QR Trackr – Effortless QR Codes for Your WordPress Site

Welcome to **QR Trackr**, the easiest way to create, manage, and track QR codes right from your WordPress dashboard. No technical skills required!

What Can QR Trackr Do For You?

- **Create QR codes** for any page, post, or custom link in just a few clicks.
- **Track every scan** – see how many times your QR codes are used and where.
- **Download and print** beautiful QR codes (PNG or SVG).
- **Change destinations anytime** – update where your QR code points, even after printing!
- **Easy-to-read links** – your QR codes use simple, pretty URLs like yourwebsite.com/qr/abc123.
- **Works everywhere** – perfect for flyers, menus, business cards, and more.

Quick Start (No Tech Skills Needed)

1. **Install the Plugin**

   - Download the latest version from GitHub Releases (https://github.com/michaelerps/wp-qr-trackr/releases).
   - In your WordPress dashboard, go to **Plugins > Add New > Upload Plugin**.
   - Upload the ZIP file and click **Activate**.

2. **Create Your First QR Code**

   - Click **QR Trackr** in your WordPress menu.
   - Choose a post, page, or enter any link you want.
   - Click **Generate QR Code**.
   - Download your QR code and use it anywhere!

3. **See Your Results**

   - Come back anytime to see how many times your QR code has been scanned.
   - Change the destination link whenever you want – your printed QR code will always work!

Why Use QR Trackr?

- **No coding, no hassle** – everything is point-and-click.
- **Instant analytics** – see what's working and what's not.
- **Safe and private** – your data stays on your site.
- **Perfect for small businesses, events, restaurants, and more!**

Upgrading? Read This!

If you're updating QR Trackr to a new version:

1. **Deactivate and reactivate the plugin** (in Plugins > Installed Plugins) to make sure everything is up to date.
2. Go to **Settings > Permalinks** and click **Save Changes** (this helps your QR codes work everywhere).
3. If you made QR codes before, just click **Regenerate** next to each one to update them.
4. That's it! Your QR codes will now use the new, easy-to-read links.

Need Help?

- If your QR code doesn't work, try step 2 above (Settings > Permalinks > Save Changes).
- Still stuck? Reach out on the GitHub project page (https://github.com/michaelerps/wp-qr-trackr/issues) or ask your website manager for help.

QR Trackr makes QR codes simple, powerful, and fun. Try it today!

Features

- Generate QR codes for WordPress posts/pages or external URLs
- Track QR code scans and access statistics
- Mobile-friendly admin interface
- Support for both pretty permalinks and plain permalinks
- Fallback QR code generation when server libraries are unavailable
- Debug logging for troubleshooting

New Features (v1.0.3)

- Users can now set a **Custom URL** as the destination for a QR code, in addition to WordPress posts/pages and external URLs.
- Multiple QR codes can be created for the same destination (no longer limited to one QR per destination).
- Both client-side and server-side validation ensure only valid URLs or published posts/pages are accepted.

QR Code Creation Form

When creating a new QR code, you can choose the destination type:

- **WordPress Post/Page**: Select from a list of published posts and pages.
- **External URL**: Enter any valid URL (must start with http:// or https://).
- **Custom URL**: Enter any valid URL (must start with http:// or https://).

The form will only allow submission if the selected destination is valid. Invalid input will show a clear error message.

Installation

1. Upload the plugin files to /wp-content/plugins/wp-qr-trackr
2. Activate the plugin through the WordPress admin interface

3. Configure the plugin settings under the QR Trackr menu

Requirements

- WordPress 5.0 or higher
- PHP 7.4 or higher
- Pretty permalinks enabled (recommended)
- Write permissions for the uploads directory

Production Dependency Requirements

The following Composer packages are required for production builds:

Package/Folder	Needed for Production?	Why?
endroid/qr-code	Yes	QR code generation
bacon/bacon-qr-code	Yes	Dependency of endroid/qr-code
dasprid/enum	Yes	Dependency of bacon/bacon-qr-code
Everything else listed	No	Dev/test/code quality only

To keep your plugin lean, only these production dependencies should be included in the distributed zip. All other packages (test, QA, code quality, etc.) should be excluded from production builds.

Modular Architecture Philosophy

Why a Finite Plugin is the Perfect Example

A QR code generator and URL redirect service is intentionally finite in scope, making it an ideal showcase for a well-built, standards-compliant WordPress plugin. Its simplicity is a strength: with a clear, bounded feature set, there is no room for accidental complexity or hidden technical debt. This allows the foundational architecture, code quality, and adherence to best practices to shine through without distraction.

