

 Microsoft / **vscode**

# How to debug Meteor app? #256

 **Closed**

spacenergy opened this issue on 19 Nov 2015 · 36 comments



spacenergy commented on 19 Nov 2015

*No description provided.* joaomoreno assigned **isidorn** on 19 Nov 2015 isidorn added the needs more info label on 23 Nov 2015

isidorn commented on 23 Nov 2015

Contributor

Have you tried to debug it like any other [node app](#)?  
If yes what issues have you run into?

 isidorn added the debug label on 24 Nov 2015

isidorn commented on 1 Dec 2015

Contributor

Closing until more info is provided

## Assignees

 weinand

## Labels

debug

feature-request

## Projects

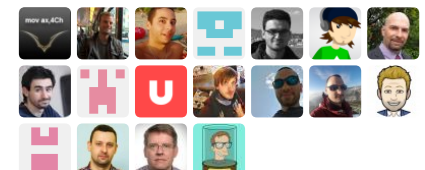
None yet


## Milestone

No milestone

## Notifications

## 18 participants



 **isidorn** closed this on 1 Dec 2015



**george2giga** commented on 22 Jan 2016

Hi, trying to do the same, wondering what would be js file to use as entry point for the debugger (for node is app.js). any help would be much appreciated !



**isidorn** commented on 22 Jan 2016

Contributor

Can you provide a minimum sample workspace for which we can try this out?



**george2giga** commented on 22 Jan 2016

Hi **@isidorn** , thanks for the quick reply.  
Any meteor.js project should be fine, anyway I've uploaded the very basic sample provided by meteor when you create a new project <https://github.com/george2giga/basic-sample>, you can try this workspace.  
Having the VS Code debugging working would be awesome.  
Thanks

 **isidorn** reopened this on 22 Jan 2016

  **isidorn** added **feature-request** and removed **needs more info** labels on 22 Jan 2016

  **isidorn** added this to the **Backlog** milestone on 22 Jan 2016



isidorn commented on 22 Jan 2016

Contributor

Ok what seems to be somewhat working for me is the following:

- run your app by `meteor debug`
- have an attach request in your `launch.json` that will attach to the debug port that meteor gives you

```
{  
  "name": "Attach to meteor",  
  "type": "node",  
  "request": "attach",  
  "port": 5858  
}
```

@george2giga can you please try this?



7



george2giga commented on 23 Jan 2016

Thanks @isidorn , works a charm :)



isidorn commented on 23 Jan 2016

Contributor

Glad it worked. Closing this



isidorn closed this on 23 Jan 2016



isidorn added the verified label on 26 Feb 2016



**winuxue** commented on 11 Jul 2016

Hi. I tried what you said, and vscode started debugging! but doesn't stop in my breakpoints. is there something that I'm missing?



**isidorn** commented on 19 Jul 2016

Contributor

@**winuxue** please open a new issue with reproducible steps (possibly a small sample) so we could investigate this further.



**stefanofiorentino** commented on 11 Nov 2016

@**isidorn** check if port is really 5858, in my case was 5959  
Debugger listening on port 5959



**jdnichollsc** commented on 14 Dec 2016

Any example using Meteor **settings**? For example with the following command:

```
meteor --settings settings.json --port 8890
```

Thanks in advance, Nicholls



**camstuart** commented on 26 Mar 2017

I can attach to the running process and set a breakpoint. But as soon as I change any server side code, meteor restarts (normal for meteor) and the debug process disconnects. then execution goes past my breakpoint because at that time, the debugger is not listening. I suspect this is what was happening @winuxue has anybody got this working correctly? a `launch.json` would be fantastic



Exilz commented on 28 Mar 2017

@camstuart I'm in the exact same situation here, I came to the same conclusion. It's not far from being an awesome workflow !



isidorn commented on 11 Apr 2017

Contributor

Ok seems to be no longer working looking at the comments -> reopening and assigning to april to investigate



isidorn reopened this on 11 Apr 2017



isidorn modified the milestones: **April 2017**, **Backlog** on 11 Apr 2017




isidorn removed the verified label on 11 Apr 2017



isidorn commented on 13 Apr 2017

Contributor

**@camstuart @Exilz** can you please try adding "restart": true to your launch configuration - which will notify the debug adapter to automatically reconnect to meteor when it restarts  
If that does not work I can reopen this issue

 **isidorn** closed this on 13 Apr 2017



**Exilz** commented on 14 Apr 2017

Thanks **@isidorn** , "restart": true prevents the debugger from disconnecting.  
However, I can't get it to stop on my breakpoints.

