



MIEIC – December 2017

# Software Development in Practice

ESOF  
Class 1 - Group F

Diogo Peixoto Pereira – up201504326  
Maria Eduarda Santos Cunha – up201506524  
Pedro Miguel Ferraz Nogueira da Silva – up201505460



# Index

<b>1. Open Source Project .....</b>	<b>3</b>
1.1. Why did we choose this Project?.....	3
1.2. How alive is this project? .....	3
1.3. How important is it? .....	3
1.4. What is it good for?.....	3
1.5. What are the technologies involved? .....	3
<b>2. Issues.....</b>	<b>4</b>
2.1. Issue #39549 .....	4
2.2. Issue #39606 .....	8



## 1. Open Source Project

### 1.1. Why did we choose this Project?

Visual Studio Code was launched on the 29th of April, in 2015 and was developed by Microsoft.

One of the reasons for the class to choose this project is because it was "alive" enough for us to get some feedback from the developers.

Also, it had a tremendous amount of issues to be fixed. That means that there are a lot of options for all groups to choose an issue and fix it.

### 1.2. How alive is this project?

Visual Studio Code has almost daily updates and the community seems to be very active.

Since its creation it's been famously growing in the developing community.

There are currently around 4000 open issues, nearly 5000 contributors and over 27000 commits.

### 1.3. How important is it?

Visual Studio is becoming more and more used and loved among programmers because it has a lot of important and useful tools such as Git, IntelliSense, auto-completion and many more that help the life of many programmers.

### 1.4. What is it good for?

Editing, building and debugging with ease.

It is able to incorporate different languages in the same code editor, with multiple extensions.

IntelliSense<sup>TM</sup> Technology allows for easier coding and auto-completion.

### 1.5. What are the technologies involved?

Visual Studio Code is developed using TypeScript as the programming language. It also uses a well-known library called Electron which allows the creation of a desktop application using web based programming languages such as JavaScript, HTML and CSS for all platforms (Windows, Linux and MacOS).



## 2. Issues

### 2.1. Issue #39549

**Issue:**

<https://github.com/Microsoft/vscode/issues/39549>

**Description:**

When searching for the files to include/exclude, they were separated by a comma, but should be separated by a ';' instead, like what happens in Visual Studio.

The screenshot shows a GitHub issue page for #39549. The title is "search 'files to include/exclude' should allow ';' as a separator". It has a green "New issue" button. Below the title, it says "Closed" and "draeger-charles-wilson opened this issue 3 days ago · 4 comments". The main body of the issue contains the following text:

```
VSCode 1.18.1
OS Windows 7

Search should allow semicolon (';') as a separator in the file inclusion / exclusion specifiers in addition to comma (',').
```

Below the text, there's a timeline of events:

- A user comment from "draeger-charles-wilson" (3 days ago) edited by "vscodebot" (bot).
- An action from "vscodebot" (bot) assigned "roblourens" 3 days ago.
- An action from "vscodebot" (bot) added the "search" label 3 days ago.

On the right side, there are sections for "Assignees" (roblourens), "Labels" (\*out-of-scope, feature-request, search), and "Projects" (None yet).

Fig 1: Issue #39549 description by the issue's submitter

**Requirements:**

In terms of user stories, we can explain our requirements in this way:

As <User> <searching for something in files to include/not in files to exclude>, I want <to separate the different files with ';'> because <the separator in proper VisualStudio is a semicolon>.



Before we solved the issue, VScode would act like this:

queryBuilder.ts - vscode - Code - OSS Dev

File Edit Selection View Go Debug Tasks Help

SEARCH query Aa ⌂ \* files to include searchViewlet.ts;queryBuilder.ts files to exclude

132 results in 2 files

▶ TS searchViewlet.ts src\vs\workben... ▶ TS queryBuilder.ts src\vs\workben...

