings**

### **Question 19**

Which command processes campaign actions and decisions?

A) bin/console mautic:campaigns:execute B) bin/console mautic:campaigns:trigger C) bin/console mautic:campaigns:process D) bin/console mautic:campaigns:run

**Correct Answer: B) bin/console mautic:campaigns:trigger**

### **Question 20**

What is the recommended frequency for running mautic:segments:update?

A) Every minute B) Every 5-15 minutes C) Once per hour D) Once per day

**Correct Answer: B) Every 5-15 minutes**

## **Category 2: Installation, Deployment, and Updates (25% - 20 Questions)**

### **Question 21**

Which installation method is recommended for production Mautic deployments?

A) ZIP package installation B) Composer-based installation C) Git clone installation D) Docker container deployment

**Correct Answer: B) Composer-based installation**

### **Question 22**

What is the first step before performing any Mautic update?

A) Clear the cache B) Update plugins first C) Create a tested backup D) Check system requirements

**Correct Answer: C) Create a tested backup**

### **Question 23**

Which command completes a Mautic update after downloading new files?

A) bin/console mautic:update:finish B) bin/console mautic:update:apply --finish C) bin/console mautic:update:complete D) bin/console doctrine:migrations:migrate

**Correct Answer: B) bin/console mautic:update:apply --finish**

### **Question 24**

What is DDEV primarily used for in Mautic development?

A) Production deployment B) Local development environment C) Database management D) Email testing

**Correct Answer: B) Local development environment**

### **Question 25**

Which file must be present in a Mautic composer project?

A) composer.json B) composer.lock C) .env D) All of the above

**Correct Answer: D) All of the above**

### **Question 26**

What is the purpose of the docroot directory in Composer-based installations?

A) Database storage location B) Log file directory C) Web server document root D) Cache storage location

**Correct Answer: C) Web server document root**

### **Question 27**

Which command creates a new Mautic project using Composer?

A) composer install mautic/mautic B) composer create-project mautic/recommended-project C) composer require mautic/core D) composer new mautic/project

**Correct Answer: B) composer create-project mautic/recommended-project**

### **Question 28**

What should you do if a Mautic update fails during the schema update phase?

A) Restore from backup immediately B) Visit /s/update/schema to check pending migrations C) Delete the cache directory D) Reinstall Mautic completely

**Correct Answer: B) Visit /s/update/schema to check pending migrations**

### **Question 29**

Which directory contains Mautic's cache files?

A) var/cache B) app/cache C) cache/ D) tmp/cache

**Correct Answer: A) var/cache**

### **Question 30**

What is the recommended approach for deploying Mautic to multiple environments?

A) Manual file copying B) FTP upload of zip packages C) Automated CI/CD pipeline with proper testing D) Direct database replication

**Correct Answer: C) Automated CI/CD pipeline with proper testing**

### **Question 31**

Which file contains the scaffolding configuration for Composer projects?

A) composer.json B) scaffold.json C) .env D) config/scaffold.yml

**Correct Answer: A) composer.json**

### **Question 32**

What happens when you run 'composer update' in a Mautic project?

A) Only Mautic core is updated B) All packages are updated to their latest compatible versions C) Only security updates are applied D) The database schema is automatically updated

**Correct Answer: B) All packages are updated to their latest compatible versions**

### **Question 33**

Which backup strategy is most comprehensive for Mautic?

A) Database backup only B) File system backup only C) Both database and file system backups with tested restore procedures D) Configuration files backup only

**Correct Answer: C) Both database and file system backups with tested restore procedures**

### **Question 34**

What is the purpose of the composer.lock file?

A) To prevent unauthorized access B) To lock specific package versions for consistent deployments C) To lock the composer installation D) To prevent updates

**Correct Answer: B) To lock specific package versions for consistent deployments**

### **Question 35**

Which command installs Mautic from the command line?

A) bin/console mautic:setup B) bin/console mautic:install C) bin/console app:install D) bin/console install:mautic

**Correct Answer: B) bin/console mautic:install**

### **Question 36**

What is the recommended web server configuration for production Mautic?

A) Apache with mod\_php B) Nginx with PHP-FPM C) Either Apache with mod\_php or Nginx with PHP-FPM D) Built-in PHP development server

**Correct Answer: C) Either Apache with mod\_php or Nginx with PHP-FPM**

### **Question 37**

Which directory should be writable by the web server in a Mautic installation?

A) var/ directory only B) media/ directory only C) var/, media/, and cache/ directories D) The entire Mautic directory

**Correct Answer: C) var/, media/, and cache/ directories**

### **Question 38**

What is the benefit of using the Mautic Recommended Project structure?

A) Easier updates and dependency management B) Better security through separation of concerns C) Simplified backup procedures D) All of the above

**Correct Answer: D) All of the above**

### **Question 39**

Which file format is used for Mautic configuration in Composer-based installations?

