



Last indexed: 27 August 2025 (c12f7a)

PadelFlow Overview

Core Application Architecture

Server Setup and Configuration

User Roles and Authentication

Tournament Management
Features

Real-time Features

Database Layer

SQLite Database Management

Database Extensions

Qt Framework Components

Image Format Support

Development Environment

IDE Configuration

Debugging Setup

Project Configuration

Development Environment

Relevant source files

Purpose and Scope

This document provides a comprehensive guide for setting up and configuring the development environment for PadelFlow. It covers the Visual Studio Code workspace configuration, debugging setup, and development workflow tools that enable efficient development and testing of the tournament management platform.

For information about the core application structure and server setup, see [Core Application Architecture](#). For database development tools, see [Database Layer](#).

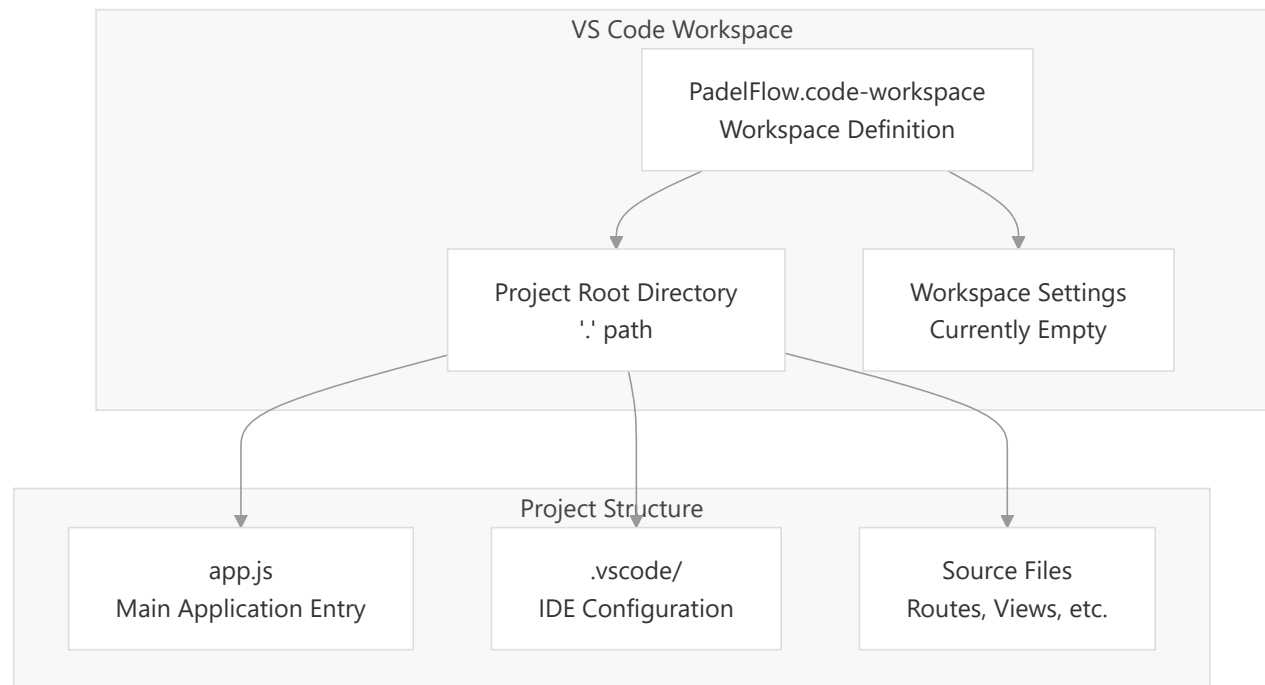
IDE Configuration

PadelFlow uses Visual Studio Code as the primary integrated development environment. The workspace is configured through a dedicated workspace file that defines the project structure and settings.

VS Code Workspace Setup

The project uses a VS Code workspace configuration to organize the development environment. The workspace file defines the project root and provides a centralized configuration for all development activities.

Workspace Configuration Diagram



The workspace configuration is minimal and extensible, allowing for future customization of development settings, extensions, and project-specific configurations.

Sources: PadelFlow.code-workspace:1-8

Workspace Structure

Component	Configuration	Purpose
Folders	Single root folder (".")	Defines project root directory
Settings	Empty object ({})	Reserved for future workspace-specific settings
Path Reference	Current directory	Relative path to project root

The workspace file follows VS Code's standard workspace format, enabling features like multi-root workspaces, workspace-specific settings, and extension recommendations.

