Automated refactoring

Tom Rochette <tom.rochette@coreteks.org>

December 21, 2025 — 77e1b28a

0.1 Context

0.2 Learned in this study

0.3 Things to explore

1 Overview

- Component extraction (find all dependencies and attempt to create a self-contained library)
- Convert function-based code into class-based code (static calls)
  - Copy all functions in a file into a new class where each function is static
  - Find all calls to the initial functions and replace them with calls to the class
  - Verify that all entry points load the autoloader (how to check that? search for the autoload.php string and require/include calls?)
- Template/logic separation
- Extraction of functions into a separate file
- Extraction and replacement of inline style
- Extraction of string resources
- Extraction of inline JS
- Extraction of inline CSS
- Conversion of raw view logic to template logic
- Removal of inline PHP code in js code
- Convert raw SQL into builder queries
- Move queries in controller into a service/repository
- Convert <? to <?php
- Convert echo calls to string concat + return
- Update outdated phpdoc
- Improve typing (parameters and return)

2 Difficulties of code refactoring

- Lot of code is SQL queries

3 Refactoring to MVC

4 See also

- Automated refactoring

5 References