

# Debugging Tools and Techniques

## Introduction

Debugging is essential in Blazor development to identify and resolve issues that impact your application's performance. This guide will walk you through using Visual Studio Code's debugging tools—like breakpoints, the Watch window, and step functions—to find and fix errors effectively in your Blazor projects.

## Steps to Debug

### 1) Set a Breakpoint

- Identify the line of code where you want to pause execution.
- Click to the left of the line number to add a red dot (breakpoint) that will halt code execution at this point.

### 2) Start the Debugger

- Open the Run and Debug tab in the Visual Studio Code sidebar.
- Click Start Debugging to compile and run your Blazor app in debug mode.
- Code execution will stop at breakpoints, allowing you to inspect the application's state.

### 3) Add Variables to the Watch Window

- Right-click on any variable and select Add to Watch to track its value.
- This enables you to monitor specific variables and expressions as you debug.

### 4) Step Through Code

- Use Step Over to advance line by line within the current function.
- Select Step Into to go deeper into functions and inspect their internals.
- Use Step Out to leave a function and return to the previous execution level.

## Conclusion

Mastering these debugging tools in Visual Studio Code will allow you to pinpoint and fix issues efficiently in your Blazor applications. By using breakpoints, monitoring variable values, and

stepping through code, you'll gain valuable insights into how your application operates and can quickly resolve errors for a smoother, more reliable app experience.