queryBuilder.ts x html snippets.json package.json searchViewlet.ts

```

123 const searchPaths = this.expandSearchPathPatterns(groups.searchPaths);
124 if (searchPaths && searchPaths.length) {
125   result.searchPaths = searchPaths;
126 }
127
128 const includePattern = patternListToIExpression(exprSegments);
129 if (includePattern) {
130   result.pattern = includePattern;
131 }
132
133 return result;
134
135 /**
136 * Takes the input from the excludePattern as seen in the searchViewlet. Runs the same algorithm as parseExcludePattern,
137 * but the result is a single IExpression that encapsulates all the exclude patterns.
138 */
139 public parseExcludePattern(pattern: string): glob.IExpression | undefined {
140   const result = this.parseSearchPaths(pattern);
141   let excludeExpression = glob.getEmptyExpression();
142   if (result.pattern) {
143     excludeExpression = objects.mixin(excludeExpression, result.pattern);
144   }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Ln 136, Col 8 Tab Size: 4 UTF-8 CRLF TypeScript 2.6.2

Fig 2: Separator of different files to include was comma.

queryBuilder.ts - vscode - Code - OSS Dev

File Edit Selection View Go Debug Tasks Help

SEARCH query Aa ⌂ \* files to include searchViewlet.ts;queryBuilder.ts files to exclude

No results found in 'searchViewlet.ts;queryBuilder.ts' - Search again in all files - Learn More

queryBuilder.ts x html snippets.json package.json searchViewlet.ts

```

273 }
274
275 function splitGlobFromPath(searchPath: string): { pathPortion: string, globPortion?: string } {
276   const globCharMatch = searchPath.match(/[\*\{\}\(\)\[\]\?]/);
277   if (globCharMatch) {
278     const globCharIdx = globCharMatch.index;
279     const lastSlashMatch = searchPath.substr(0, globCharIdx).match(/[^/\\][^\\]*$/);
280     if (lastSlashMatch) {
281       let pathPortion = searchPath.substr(0, lastSlashMatch.index);
282       if (!pathPortion.match(/[^/\\]/)) {
283         // If the last slash was the only slash, then we now have '' or 'C:'. Append a slash.
284         pathPortion += '/';
285       }
286
287       return {
288         pathPortion,
289         globPortion: searchPath.substr(lastSlashMatch.index + 1)
290       };
291     }
292   }
293
294 // No glob char, or malformed

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Ln 136, Col 8 Tab Size: 4 UTF-8 CRLF TypeScript 2.6.2

Fig 3: Using ';' as separator, VScode could not distinguish different files.



What was intended was that VScode would act like this:

The screenshot shows the Visual Studio Code interface with a search results panel open. The search bar at the top has the text 'query'. Below it, the results list shows '132 results in 2 files' with two entries highlighted: 'searchViewlet.ts' and 'queryBuilder.ts'. A red circle highlights these two entries. The code editor on the right displays a portion of the 'queryBuilder.ts' file.

```

297     };
298 }
299
300 function patternListToIExpression(patterns: string[]): glob.IExpression {
301   return patterns.length ?
302     patterns.reduce((glob, cur) => { glob[cur] = true; return glob; }, Object.create(null)) :
303     undefined;
304 }
305
306 function splitGlobPattern(pattern: string): string[] {
307   return glob.splitGlobAware(pattern, ';')
308     .map(s => s.trim())
309     .filter(s => !s.length);
310 }
311
312 /**
313  * Note - we used {} here previously but ripgrep can't handle nested {} patterns. See https://github.com/
314 */
315 function expandGlobalGlob(pattern: string): string[] {
316   return [
317     `**/${pattern}/**`,
318     `**/${pattern}`
319   ];
  
```

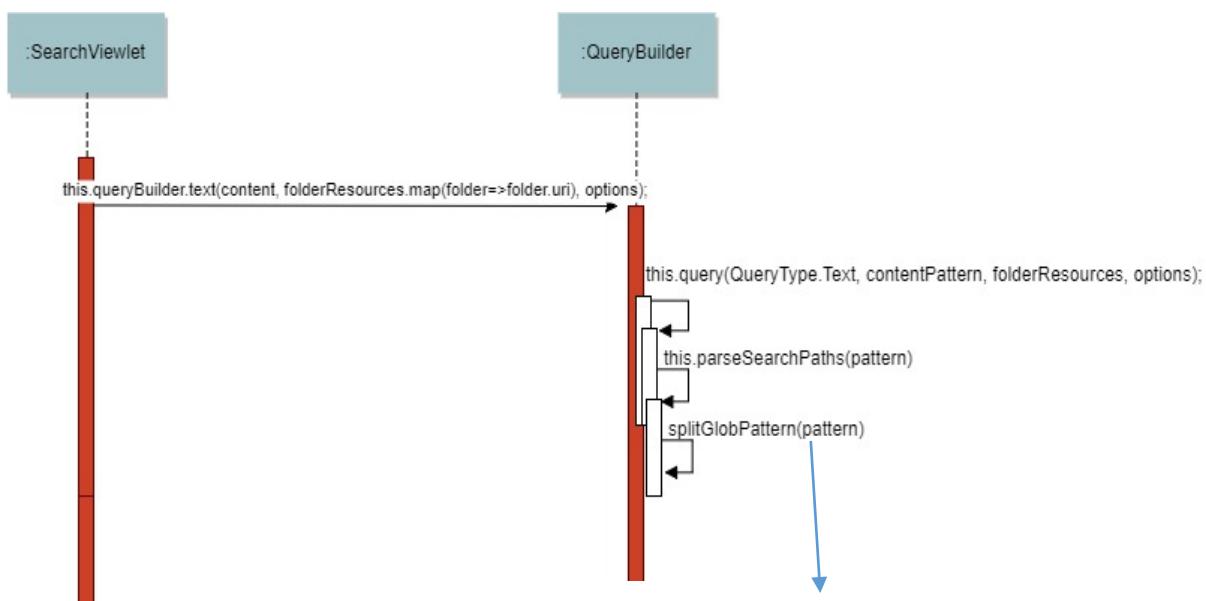
Fig 4: Using ';' as separator, like in Visual Studio.