If an exception is caught, the debugger will display it properly, but it's still ignoring breakpoints.  
Not sure wether this is a bug or if I'm just missing something in my configuration

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "node",
      "request": "attach",
      "name": "Attach to meteor",
      "port": 5959,
      "restart": true,
      "sourceMaps": true
    }
  ]
}
```



**valdirSalgueiro** commented on 24 Apr 2017

im having the exact same bug **@Exilz**



**isidorn** commented on 24 Apr 2017

Contributor

Assigning to **@weinand** as I am out of ideas  
fyi **@roblourens**

🚫 **isidorn** removed this from the **April 2017** milestone on 24 Apr 2017

👤 **isidorn** assigned **weinand** and unassigned **isidorn** on 24 Apr 2017

🔁 **isidorn** reopened this on 24 Apr 2017



**jesperordrup** commented on 21 May 2017

It works for me. I found that even though meteor tells me to use 5858 its really 5959.

I run

meteor debug

and meteor tells

To debug the server process using a graphical debugging interface,  
visit this URL in your web browser:

<http://localhost:8080/?port=5858>

but this launch config works

```
{  
  "type": "node",
```

```
"request": "attach",
"name": "Attach to meteor 5959",
"port": 5959,
"restart": true,
"sourceMaps": true
}
```



**BaobabCoder** commented on 13 Jun 2017

You can use `debugger;` to stop in your code or set breakpoint in the compiled js application.



**MartinBucko** commented on 30 Jun 2017 • edited ▼

Good news!

Meteor 1.6-beta.4 comes with Node v8.1.2. So it mean `"protocol": "inspector"` now works with VScode.

Update Meteor:

```
meteor update --release 1.6-beta.4
```

In launch.json:

```
...
{
    "type": "node",
    "request": "launch",
    "name": "Meteor Server",
    "cwd": "${workspaceRoot}/",
    "runtimeExecutable": "${workspaceRoot}/.meteor/local/dev_bundle/bin/npm",
    "restart": true,
    "timeout": 100000,
    "stopOnEntry": false,
    "runtimeArgs": [
```



```
    "run",
    "dev-debug"
  ],
  "sourceMaps": true,
  "protocol": "inspector",
  "port": 5959
}
...
```

In package.json:

```
...
"scripts": {
  "start": "meteor run",
  "dev-debug": "meteor debug --debug-port 5959"
}
...
```

And now it supports VScode breakpoints! 🍌



1



2



**isidorn** commented on 30 Jun 2017

Contributor

@**MartinBucko** thanks for letting us know, closing this issue then



1



**isidorn** closed this on 30 Jun 2017



**MartinBucko** referenced this issue in **meteor/meteor** on 30 Jun 2017

## Meteor node debugging does not map origin source files to files on disk. Required for VS Code. #8590

 Closed

**grimmer0125** added a commit to grimmer0125/newCARTAMeteorApp that referenced this issue on 21 Jul 2017



1. update to Meteor@1.6-beta.9 for using debugger of Visual Studio Code. ...

158d1fa



**chmac** commented on 8 Aug 2017

Just a little FYI for anyone finding this issue via Google as I have...

Meteor runs 2 node processes, one that "builds" your app and then one that gets killed and restarted on every "hot code reload". Let's call them parent and child. The parent process is the one that listens on port 3000, and it proxies connections to the child process. That's how hot code reload works. I **believe** that it's doing the same from 5858 to the child process which runs its debugger on 5959. However, it's been **extremely** flakey in all my tests.

I find that I get better results running meteor with the `NODE_OPTIONS` environment variable so I can start it like so:

