

Cannot debug in VSCode by attaching to Chrome

Asked 2 years, 1 month ago Active 1 month ago Viewed 17k times

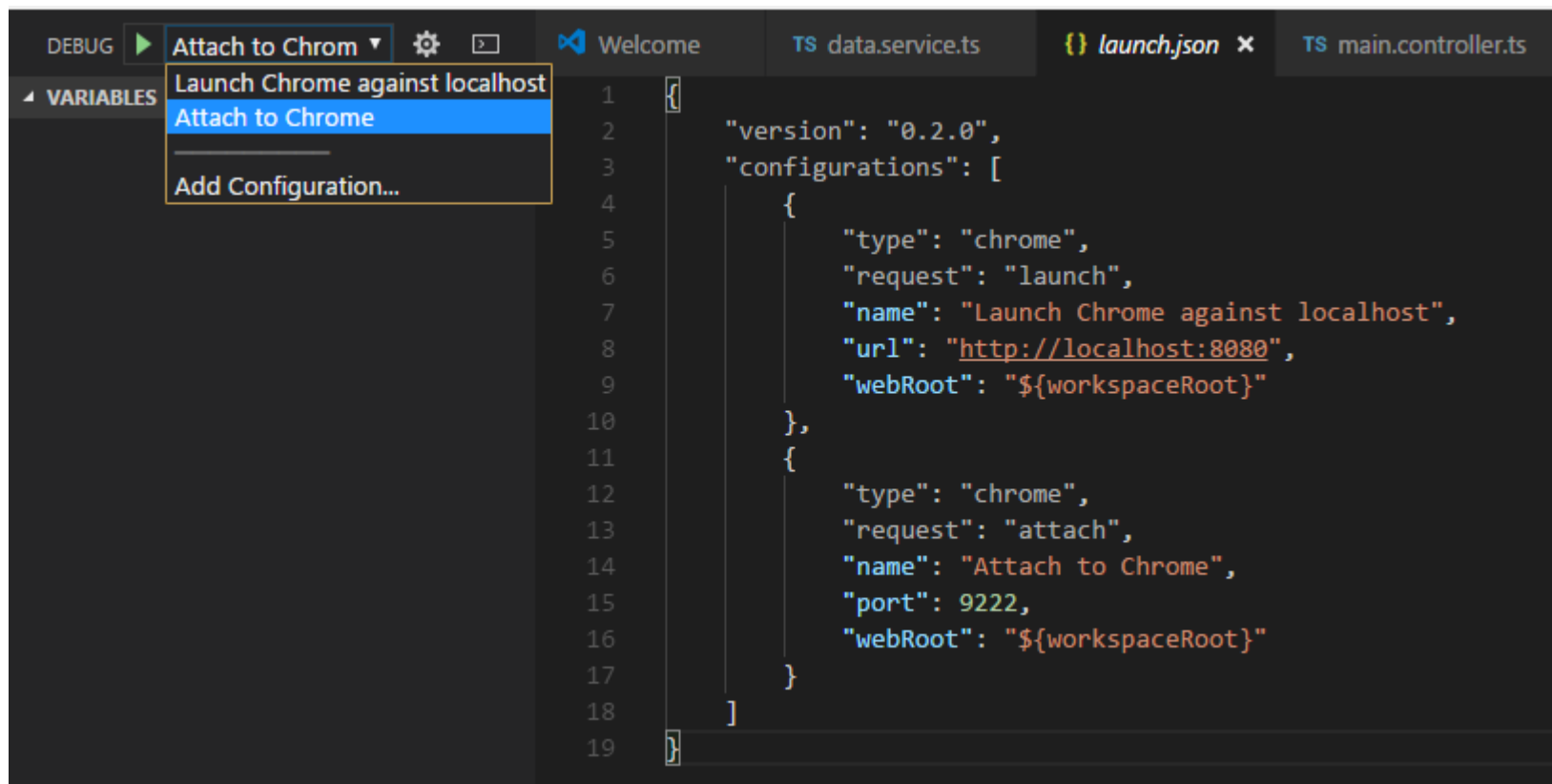
I have default config in launch.json, the site runs on port 8080