Why This Matters

- **Demonstrates Best Practices**: With no "feature bloat" to hide behind, every line of code must be purposeful, secure, and maintainable. This plugin is a living example of how to structure, document, and test a WordPress plugin the right way.
- **Educational Value**: Developers can study the codebase to learn about modular design, separation of concerns, error handling, security, and WordPress standards—all in a context that is easy to understand and reason about.
- **Foundation for Growth**: By starting with a robust, standards-compliant base, the plugin can be safely extended with new features or adapted for more complex use cases, without sacrificing quality or maintainability.
- **Transparency**: The limited scope means that every architectural decision is visible and justifiable. This transparency is invaluable for code reviews, audits, and onboarding new contributors.

Showcasing the Foundation

Because the plugin is so focused, it serves as a "reference implementation" for:

- Modular WordPress plugin structure
- Secure and sanitized input/output handling

- Proper use of hooks, actions, and filters
- Automated testing and CI/CD integration
- Documentation and developer onboarding

We are preparing a video walkthrough to explain what this plugin can do, how it is built, and why this approach is helpful for both new and experienced WordPress developers. By highlighting the core architecture and design decisions, we aim to inspire others to build plugins that are not just functional, but also maintainable, secure, and a pleasure to work with.

Learning from Experience

The QR Trackr plugin's modular architecture has evolved through practical experience and continuous improvement. Our recent restructuring has taught us valuable lessons about WordPress plugin development, which we've documented in our .cursorrules file. This living document serves as both a guide and a historical record of our development journey.

Cursor Rules as a Foundation

The .cursorrules file acts as a central repository of best practices and lessons learned. It helps us:

1. **Prevent Past Mistakes**: By documenting issues we've encountered (like rewrite rule initialization problems), we ensure future development avoids similar pitfalls.

2. **Maintain Consistency**: The rules provide a clear framework for how modules should be structured, tested, and documented, ensuring consistency across the codebase.

3. **Guide Future Development**: New features and modules can be developed with confidence, knowing they follow established patterns and best practices.

4. **Facilitate Onboarding**: New developers can quickly understand our architectural decisions and coding standards through the documented rules.

Strengthening the Modular Foundation

Our modular architecture is made more robust through the .cursorrules file in several ways:

1. **Clear Module Boundaries**: The rules define explicit boundaries between modules, preventing feature creep and maintaining separation of concerns.

2. **Standardized Error Handling**: Each module follows consistent error handling and logging patterns, making debugging more efficient.

3. **Performance Optimization**: Module-specific performance guidelines ensure each component is optimized without affecting others.

4. **Security Best Practices**: Security rules are applied consistently across all modules, creating a more secure overall system.

5. **Testing Requirements**: Each module must meet specific testing criteria, ensuring reliability and maintainability.

Module Organization

The plugin is organized into the following modules:

- module-admin.php: Admin interface and settings
- module-ajax.php: AJAX handlers for dynamic operations
- module-qr.php: QR code generation and management
- module-requirements.php: System requirements checking
- module-activation.php: Plugin activation and deactivation
- module-settings.php: Settings management
- module-rewrite.php: URL rewriting and tracking
- module-debug.php: Debug logging and troubleshooting

Benefits of This Approach

1. **Maintainability**

    - Easier to locate and fix issues
    - Simpler to add new features
    - Better code organization
    - Reduced technical debt

2. **Scalability**

    - New features can be added as separate modules
    - Existing modules can be enhanced independently
    - Better performance through focused optimization
    - Easier to implement caching strategies

3. **Testability**

    - Modules can be tested in isolation
    - Easier to write unit tests
    - Better error isolation
    - Simpler debugging process

4. **Security**

    - Clear boundaries for data handling
    - Easier to implement security measures
    - Better control over access points
    - Simpler to audit and review

Moving Forward

As the plugin evolves, we maintain these principles by:

1. **Documentation**

    - Clear module purpose and boundaries
    - API documentation for module interactions
    - Usage examples and best practices
    - Change logs and upgrade guides

2. **Code Standards**

    - Consistent coding style
    - Comprehensive error handling
    - Proper input validation

- Security best practices

3. **Testing**
    - Unit tests for each module
    - Integration tests for module interactions
    - Performance testing
    - Security testing

4. **User Experience**
    - Consistent interface design
    - Clear error messages
    - Helpful documentation
    - Responsive design

Development Guidelines

When contributing to the plugin:

1. **New Features**
    - Create a new module if functionality is distinct
    - Extend existing modules if closely related
    - Document module purpose and usage
    - Include tests and documentation

2. **Bug Fixes**
    - Identify the affected module
    - Fix the issue without breaking module boundaries
    - Add tests to prevent regression
    - Update documentation if needed

3. **Performance**
    - Profile module performance
    - Optimize critical paths
    - Implement caching where appropriate
    - Monitor resource usage

4. **Security**
    - Validate all input
    - Sanitize all output
    - Use WordPress security functions
    - Follow security best practices

Contributing

We welcome contributions to QR Trackr! Please follow these guidelines:

1. Fork the repository
2. Create a feature branch
3. Make your changes

4. Submit a pull request

License

This plugin is licensed under the GPL v2 or later.