A) YAML B) JSON C) PHP D) XML

**Correct Answer: C) PHP**

### **Question 40**

What should you verify after completing a Mautic update?

A) Login functionality and basic operations B) Email sending capability C) Database integrity D) All of the above

**Correct Answer: D) All of the above**

## **Category 3: Integration and Development (30% - 24 Questions)**

### **Question 41**

Which authentication method is recommended for Mautic API access?

A) Basic authentication B) OAuth 2.0 C) API key authentication D) Session-based authentication

**Correct Answer: B) OAuth 2.0**

### **Question 42**

What is the maximum rate limit for API requests in Mautic?

A) 100 requests per minute B) 1000 requests per hour C) Configurable based on server capacity D) No rate limiting by default

**Correct Answer: C) Configurable based on server capacity**

### **Question 43**

Which webhook event type is triggered when a contact is created?

A) mautic.contact\_pre\_save B) mautic.contact\_post\_save C) mautic.lead\_post\_save D) mautic.contact\_created

**Correct Answer: C) mautic.lead\_post\_save**

### **Question 44**

What is the correct way to test webhook endpoints during development?

A) Use production Mautic instance B) Use ngrok or similar tunneling service for local testing C) Test directly on production servers D) Webhook testing is not necessary

**Correct Answer: B) Use ngrok or similar tunneling service for local testing**

### **Question 45**

Which tool is commonly used for API testing and development?

A) Postman B) curl C) Insomnia D) All of the above

**Correct Answer: D) All of the above**

### **Question 46**

What is the primary advantage of using n8n with Mautic?

A) Better email deliverability B) Visual workflow automation and integration C) Improved database performance D) Enhanced security features

**Correct Answer: B) Visual workflow automation and integration**

### **Question 47**

Which plugin type is most commonly used for external system integrations?

A) Form plugins B) Email plugins C) Integration plugins D) Campaign plugins

**Correct Answer: C) Integration plugins**

### **Question 48**

What is the recommended approach for handling API errors in integrations?

A) Ignore errors and continue processing B) Implement retry logic with exponential backoff C) Log errors but don't retry D) Stop all processing on first error

**Correct Answer: B) Implement retry logic with exponential backoff**

### **Question 49**

Which database table stores contact information in Mautic?

A) contacts B) leads C) users D) persons

**Correct Answer: B) leads**

### **Question 50**

What is the purpose of the Mautic Marketplace?

A) To sell Mautic licenses B) To distribute community plugins and themes C) To provide hosting services D) To offer training courses

**Correct Answer: B) To distribute community plugins and themes**

### **Question 51**

Which CSS framework is used in Mautic's default theme?

A) Bootstrap B) Foundation C) Bulma D) Tailwind CSS

**Correct Answer: A) Bootstrap**

### **Question 52**

What is the recommended way to add custom JavaScript to Mautic forms?

A) Edit core JavaScript files directly B) Use theme customization and form templates C) Inject via external scripts D) Custom JavaScript is not supported

**Correct Answer: B) Use theme customization and form templates**

### **Question 53**

Which PHP framework does Mautic use as its foundation?

A) Laravel B) CodeIgniter C) Symfony D) Zend Framework

**Correct Answer: C) Symfony**

### **Question 54**

What is the purpose of Doctrine ORM in Mautic?

A) Email template rendering B) Database abstraction and entity management C) Caching management D) Session handling

**Correct Answer: B) Database abstraction and entity management**

### **Question 55**

Which method is used to create custom entities in Mautic?

A) Direct database table creation B) Doctrine entity annotations and migrations C) JSON configuration files D) XML mapping files

**Correct Answer: B) Doctrine entity annotations and migrations**

### **Question 56**

What is the correct way to handle large data synchronization between Mautic and external systems?

A) Process all data at once B) Use batch processing with appropriate chunk sizes C) Use real-time synchronization only D) Manual data export/import

**Correct Answer: B) Use batch processing with appropriate chunk sizes**

### **Question 57**

Which service container pattern does Mautic use for dependency injection?

A) Manual dependency injection B) Symfony's service container C) Custom dependency injection container D) No dependency injection is used

**Correct Answer: B) Symfony's service container**

### **Question 58**

What is the recommended approach for plugin development in Mautic?

A) Modify core files directly B) Create separate plugin bundles with proper structure C) Use only configuration-based customizations D) Plugin development is not supported

**Correct Answer: B) Create separate plugin bundles with proper structure**

### **Question 59**

Which testing framework is recommended for Mautic plugin development?

A) PHPUnit B) Behat C) Codeception D) All of the above can be used

**Correct Answer: D) All of the above can be used**

### **Question 60**

What is the purpose of the mautic:integration:fetchleads command?

A) To generate new leads B) To synchronize data from external integrations C) To update lead scores D) To process email campaigns

**Correct Answer: B) To synchronize data from external integrations**

### **Question 61**

Which format is commonly used for API data exchange in Mautic?

