












ArtMin96 / yii2-mcp-server Public

<> Code Issues 1 Pull requests 2 Actions Projects Security and quality

1 Branch 0 Tags Go to file Go to file <> Code ...

artmin96 and claude  Relate... delete MCP Server with Com...  

5cec3ff · 9 months ago

-  9 months ago
-  9 months ago
-  9 months ago
-  9 months ago
-  9 months ago
-  9 months ago
-  package... 9 months ago
-  tsconfig.js... MCP Ser... 9 months ago

README

Yii2 MCP Server

A Model Context Protocol (MCP) server for Yii2 framework, providing database schema inspection, console command execution, and project management tools for Claude Code.

Features

Database Tools

- **db_list_tables**: List all database tables with details (rows, size, comments)

- **db_describe_table**: Get detailed schema information for a specific table
- **db_table_relationships**: Analyze foreign key relationships
- **db_execute_query**: Execute SELECT queries safely with results

Database Optimization & Analysis (NEW)

- **db_analyze_table**: Comprehensive table analysis including fragmentation, index efficiency, and storage engine recommendations
- **db_get_index_suggestions**: Smart index recommendations based on table structure and query patterns
- **db_profile_query**: Query performance profiling with execution plans and optimization suggestions
- **db_monitor_connections**: Real-time database connection monitoring with alerting

Yii2 Console Tools

- **yii_list_commands**: List all available Yii console commands with descriptions
- **yii_command_help**: Get detailed help for specific commands
- **yii_execute_command**: Execute safe console commands (dangerous commands blocked)
- **yii_list_migrations**: Show migration status (applied and pending)

Model & Project Tools

- **yii_list_models**: List all ActiveRecord models with relationships
- **yii_analyze_model**: Analyze specific model classes
- **project_structure**: Get overview of project directory structure
- **module_info**: Detailed information about specific modules
- **config_inspect**: Inspect application configuration files

Advanced Module Analysis

- **module_detailed_analysis**: Comprehensive module analysis including controllers, models, assets, widgets, jobs, services, and traits
- **asset_list_bundles**: List all asset bundles with CSS/JS files and dependencies
- **asset_analyze_dependencies**: Analyze asset dependency chains and registration order
- **widget_list**: Discover widgets and their associated asset bundles

Migration Management

- **migration_create**: Create new database migrations
- **migration_diff**: Generate migrations from model/database differences
- **migration_rollback**: Rollback migrations safely
- **migration_generate_from_model**: Generate migrations from ActiveRecord models

Code Generation Tools

- **generate_crud**: Generate CRUD controllers and views
- **generate_model**: Generate ActiveRecord models from database tables
- **generate_api**: Generate REST API controllers
- **generate_form**: Generate form classes

Cache Management

- **cache_clear**: Clear application cache with options
- **cache_inspect**: Inspect cache configuration and status
- **cache_performance**: Analyze cache performance and hit rates

RBAC (Role-Based Access Control)

- **rbac_permission_tree**: Visualize permission hierarchy
- **rbac_user_access**: Check user permissions and access levels
- **rbac_role_analysis**: Analyze role assignments and permissions
- **rbac_access_debug**: Debug access control issues

Queue System Management

- **queue_job_status**: Monitor queue job status and progress
- **queue_retry_failed**: Retry failed queue jobs
- **queue_worker_status**: Check queue worker status
- **queue_clear_failed**: Clear failed jobs from queue

Logging & Debugging

- **log_error_analysis**: Analyze error logs and patterns
- **log_performance_issues**: Identify performance bottlenecks in logs
- **log_search**: Search through application logs

- **log_tail**: Real-time log monitoring

Advanced Database Analysis

- **db_index_suggestions**: Advanced index optimization recommendations
- **db_query_profiler**: Detailed query performance profiling
- **db_table_analysis**: In-depth table structure analysis
- **db_connection_monitor**: Real-time database connection monitoring

Testing & Quality Assurance

- **test_run**: Execute test suites
- **test_generate**: Generate test cases
- **test_coverage_report**: Generate code coverage reports

Performance Analysis

- **performance_profile**: Application performance profiling
- **asset_performance_analysis**: Frontend asset performance analysis

Installation

Option 1: NPM Install (Recommended)

```
npm install -g yii2-mcp-server
```



Then configure Claude Code from your Yii2 project directory:

```
cd /path/to/your/yii2/project
claude mcp add yii2 yii2-mcp-server
```



Option 2: Manual Installation

1. Clone and Install

```
git clone https://github.com/ArtMin96/yii2-mcp-server.git
cd yii2-mcp-server
npm install
npm run build
```



2. Configure Claude Code

Using Claude CLI:

```
cd /path/to/your/yii2/project
claude mcp add yii2 node /path/to/yii2-mcp-server/dist/index.js
```