### Source code files:

In order to solve the issue, we made a change in the function `splitGlobPattern(pattern: string)`, which can be found in the line 306, in the file [vscode/src/vs/workbench/parts/search/common/queryBuilder.ts](#).

### Design of the fix:

With this simple UML, we are trying to show the sequence of function calls that allowed us to plan and find the correct place to solve the issue:



And it was in this function that we changed the '' separator to ;'

**Fixed source code:**

In order to solve the issue, we changed the ‘,’ argument of function splitGlobAware, which is called inside the splitGlobPattern function mentioned above, to ‘;’.

```
src/vs/workbench/parts/search/common/queryBuilder.ts | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
diff --git a/src/vs/workbench/parts/search/common/queryBuilder.ts  
b/src/vs/workbench/parts/search/common/queryBuilder.ts  
index 5c8e0d65015..5c0cabdd0aa 100644  
--- a/src/vs/workbench/parts/search/common/queryBuilder.ts  
+++ b/src/vs/workbench/parts/search/common/queryBuilder.ts  
@@ -304,7 +304,7 @@ function patternListToIExpression(patterns: string[]):  
glob.IExpression {  
}  
  
function splitGlobPattern(pattern: string): string[] {  
-    return glob.splitGlobAware(pattern, ',')  
+    return glob.splitGlobAware(pattern, ';')  
        .map(s => s.trim())  
        .filter(s => !s.length);  
}
```

**Submit the Fix:**

The way it is supposed to submit fixes in our project is by doing a pull request, that can be found here:

<https://github.com/Microsoft/vscode/pull/39821>



## 2.2. Issue #39606

### Issue:

<https://github.com/Microsoft/vscode/issues/39606>

### Description:

Allow for Html code completion for scripts.

In this case, this user wanted to type some parts of script/Html relatable code and have VScode automatically suggest the completion of it.

The screenshot shows a GitHub comment from YanQi-Zhang. The comment text is as follows:

- VSCode Version: Code 1.18.1 (929bacb, 2017-11-16T18:32:36.023Z)
- OS Version: Windows\_NT x64 10.0.16299
- Extensions:
 

Extension	Author (truncated)	Version
vscode-eslint	dba	1.4.3
vscode-html-css	ecm	0.1.8
open-in-browser	tec	1.1.0

 (1 theme extensions excluded)

Steps to Reproduce:

- 1.<script type="">
- In the type tag  
The code can not be automatically completed  
For example, can not be automatically completed: text/javascript

Reproduces without extensions: Yes/No

On the right side of the screenshot, there are sections for Assignees (aeschli), Labels (feature-request, html), Projects (None yet), Milestone (Backlog), Notifications (Subscribe), and Participants (3 participants).