20



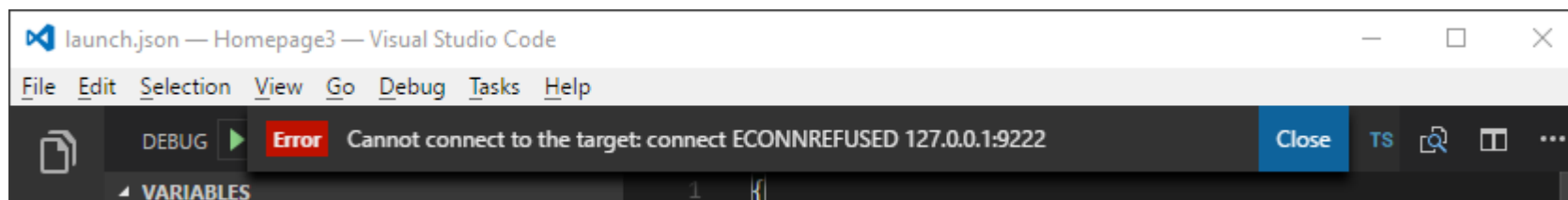
4

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "chrome",
      "request": "launch",
      "name": "Launch Chrome against localhost",
      "url": "http://localhost:8080",
      "webRoot": "${workspaceRoot}"
    },
    {
      "type": "chrome",
      "request": "attach",
      "name": "Attach to Chrome",
      "port": 9222,
      "webRoot": "${workspaceRoot}"
    }
  ]
}
```



However when I click on Debug button, I get this error:

Cannot connect to the target: connect ECONNREFUSED 127.0.0.1:9222



Question1: why VSCode assigns port 9222 when creating this json

What is so special about this port that MS decided to put it in this launch.json?

Question2: What do I need to change to make things work.

The Launch debug always launches a new window. I am asking **specifically about Attach debug option**, so that it will open in a new tab instead.

Thanks.

visual-studio-code

asked Aug 9 '17 at 13:47



4 Answers



16



1. You need to install Debugger for Chrome extension for this to work. Open extensions in VS Code and search for Debugger for Chrome
2. You need to run a web server on the URL specified in the first configuration (default to <http://localhost:8080>). I use `serve` npm package that I installed globally. From my app folder I run `serve -p 8080`
3. Select Launch Chrome against localhost option. It will launch the browser and you can set breakpoints in your code and the debugging should work.

Regarding the second configuration (Attach to Chrome). There's nothing special about the port. In order to attach to Chrome you need to run Chrome with remote debugging enabled on port specified in the config. For example `chrome.exe --remote-debugging-port=9222`. I personally never use this options. Just follow the three steps above and you should be fine.

answered Aug 9 '17 at 16:40



19



When using the configuration `url`, vscode will search for a tab with the EXACT url and attach to it if found.

Use the configuration `urlFilter` which can have wildcards like `*`, to attach the debugger to any sub route in your url.

e.g. `"urlFilter": "http://localhost:4200/*"`

The complete exacts steps to take:

1. configure your launch.json file to looks something like this:

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "chrome",
      "request": "attach",
      "name": "Attach to Chrome",
      "port": 9222,
      "urlFilter": "http://localhost:4200/*",
      "webRoot": "${workspaceFolder}"
    }
  ]
}
```

2. Close all opened chrome instances (make sure that all of them are killed using task manager in windows).
3. Lunch chrome with the following parameter: --remote-debugging-port=9222

make sure that the port in this parameter is the same as the one configured in 'port' property of the attache to chrome configuration in the lunch.json file (like the example above)

4. Navigate to your site. In this example <http://localhost:4200>
5. Run 'Start debugging' in vscode.

edited Nov 29 '18 at 9:57

answered Nov 15 '18 at 12:40



benshabatnoam
2,821 12 27

1 Adding * in the URL fixed this for me. Thanks for the pointer. – [Mickers](#) Nov 19 '18 at 21:10

1 This is the answer I have waited for a long time. Thanks. To me worked with url either. – [Alexandre N.](#) Nov 24 '18 at 14:02 ✎

2 Using urlFilter instead of url worked for me. Here is the docs on this topic [github.com/Microsoft/...](https://github.com/Microsoft/vscode/blob/master/src/debugger/debuggerHost.ts) – [amankkg](#) Nov 29 '18 at 7:11 ✎

1 Thanks for "urlFilter"! – [Marek Javůrek](#) Jan 18 at 9:51



I came across this question when looking for help using the "Attach to Chrome" configuration in VSCode. While the accepted answer did

10

give me a few hints, I did have to do some more digging. Here are the steps that worked for me in case anyone else finds them useful:

1. Install the Debugger for Chrome extension in VSCode
2. Serve your files with a web server of your choice
3. [Launch Chrome with remote debugging enabled](#)
4. In this new Chrome window navigate to the url that your web server is hosting (<http://localhost:8080> for example).
5. In VSCode, add a configuration to your `launch.json` file that looks like this:

```
"configurations": [  
  {  
    "type": "chrome",  
    "request": "attach",  
    "port": 9222,  
    "name": "Attach Chrome",  
    "url": "http://localhost:8080/",  
    "webRoot": "${workspaceFolder}"  
  }  
]
```

6. Press the play button in VSCode with the 'Attach to Chrome' option selected from the drop down.

The key thing needed in the configuration file is the `url` field. This needs to be the URL where your files are being hosted and this URL needs to be currently open in the Chrome window that you just launched with remote debugging enabled. If you enter everything else right except this field, VSCode will give you an error message that says which urls are available. Something like `Cannot connect to runtime process, timeout after 10000 ms - (reason: Can't find a valid target that matches: localhost:8080/. Available pages: ["http://localhost:8080",...`

For the sake of completeness, here's how you launch Chrome with remote debugging enabled ([from the Debugger for Chrome README](#)):

Windows:

- Right click the Chrome shortcut, and select properties
- In the "target" field, append `--remote-debugging-port=9222`
- Or in a command prompt, execute `<path to chrome>/chrome.exe --remote-debugging-port=9222`

MacOS:

- In a terminal, execute `/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome --remote-debugging-port=9222`

Linux:

- In a terminal, launch `google-chrome --remote-debugging-port=9222`

answered Jul 18 '18 at 1:10



[quicklikerabbit](#)

844 2 14 25

-
- 1 Best answer. I should also add that the "url" field should be set to whatever your page address is when open. The debugger tries to find the **exact** match in your browser tab. I was working with vue.js and the default url "http://localhost:8080/" does not work, it should be "http://localhost:8080/#/" . – [Edu](#) Jun 22 at 17:55
-



0



Peek into Task Manager. You may have Chrome instances hanging there. Only after killing them will you be able to run the remote and successfully start the debugger. Worked for me.

answered Aug 5 at 16:36



[Duarte Lucas](#)

1