Credits

Developed by [Your Name/Company]

Recent Updates

Version 1.0.2

- Added destination verification toggle
- Enhanced URL validation and sanitization
- Improved error handling and user feedback
- Added duplicate destination checking
- Enhanced security measures
- Improved mobile responsiveness
- Added detailed error logging
- Fixed permission issues
- Added proper nonce verification
- Improved AJAX handling

Version 1.0.1

- Added fallback QR code generation
- Improved library detection
- Enhanced error handling
- Added user-friendly messages
- Fixed activation issues

Version 1.0.0

- Initial release
- Basic QR code generation
- WordPress post/page linking
- External URL support
- Basic tracking functionality

Security Features

- Input sanitization for all user data
- Nonce verification for all forms
- Capability checking for admin actions
- URL validation and sanitization
- XSS prevention
- CSRF protection

Development

File Structure

```
wp-qr-trackr/
├──── assets/
│     ├──── css/
│     │     └──── admin.css
│     └──── js/
│           └──── admin.js
├──── includes/
│     ├──── module-admin.php
│     ├──── module-ajax.php
│     ├──── module-rewrite.php
│     ├──── module-requirements.php
│     └──── module-utils.php
├──── wp-qr-trackr.php
└──── README.md
```

Adding New Features

1. Create a new module in the includes/ directory
2. Register your module in the main plugin file
3. Follow WordPress coding standards
4. Add proper documentation
5. Include error handling
6. Test thoroughly

Troubleshooting

Common Issues

1. **QR Code Generation Fails**

   - Check if the QR code library is properly installed
   - Verify upload directory permissions
   - Check server requirements

2. **URL Verification Issues**

   - Disable URL verification in settings
   - Check if the URL is accessible
   - Verify URL format

3. **Permission Errors**

   - Ensure you're logged in as an administrator
   - Check user capabilities
   - Verify nonce values

Debug Mode

Enable debug mode by adding to wp-config.php:

```
define('QR_TRACKR_DEBUG', true);
```

Support

For support, please:

1. Check the documentation
2. Search existing issues
3. Create a new issue if needed

Changelog

1.0.2

- Added destination verification toggle
- Enhanced URL validation
- Improved error handling
- Added security features
- Fixed permission issues
- Added proper nonce verification
- Improved AJAX handling
- Enhanced mobile responsiveness

1.0.1

- Added fallback QR generation
- Improved library detection
- Enhanced error handling
- Added user messages
- Fixed activation issues

1.0.0

- Initial release

Educational & Creative Resources: Video Scripts

Combined Primary Script (For All Audiences)

Tell Them What You're Going to Tell Them

Welcome! In this video, we'll show you why QR Trackr—a simple QR code generator and URL redirect plugin—is a perfect example of a well-built, standards-compliant WordPress plugin. We'll explain the philosophy behind its design, walk through its modular architecture, and demo how it works for users, developers, and product owners alike. By the end, you'll see how a strong foundation benefits everyone, from beginners to experts.

Tell Them

- **Philosophy & Simplicity:** QR Trackr is intentionally finite. Its focused feature set means every line of code is purposeful, secure, and maintainable. This makes it a living example of best practices in WordPress plugin development.

- **Modular Architecture:** Each part of the plugin—admin UI, QR code generation, AJAX, tracking, and more—is separated into its own module. This makes the codebase easy to understand, extend, and maintain.
- **Demo for All:**
    - *For Users/Novices:* Easily create a QR code for any page or link, download it, and track scans in real time.
    - *For Developers:* Explore the codebase to see modular design, security, and testing in action. Each module is focused and well-documented.
    - *For Product Owners:* See how the plugin's structure makes it reliable, secure, and easy to extend or maintain as your needs grow.
- **Best Practices:** Input is sanitized, output is escaped, and security is enforced throughout. Automated tests and clear documentation ensure reliability and ease of onboarding.

Tell Them What You've Told Them

To recap: QR Trackr's simplicity lets its architecture and best practices shine. Whether you're a user, developer, or product owner, you benefit from a plugin that's easy to use, easy to maintain, and ready to grow. Check out our documentation or codebase to learn more, and stay tuned for more educational content!

Developer-Focused Script

Tell Them What You're Going to Tell Them

Welcome! In this video, I'll show you why QR Trackr is a model WordPress plugin for developers. We'll cover its modular architecture, code quality, and best practices, and I'll walk you through the codebase and a live demo of how the plugin works in practice. By the end, you'll see how a finite plugin can be a perfect foundation for scalable, maintainable WordPress development.