A) XML only B) JSON only C) Both JSON and XML D) CSV format

**Correct Answer: C) Both JSON and XML**

### **Question 62**

What is the recommended way to handle sensitive data in Mautic integrations?

A) Store in plain text configuration files B) Use environment variables and encryption C) Hardcode in source code D) Store in the database unencrypted

**Correct Answer: B) Use environment variables and encryption**

### **Question 63**

Which event system does Mautic use for plugin hooks?

A) Custom event system B) Symfony EventDispatcher C) jQuery events D) WordPress-style hooks

**Correct Answer: B) Symfony EventDispatcher**

### **Question 64**

What is the primary benefit of using Git branches in Mautic development?

A) Faster performance B) Parallel development and feature isolation C) Better security D) Reduced disk space usage

**Correct Answer: B) Parallel development and feature isolation**

## **Category 4: Security, Performance, and Maintenance (20% - 16 Questions)**

### **Question 65**

Which command should be used to securely set file ownership for Mautic?

A) chmod 777 -R /path/to/mautic B) chown -R www-data:www-data /path/to/mautic C) chown -R root:root /path/to/mautic D) File ownership doesn't matter for Mautic

**Correct Answer: B) chown -R www-data:www-data /path/to/mautic**

### **Question 66**

What is the recommended approach for managing Mautic in a load-balanced environment?

A) Use sticky sessions B) Configure shared storage for media and cache C) Run separate instances with no coordination D) Load balancing is not supported

**Correct Answer: B) Configure shared storage for media and cache**

### **Question 67**

Which caching strategy provides the best performance for Mautic?

A) File-based caching only B) Database caching only C) Redis or Memcached for application cache with APCu for opcode cache D) No caching is recommended

**Correct Answer: C) Redis or Memcached for application cache with APCu for opcode cache**

### **Question 68**

What is the primary security risk when working with shell commands?

A) Slow performance B) Command injection and privilege escalation C) High memory usage D) Network connectivity issues

**Correct Answer: B) Command injection and privilege escalation**

### **Question 69**

Which email authentication methods should be implemented for optimal deliverability?

A) SPF only B) DKIM only C) SPF, DKIM, and DMARC D) No authentication is necessary

**Correct Answer: C) SPF, DKIM, and DMARC**

### **Question 70**

What is the recommended approach for monitoring Mautic performance in production?

A) Manual checking only B) Application Performance Monitoring (APM) tools and server monitoring C) Database monitoring only D) Email delivery monitoring only

**Correct Answer: B) Application Performance Monitoring (APM) tools and server monitoring**

### **Question 71**

Which MJML component is used for responsive email layout?

A) mj-container B) mj-section C) mj-column D) All of the above

**Correct Answer: D) All of the above**

### **Question 72**

What is the purpose of implementing database indexing in Mautic?

A) To increase storage space B) To improve query performance C) To enhance security D) To enable backups

**Correct Answer: B) To improve query performance**

### **Question 73**

Which tool is recommended for monitoring email deliverability and sender reputation?

A) Google Analytics B) Google Postmaster Tools C) Social media analytics D) Database monitoring tools

**Correct Answer: B) Google Postmaster Tools**

### **Question 74**

What is the recommended approach for handling sensitive configuration data?

A) Store in version control B) Use environment variables and secure storage C) Hardcode in application files D) Store in public directories

**Correct Answer: B) Use environment variables and secure storage**

### **Question 75**

Which maintenance task should be performed regularly on Mautic databases?

A) Table optimization and cleanup B) Index rebuilding C) Statistics updates D) All of the above

**Correct Answer: D) All of the above**

### **Question 76**

What is the primary benefit of using SSL/TLS certificates with Mautic?

A) Faster performance B) Data encryption and authentication C) Better email deliverability D) Reduced server load

**Correct Answer: B) Data encryption and authentication**

### **Question 77**

Which approach is recommended for managing multiple Mautic environments?

A) Use the same configuration for all environments B) Environment-specific configurations with Infrastructure as Code C) Manual configuration for each environment D) Copy production settings to all environments

**Correct Answer: B) Environment-specific configurations with Infrastructure as Code**

### **Question 78**

What is the recommended frequency for reviewing and updating Mautic security settings?

A) Once per year B) Monthly C) Quarterly with immediate updates for security patches D) Only when problems occur

**Correct Answer: C) Quarterly with immediate updates for security patches**

### **Question 79**

Which strategy is most effective for optimizing large Mautic databases?

A) Increase server memory only B) Implement data archiving, partitioning, and query optimization C) Use faster storage devices only D) Add more CPU cores only

**Correct Answer: B) Implement data archiving, partitioning, and query optimization**

### **Question 80**

What is the most important factor in maintaining long-term Mautic system performance?

A) Hardware upgrades only B) Regular monitoring, maintenance, and proactive optimization C) Using the latest software versions only D) Increasing database connections only

**Correct Answer: B) Regular monitoring, maintenance, and proactive optimization**