Manual JSON Configuration:

```
{
  "mcpServers": {
    "yii2": {
      "command": "node",
      "args": ["/path/to/yii2-mcp-server/dist/index.js"],
      "cwd": "/path/to/your/yii2/project"
    }
  }
}
```



Configuration

The MCP server automatically detects your Yii2 project configuration:

Database Configuration

The server looks for database configuration in this order:

1. `.env` file (for environment variables like `DB_HOST`, `DB_DATABASE`, etc.)
2. `config/db.php` file (falls back to default MySQL settings)

Project Detection

- Must be run from a directory containing a `yii` console script
- Automatically scans `modules/`, `models/`, `config/` directories
- Supports standard Yii2 directory structure

Usage Examples

Once configured, you can use these tools through Claude Code:



```
# Database Analysis
Use db_list_tables tool
Use db_describe_table with table_name: "user"
Use db_table_relationships with table_name: "user"

# Database Optimization (NEW)
Use db_analyze_table with table_name: "user"
Use db_get_index_suggestions with table_name: "user"
Use db_profile_query with query: "SELECT * FROM user WHERE status = 'active'"
Use db_monitor_connections with alert_threshold: 50

# Yii2 Console Commands
Use yii_list_commands
Use yii_list_migrations

# Project Structure
Use project_structure
Use yii_list_models with module: "fastnet"

# Advanced Module Analysis
Use module_detailed_analysis with module_name: "fastnet"
Use module_detailed_analysis with module_name: "billing"

# Asset Bundle Analysis
Use asset_list_bundles
Use asset_list_bundles with module_filter: "rbac"
Use asset_analyze_dependencies
Use asset_analyze_dependencies with asset_name: "AppAsset"

# Widget Discovery
Use widget_list
Use widget_list with module_filter: "fastnet"

# Migration Management
Use migration_create with table_name: "user_profiles"
Use migration_diff to generate from model changes
Use migration_rollback with steps: 1

# Code Generation
Use generate_crud with table_name: "posts"
Use generate_model with table_name: "categories"
Use generate_api with model_class: "User"

# Cache Operations
Use cache_clear with component: "db"
Use cache_inspect
Use cache_performance
```

```
# RBAC Analysis
Use rbac_permission_tree
Use rbac_user_access with user_id: 1
Use rbac_role_analysis with role: "admin"

# Queue Management
Use queue_job_status
Use queue_retry_failed with job_id: 123
Use queue_worker_status

# Logging & Debugging
Use log_error_analysis with hours: 24
Use log_search with pattern: "database error"
Use log_tail with lines: 50

# Testing
Use test_run with suite: "unit"
Use test_coverage_report

# Performance Analysis
Use performance_profile with duration: "5m"
Use asset_performance_analysis
```

Module Analysis Features

Comprehensive Module Understanding:

- Controllers, models, views structure
- Asset bundles and their dependencies
- Widgets and custom components
- Queue jobs and service classes
- Traits and utility classes
- Bootstrap configuration

Asset Bundle Intelligence:

- CSS and JavaScript file mapping
- Dependency tree analysis
- Circular dependency detection
- Optimal registration order
- Source path resolution

Widget Discovery:

- Widget-to-asset-bundle relationships
- Module-specific widget organization
- Dependency mapping

Safety Features

- **Read-Only Database Access:** Only SELECT queries allowed
- **Command Safety:** Dangerous commands (like `migrate/fresh`) are blocked
- **Configuration Security:** Sensitive config values are not exposed
- **Timeout Protection:** Commands have 30-second timeout limits
- **Interactive Command Detection:** Commands requiring user input are flagged

Requirements

- Node.js 18+
- Yii2 project with console access
- MySQL database
- PHP CLI available in PATH

Troubleshooting

Common Issues

1. "Yii script not found"

- Ensure you're running from the Yii2 project root directory
- Check that the `yii` console script exists and is executable

2. Database connection errors

- Verify your `.env` file has correct database credentials
- Ensure MySQL is running and accessible
- Check that the database specified in config exists

3. Permission errors

- Ensure the server has read access to your project files
- Check that PHP CLI is available and can execute the Yii script

Debug Mode

For development and debugging, you can run the server directly:

```
npm run dev
```



This will watch for file changes and rebuild automatically.

Development

The server is built with TypeScript and consists of:

- `src/index.ts` - Main MCP server setup and tool routing
- `src/database.ts` - Database connection and schema inspection
- `src/yii2.ts` - Yii2 console command execution and project analysis
- `src/config.ts` - Configuration management and project detection

To contribute:

1. Fork the repository
2. Make your changes
3. Run `npm run build` to compile
4. Test with your Yii2 project
5. Submit a pull request

License

Releases

No releases published

Packages

No packages published

Contributors 1



claude Claude

Languages



- TypeScript 68.6%
- JavaScript 29.1%
- Shell 2.3%