```
NODE_OPTIONS='--debug' meteor run
```

This results in the child process being run like `node --debug` which causes its debugger to listen on 5858. Then I can connect to that with vscode. Thanks to this thread I've discovered the `restart: true` flag which means vscode reconnects to this child process (it gets killed on every code reload remember!).

I still can't get breakpoints to work though. Adding `sourceMaps: true` to my `launch.json` doesn't appear to work either. If I add a `debugger;` statement to my code, then vscode opens a "read-only inlined content from source map" version of my original file, rather than showing me the debugger in the original source file on disk. I'd guess that's the root cause of the breakpoints not working, vscode doesn't know how to map the files on disk to the es5 code that the child meteor process actually runs. If we could solve that, I'd guess that breakpoints would work.



**aadamsx** commented on 18 Aug 2017 • edited ▼

Why has this been closed? it still doesn't work for me on 1.5.1 (the current official, non-beta, release).



**ntziolis** commented on 21 Aug 2017 • edited ▼

I'd guess that's the root cause of the breakpoints not working, vscode doesn't know how to map the files on disk to the es5 code that the child meteor process actually runs. If we could solve that, I'd guess that breakpoints would work.

Same here, can someone that already runs Meteor 1.6 confirm that the breakpoint issue is solved as well? Not just the debugger starting but that breakpoints are actually properly hit? Otherwise I would reopen this case.



**aadamsx** commented on 21 Aug 2017 • edited ▼

We need proper, comprehensive, instructions on how to set this up with Meteor -- to make sure some issues are not just because of configuration or process -- and then whatever falls out of that should be fixed.

I've followed and added most-all configuration instructions, yet I still cannot get things working with breakpoints -- heck I can't even tell if its working at all. For now I'll have to stick with using the console/node-inspector for debugging sever side, and not get the full VSCode experiance I believe MS intended.

NOTE: I'm on Meteor 1.5.1, and VSCode 1.15.1



**Miniwe** commented on 21 Aug 2017 • edited ▼

My way:

1: launch.json

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Meteor Attach",
      "type": "node",
      "request": "attach",
      "port": 5858,
      "address": "localhost",

      "processId": "${command:PickProcess}",
      "smartStep": true,
      "trace": false,
      "restart": true,

      "timeout": 20000,
      "sourceMaps": true,
      "outFiles": [
        "${workspaceRoot}/.meteor/local/build/programs/web.browser/app/app.js",
        "${workspaceRoot}/.meteor/local/build/programs/server/app/app.js"
      ]
    }
  ]
}
```

2. Start Meteor
  3. run debug in vcode
  4. select meteor process
  5. open build/programs/server/app/app.js and add breakpoint in this file
  6. when program will reach breackpoint VCode will open server file for read only generated from sourcemap same as original file from app
  7. in this file we can add breakpoints and variables in watch
- for client only debugger or work over chrome dev tools



**MartinBucko** commented on 21 Aug 2017

My previously mentioned config works with meteor 1.6-beta.15 (node v8.2.1, correct version for node inspector) but not with Meteor 1.5 (lower node version, don't support inspector!)



**jesperordrup** commented on 21 Aug 2017 • edited ▼

@**Miniwe** Which versions (Meteor, nodejs, ...) is this for? Thanks



**weinand** commented on 21 Aug 2017

Member

@**Miniwe** Is it really necessary to select the meteor process with the process picker in the launch config from above? If meteor is launched in debug mode, then specifying the port is all you need to attach the debugger. Attaching by processId (or with the picker) is only useful if meteor was not started in debug mode.

The "outfiles" attributes accepts glob patterns, so you can just set it to this:

```
"outFiles": [  
  "${workspaceRoot}/.meteor/**/*.js",  
]
```

This selects all \*.js files under the .meteor directory (on any folder level).



**Miniwe** commented on 22 Aug 2017 • edited ▼

Meteor 1.5.1  
node v7.8.0  
[Screenshot](#)



**roblourens** commented on 22 Aug 2017 • edited ▼

Member

This should work as long as you're using a Meteor version that supports the Inspector protocol. I'm not super familiar with Meteor but from [#256 \(comment\)](#) it sounds like this will only work with Meteor 1.6+. This is because the inspector debug adapter supports the path mapping needed to fix Meteor's weird sourcemap paths, and fixes the default one automatically - <https://github.com/Microsoft/vscode-node-debug2/blob/master/src/nodeDebugAdapter.ts#L26>



**weinand** commented on 22 Aug 2017

Member

We should create a Meteor recipe!  
-> [weinand/vscode-recipes#21](#)



**weinand** referenced this issue in **Microsoft/vscode-recipes** on 22 Aug 2017

## Meteor Recipe #21

Closed



**ntziolis** commented on 22 Aug 2017 • edited ▼

Thank you **@Miniwe**, I wasn't expecting to have to set breakpoints outside of my actual source, I guess I'm spoiled by now. So from this thread and all others this is my current conclusion:

Meteor pre 1.6

Debugging is possible, BUT only by setting breakpoints in the output that Meteor generates. NOT in your actual source files. This is better than no debugging at all, but not being able to fix issues where you find them is not optimal + setup is complex

Post 1.6

Proper debugging with breakpoints in actual source files.

... preying for 1.6 to arrive soon :)



**MartinBucko** commented on 22 Aug 2017 • edited ▼

**@Miniwe** Meteor 1.5.1 have node version 4.8.4 !

Here is example how to get correct node version under meteor:

```
vertest → meteor --version
Meteor 1.5.1
vertest → meteor node --version
v4.8.4
vertest → meteor npm --version
4.6.1
```



1



 **grimmer0125** referenced this issue in **grimmer0125/newCARTAMeteorApp** on 7 Sep 2017

**note, todolist #1**

 7 of 8 tasks complete

 **Open**



**vscodebot** bot locked and limited conversation to collaborators on 18 Nov 2017