Code Review Prompts MCP Server

A Model Context Protocol (MCP) server that provides code review prompts for various programming platforms and detail levels. Compatible with VSCode Cline, Google Antigravity, and other MCP clients.

Features

- **Platform-specific prompts**: Android, TypeScript, Java, Python, React, Kotlin
- **Multiple detail levels**: Short (critical only), Medium (comprehensive), Detailed (in-depth)
- **MCP compatible**: Works with any MCP-compatible client
- **Easy to extend**: Add new platforms and prompts easily

Installation

1. Install dependencies

```
cd mcp-server
npm install
```

2. Build the server

```
npm run build
```

3. Configure your MCP client

For VSCode Cline

Add to your Cline MCP settings (`~/.config/Code/User/globalStorage/rooveterinaryinc.roo-cline/settings/cline_mcp_settings.json`):

```
{
  "mcpServers": {
    "code-review-prompts": {
      "command": "node",
      "args": ["/absolute/path/to/mcp-prompt/mcp-server/dist/index.js"]
    }
  }
}
```

For Claude Desktop

Add to `~/Library/Application Support/Claude/claude_desktop_config.json` (macOS):

```
{
  "mcpServers": {
    "code-review-prompts": {
      "command": "node",
      "args": ["/absolute/path/to/mcp-prompt/mcp-server/dist/index.js"]
    }
  }
}
```

Usage

Available Tools

The server provides two tools:

1. `get_code_review_prompt`

Retrieve a specific code review prompt.

Parameters:

- `platform` (required): One of: android, typescript, java, python, react, kotlin
- `detailLevel` (required): One of: short, medium, detailed

Example:

```
// Request a medium-detail Android review prompt
{
  "platform": "android",
  "detailLevel": "medium"
}
```

2. `list_available_prompts`

List all available prompts.

Example response:

```
# Available Code Review Prompts

- **android** (short)
- **android** (medium)
- **android** (detailed)
- **typescript** (short)
- **typescript** (medium)
- **java** (short)
- **java** (medium)
- **python** (short)
```

```
Total: 8 prompts
```

Using with Cline

Once configured, you can use natural language in Cline:

```
"Get me the Android medium detail code review prompt"
"List all available code review prompts"
"Use the TypeScript short review prompt for this PR"
```

Cline will automatically call the appropriate MCP tool.

Prompt Detail Levels

Short

- Focuses on critical issues only
- Brief, actionable feedback
- Best for quick reviews or experienced teams

Medium

- Comprehensive coverage of common issues
- Includes performance, architecture, and best practices
- Suitable for most code reviews

Detailed

- In-depth analysis with complete workflow
- Includes autonomous execution steps
- Best for complex PRs or learning scenarios

Supported Platforms

Platform	Short	Medium	Detailed
Android	✅	✅	✅
TypeScript	✅	✅	⏳
Java	✅	✅	⏳
Python	✅	⏳	⏳
React	⏳	⏳	⏳
Kotlin	⏳	⏳	⏳

✅ = Available, ⏳ = Coming soon

Adding New Prompts

1. Create a new markdown file in `prompts/` directory:

```
prompts/
  ├── platform-short.md
  ├── platform-medium.md
  └── platform-detailed.md
```

2. Follow the naming convention: `{platform}-{detailLevel}.md`

3. Rebuild the server:

```
npm run build
```

4. Restart your MCP client

Project Structure

```
mcp-server/
├── src/
│   └── index.ts          # Main MCP server implementation
├── prompts/              # Prompt templates
│   ├── android-short.md
│   ├── android-medium.md
│   ├── android-detailed.md
│   ├── typescript-short.md
│   ├── typescript-medium.md
│   ├── java-short.md
│   ├── java-medium.md
│   └── python-short.md
├── dist/                 # Compiled JavaScript (generated)
├── package.json
├── tsconfig.json
└── README.md
```

Development

Running in development mode

```
npm run dev
```

This starts TypeScript in watch mode, automatically recompiling on changes.

Testing the server

You can test the server using the MCP Inspector:

```
npx @modelcontextprotocol/inspector node dist/index.js
```

Troubleshooting

Server not appearing in Cline

1. Check that the path in your MCP config is absolute
2. Ensure the server is built: `npm run build`
3. Restart VSCode/Cline
4. Check Cline's MCP logs for errors

"Prompt not found" error

- Verify the prompt file exists in `prompts/` directory
- Check the filename follows the pattern: `{platform}-{detailLevel}.md`
- Rebuild the server after adding new prompts

Permission denied

Ensure the compiled `dist/index.js` has execute permissions:

```
chmod +x dist/index.js
```

Contributing

To add support for a new platform:

1. Create prompt files for all detail levels
2. Add the platform to the `PLATFORMS` constant in `src/index.ts`
3. Update the README documentation
4. Test with MCP Inspector
5. Submit a pull request

License

MIT

Related Projects

- MCP SDK - Model Context Protocol SDK
- Cline - VSCode AI coding assistant
- Original project: mcp-prompt

Support

For issues and questions:

- Open an issue on GitHub
- Check existing prompts in the `prompts/` directory
- Refer to MCP documentation for client configuration