Fig 5: Issue #39606 description by the issue's submitter

### Requirements:

In terms of user stories, we can explain our requirements in this way:





Before the issue was solved, VScode would act like this:

```

1 <html>
2   <head>
3     <title>Hello Prello - Easy task Distribution</title>
4     <link rel="stylesheet" type="text/css" href="edit_profile.css">
5     <script src="edit_profile.js" defer></script>
6   </head>
7   <body>
8     <div id="myHeader" class="header">
9       <div id="myLogo" class="header">
10      <img alt="User Logo" href="#">
11    </div>
12    <div id="myName" class="header">Prello</div>
13    <div id="myNav" class="header">
14      <a href="#">Home
15      <a href="#">About
16      <a href="#">Contact
17      <a href="#">Logout
18    </div>
19  </div>
20  <div id="mySidebar" class="sidenav">
21    <img alt="Userfoto" href="#" alt="Userfoto" width="80" height="100">
22  </div>
23  <div id="myContent" class="content">
24    <div id="myInfo" href="#">
25      <img alt="Userfoto" href="#">
26      <a href="#" onclick="listTasks();">Tasks</a>
27      <a href="#" onclick="listSettings();">User</a>
28      <a href="#" onclick="listSocial();">Social</a>
29    </div>
30    <div id="myAbout" href="#">
31      <a href="#">Edit Profile</a>
32      <a href="#">Logout</a>
33    </div>
34  </div>
35</body>
36</html>

```

Fig 6: Different code typing that shows there is not a suggestion to automatically complete this html relatable code

What was intended was that VScode would act like this:

```

1 <html>
2   <head>
3     <title>Hello Prello - Easy task Distribution</title>
4     <link rel="stylesheet" type="text/css" href="edit_profile.css">
5     <script src="edit_profile.js" defer></script>
6   </head>
7   <body>
8     <div id="myHeader" class="header">
9       <div id="myLogo" class="header">
10      <img alt="User Logo" href="#">
11    </div>
12    <div id="myName" class="header">Prello</div>
13    <div id="myNav" class="header">
14      <a href="#">Home
15      <a href="#">About
16      <a href="#">Contact
17      <a href="#">Logout
18    </div>
19  </div>
20  <div id="mySidebar" class="sidenav">
21    <img alt="Userfoto" href="#" alt="Userfoto" width="80" height="100">
22  </div>
23  <div id="myContent" class="content">
24    <div id="myInfo" href="#">
25      <img alt="Userfoto" href="#">
26      <a href="#" onclick="listTasks();">Tasks</a>
27      <a href="#" onclick="listSettings();">User</a>
28      <a href="#" onclick="listSocial();">Social</a>
29    </div>
30    <div id="myAbout" href="#">
31      <a href="#">Edit Profile</a>
32      <a href="#">Logout</a>
33    </div>
34  </div>
35</body>
36</html>

```

Fig 7: Now, we can see that VScode suggests completion for html relatable code.



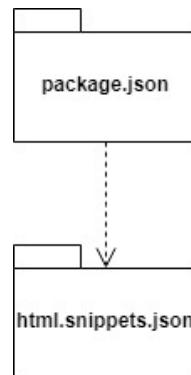
### Source code files:

In order to solve the issue, we added new snippets in file [extensions/html/snippets/html.snippets.json](#), from lines 24 to 55.

### Design of the fix:

To solve this issue, we thought of how VSCode could suggest code completion.

We thought a good way would be to add some snippets. Since this issue was specific to html code, we found the place where these snippets were mentioned and followed the path:



### Fixed source code:

```

extensions/html/snippets/html.snippets.json | 33
+++++
1 file changed, 33 insertions(+)

diff --git a/extensions/html/snippets/html.snippets.json b/extensions/html/snippets/html.snippets.json
index 7a858455a69..777b5fdce28 100644
--- a/extensions/html/snippets/html.snippets.json
+++ b/extensions/html/snippets/html.snippets.json
@@ -18,5 +18,38 @@
        "</html>"
    ],
    "description": "Simple HTML5 starting point"
+
},
+
+
+ "Script Type": {
+     "prefix": "type",
+     "body": [
+         "${1|type= text/javascript,type= application/javascript|}"
+     ],
+     "description": "Define type to text/javascript"
},
+
+ "Script async": {
+     "prefix": "as",

```



```
+     "body": [
+         "$async></script>"
+     ],
+     "description": "Script is run as soon as it is downloaded"
+ },
+
+ "Script defer": {
+     "prefix": "def",
+     "body": [
+         "defer></script>"
+     ],
+     "description": "Script is executed only when the page is loaded and
in order."
+ },
+
+ "Script source": {
+     "prefix": "sr",
+     "body": [
+         "src=\"$1:foo}.js\" "
+     ],
+     "description": "External script being used here."
}
}
```

**Submit the Fix:**

The way it is supposed to submit fixes in our project is by doing a pull request, that can be found here:

<https://github.com/Microsoft/vscode/pull/39839>