# Debugging

Before starting to work with particular frameworks that are supported by Visual Studio Code, I want to cover one more topic – Debugging.

Visual Studio Code supports debugging for Node.js and Mono runtimes. Because we still didn’t discuss how to work with Node.js or ASP.NET in Code I am going to show debugging features using our experience in integration of Code and Unity. Because Unity uses Mono, we can try to use debugger there. I would recommend to use [Unity plugin](https://github.com/dotBunny/VSCode/) from @reapazor in order to integrate Unity and Code, but you can setup everything yourself from scratch (see my previous posts about Code and Unity).

In order to activate debugging features we need to add a **launch.json** file to our project. If you decide to make everything from scratch you can open Debug view in Code and click Settings button in order to add **launch.json**:



We cannot use this **launch.json** because it is adapted for Node.js but you can review the most common element there.

Just remove all content from **launch.json** and use this one:

*{*

*"version":"0.1.0",*

*"configurations":[*

*{*

*"name":"Unity",*

*"type":"mono",*

*"address":"localhost",*

*"port":56060*

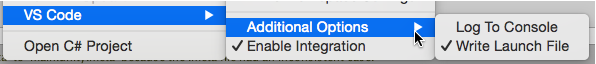
*}*

*]*

*}*

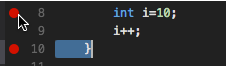
The most important attributes there are **type** that should be set to **mono**, **address** that should be set to **localhost,** and **port** that you are going to use for mono debugger.

If you decide to use [Unity plugin](https://github.com/dotBunny/VSCode/), you need to make sure that Write Launch File is enabled:



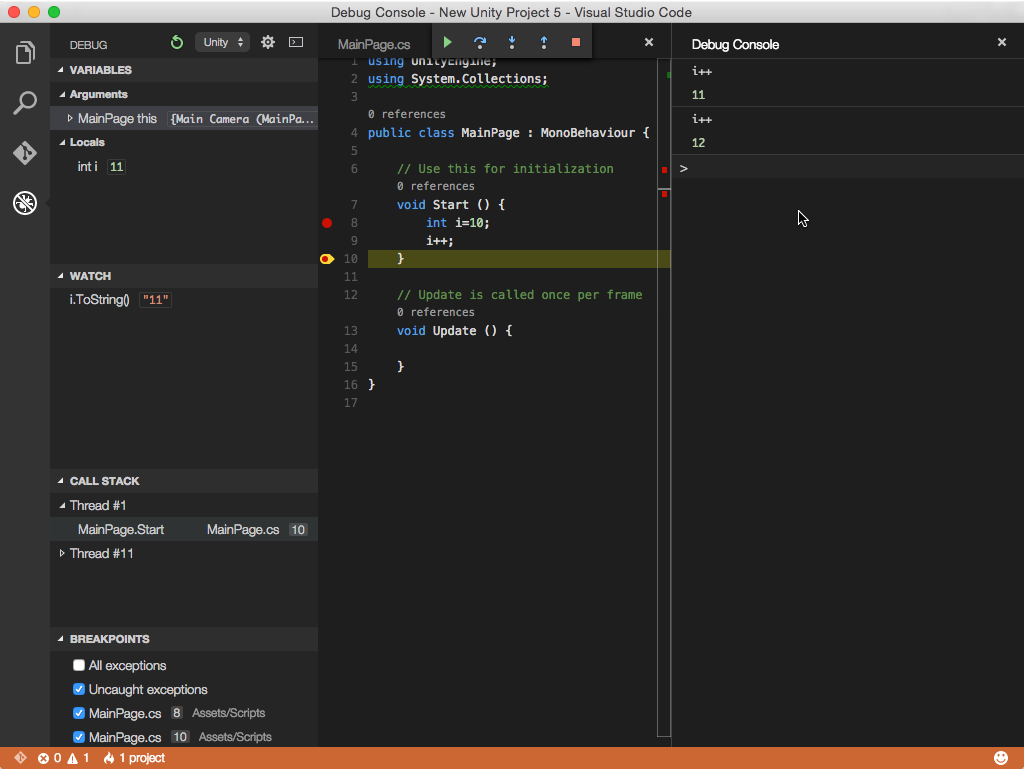
Once you start your game in Unity player, the plugin will create **launch.json** file for you.

In order to test some features in Code you can simply create a C# script, type some code and associate the script with any Unity object. For example, you can drag the script to Camera object. If you have some code it’s time to set breakpoint. You can do it by clicking empty space before line numbers in Editor:



To start debugging you need to open Debug view and click Start button. Right after that you can open Unity and start your game in Unity player.

If everything is OK you will see a window similar to the one below:



There are the following elements:

* **Breakpoints** – you can see and manage all available breakpoints in the project. This window contains information like name of a file, line, subfolder. In order to manage breakpoints you can use context menu;
* **Call Stack**;
* **Watch** – using this window you can evaluate expressions with variables that are accessible in the context;
* **Variables** – this window shows local variables and their values. Additionally, you can check parameters and “**this”** object;
* **Debug actions panel** – using this panel you can pause or stop the process. Additionally, you can move through your code step by step;
* **Debug console** – one more way to evaluate expressions;

So, Debug view has all needed attributes of professional debuggers and I hope that you will enjoy it. In the next posts we will discuss Node.js and ASP.NET, and I am going to talk about the debugger a little bit more.