Sources: PadelFlow.code-workspace:2-7

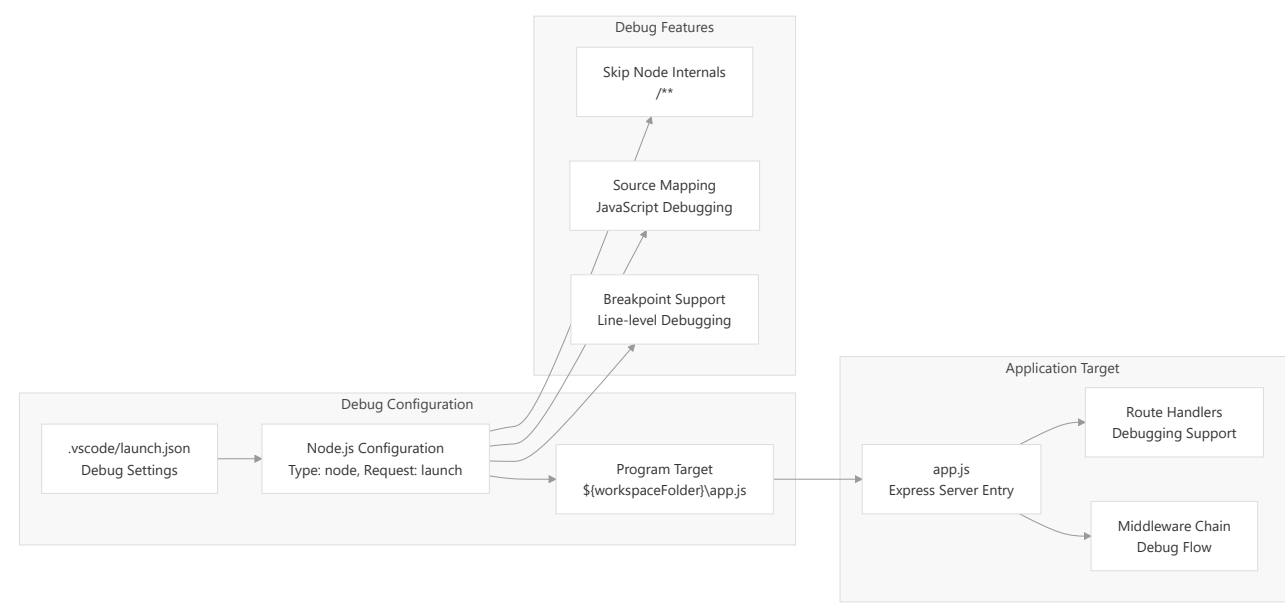
Debugging Configuration

PadelFlow includes a comprehensive debugging setup through VS Code's launch configuration system. The debugging configuration enables developers to run and debug the Node.js application directly from the IDE.

Node.js Debug Setup

The debugging configuration targets the main application entry point and provides source-level debugging capabilities for the Express.js server.

Debug Configuration Diagram



The debug configuration provides direct access to the application's execution flow, enabling developers to inspect variables, step through code, and analyze the tournament management logic.

Sources: .vscode/launch.json:1-17

Debug Configuration Details

The launch configuration includes several key components that optimize the debugging experience:

Setting	Value	Purpose
Version	"0.2.0"	VS Code launch configuration schema version
Type	"node"	Specifies Node.js debugging engine
Request	"launch"	Launch mode (vs. attach mode)

Setting	Value	Purpose
Name	"Launch Program"	Human-readable configuration name
Program	"\${workspaceFolder}\\app.js"	Target application entry point
Skip Files	["<node_internals>/**"]	Excludes Node.js internal modules from debugging

The configuration uses workspace-relative paths, ensuring portability across different development environments and team members' local setups.

Sources: .vscode/launch.json:6-16

Debugging Workflow

The debugging setup enables a streamlined development workflow:

1. **Launch Configuration:** Use `F5` or the debug panel to start debugging
2. **Breakpoint Management:** Set breakpoints in any JavaScript file
3. **Variable Inspection:** Examine application state during execution
4. **Call Stack Navigation:** Trace execution flow through the application layers
5. **Internal Module Filtering:** Focus on application code by skipping Node.js internals

This configuration particularly supports debugging of:

- Express route handlers
- Tournament management logic
- Real-time event processing
- Database operations