Tell Them

- *Philosophy and Architecture:* QR Trackr is intentionally simple, so the architecture and code quality are front and center. The plugin is split into modules: admin, AJAX, QR code generation, rewrite rules, and more. Each module has a single responsibility, making the codebase easy to navigate and extend.
- *Demo: Exploring the Codebase:* Here's the includes/ directory, where each module lives. Notice how module-admin.php handles only admin UI, while module-qr.php is responsible for QR code generation. The main plugin file just bootstraps these modules—no business logic is mixed in.
- *Demo: Creating and Tracking a QR Code:* In the WordPress admin, go to the QR Trackr menu. Click 'Create New QR Code'. Choose a destination—either a WordPress post/page or an external URL. Click 'Create'. The plugin generates a QR code, stores the link, and displays it in the list. Scan the QR code with your phone. The plugin tracks the scan and updates the stats. Check the stats page to see the scan count update in real time.
- *Best Practices in Action:* All input is sanitized, output is escaped, and security is enforced at every step. The codebase is fully documented, and automated tests ensure reliability. This is a reference implementation for modern WordPress plugin development.

Tell Them What You've Told Them

We've explored QR Trackr's modular codebase, seen a live demo, and discussed how its simplicity lets best practices shine. As a developer, you can use this as a foundation for your own plugins, confident that you're building on solid ground.

Product Owner-Focused Script

Tell Them What You're Going to Tell Them

Hi! In this video, I'll show you why QR Trackr is a great example of a well-built WordPress plugin from a product owner's perspective. We'll look at how its design makes it reliable, secure, and easy to extend, and I'll demo how it works in the WordPress admin. By the end, you'll see why investing in quality foundations pays off.

Tell Them

- *Philosophy and Value:* QR Trackr is focused: it generates QR codes and tracks scans. This limited scope means the plugin is easy to maintain, secure, and reliable—no hidden complexity or technical debt.
- *Demo: Using the Plugin:* In your WordPress dashboard, open the QR Trackr menu. Click 'Create New QR Code'. Select a destination—either a page on your site or an external link. Click 'Create'. Instantly, you get a QR code you can download or print. Every scan is tracked, and you can view stats at a glance.
- *Why This Matters for Product Owners:* Because the plugin is modular and standards-compliant: It's easy to add new features or integrate with other systems. Security and performance are built in. Maintenance is straightforward, reducing long-term costs. Documentation and code quality make onboarding new developers easy.

Tell Them What You've Told Them

We've seen how QR Trackr's focused design and quality code make it a reliable, extensible solution. As a product owner, you can trust that this plugin is built to last and ready to grow with your needs.

Novice/Beginner-Focused Script

Tell Them What You're Going to Tell Them

Hey there! If you've never built a WordPress plugin before, this video is for you. I'll show you how QR Trackr—a simple QR code generator—demonstrates all the basics of a well-made plugin. We'll walk through what the plugin does, how it's organized, and I'll show you step-by-step how to use it.

Tell Them

- *What is QR Trackr?* QR Trackr is a plugin that lets you create QR codes for your website's pages or any link you want. It also tracks how many times each QR code is scanned.
- *Demo: Step-by-Step Usage:* Go to your WordPress dashboard and find the QR Trackr menu. Click 'Create New QR Code'. Choose where you want the QR code to point—maybe your homepage, a blog post, or an external website. Click 'Create'. You'll see a QR code appear—download it or print it out. When someone scans the code, the plugin tracks it. You can see how many times it's been scanned on the stats page.
- *How is the Plugin Built?* Behind the scenes, QR Trackr is organized into separate pieces, or 'modules'. Each module has a specific job—like handling the admin screen, generating QR codes, or tracking scans. This makes the plugin easier to understand, update, and keep secure.

Tell Them What You've Told Them

So, we've learned what QR Trackr does, how to use it, and why its simple, organized structure is a great example for anyone starting out with WordPress plugins. If you want to learn more, check out the documentation or try exploring the codebase yourself!

Recent Changes

- The QR code admin table now includes a compact, padded ID column for each QR code, making it easier to reference and manage individual codes.
- The edit QR code destination feature uses a secure, consistent nonce and only one AJAX handler for reliability and security.
- The plugin's admin UI is now more responsive and visually balanced, with improved table column sizing and padding for better readability.

v1.0.4 (2025-06-17)

Security

- All admin AJAX actions (edit, delete, regenerate) now use their own separate, localized nonces for maximum security.

Contributor Notes

- PHPCS (WordPress Coding Standards) requires at least 1GB of memory to lint the codebase. For large codebases or CI, 4GB is recommended. See .cursorrules for details.

---

wp-qr-trackr

... (existing content) ...