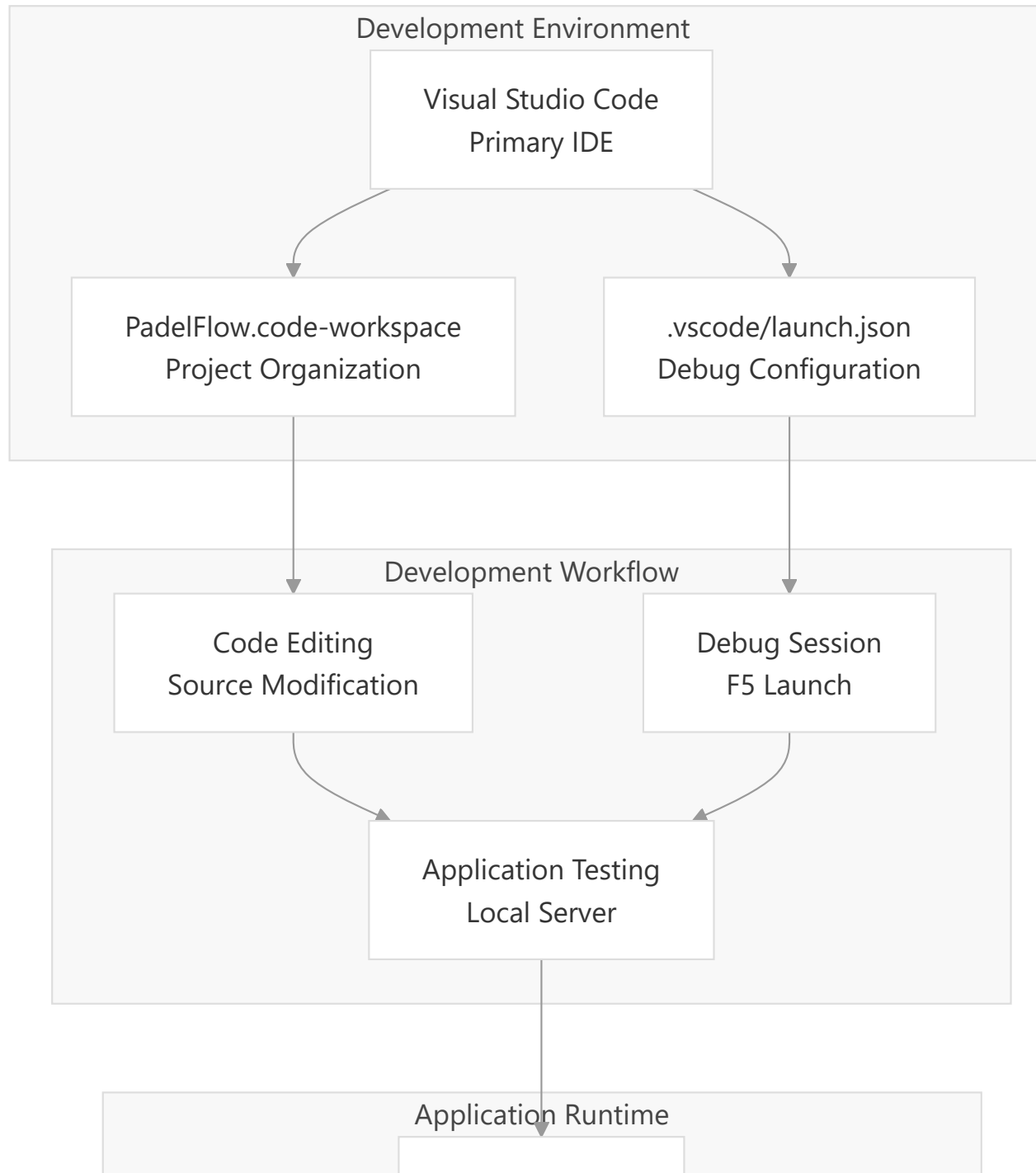
- Authentication middleware

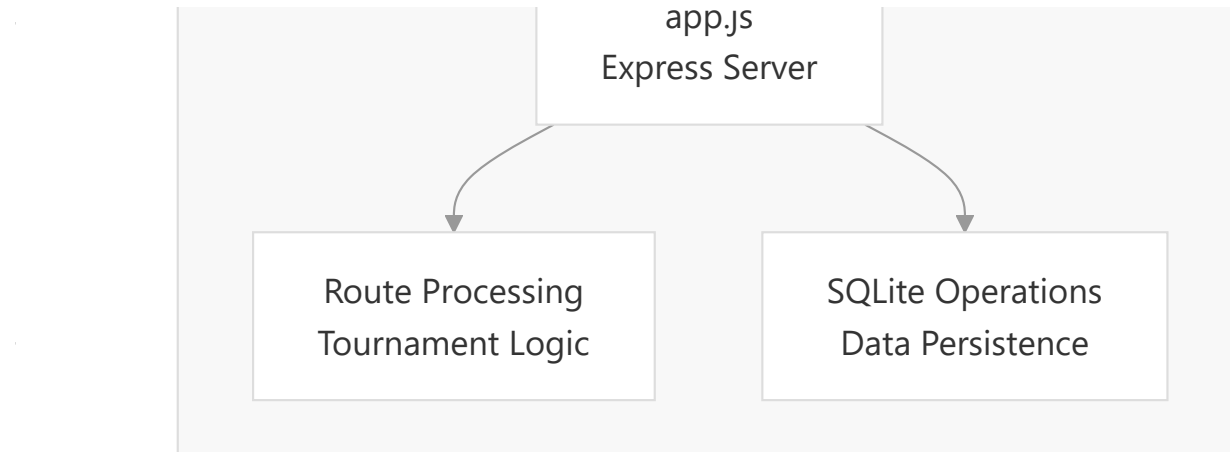
Sources: .vscode/launch.json:11-14

Development Workflow Integration

The development environment configuration supports efficient iteration and testing cycles for PadelFlow development. The combination of workspace organization and debugging capabilities provides a comprehensive development platform.

Development Tool Integration





• **Git Integration:** Source control management within the IDE

- **Terminal Integration:** Access to command-line tools and npm scripts
- **Extension Ecosystem:** Support for Node.js, Express.js, and web development extensions

The minimal workspace configuration allows teams to add project-specific settings and extension recommendations as the project evolves.

Sources: PadelFlow.code-workspace:7