✓ 100 XP

Exercise - Use Copilot to autocomplete code

10 minutes

As you develop with Copilot, you'll find that it can help you complete your code faster. Copilot can autocomplete code that you're writing, and also suggest code that you might need. In this exercise, you'll use Copilot to generate new methods in the Trie project. Let's get started!

Use Copilot to autocomplete code

1. Open the **Trie.cs** file

2. Enter a new line after the `Trie` constructor, then enter the following code:

   c#

   ```csharp
   // Search for a word in the trie
   public bool Search(string word)
   {
   ```

3. Wait for Copilot to generate code.

   Copilot might suggest code similar to the following code block:

   c#

   ```csharp
   // Search for a word in the trie
   public bool Search(string word)
   {
       TrieNode current = root;
       foreach (char c in word)
       {
           if (!current.HasChild(c))
           {
               return false;
           }
           current = current.Children[c];
       }
       return current.IsEndOfWord;
   }
   ```

4. Press `Tab` or click **Accept** to apply the code suggestion.

5. Navigate to the `AutoSuggest` method.

   This method relies on another method `GetAllWordsWithPrefix` that is incomplete. You can use Copilot to generate this method.

6. Navigate to the `GetAllWordsWithPrefix` method and remove the return statement;

   c#

   ```
   private List<string> GetAllWordsWithPrefix(TrieNode node, string prefix)
   {

   }
   ```

7. Wait for Copilot to generate code.

   Copilot should automatically suggest code when you remove the return statement. If not, you can try entering the following: `List<string> words =` and Copilot should complete the code.

   Copilot might suggest code similar to the following code block:

   c#

   ```
   private List<string> GetAllWordsWithPrefix(TrieNode node, string prefix)
   {
       List<string> words = new List<string>();

       if (node.IsEndOfWord)
       {
           words.Add(prefix);
       }

       foreach (var child in node.Children)
       {
           words.AddRange(GetAllWordsWithPrefix(child.Value, prefix + child.Key));
       }

       return words;
   }
   ```

8. Press `Tab` or click **Accept** to apply the code suggestion.

Check your work

In this task, you'll test the methods you created with Copilot to verify that they work as expected.

1. Open the **Program.cs** file.

2. Uncomment the `SearchWord` method call:

   C#

   ```
   Trie dictionary = InitializeTrie(words);
   SearchWord();
   // PrefixAutocomplete();
   // DeleteWord();
   // GetSpellingSuggestions();
   ```

3. Navigate to the `SearchWord()` method and uncomment the following code block:

   C#

   ```
   /*
   if (input != null && dictionary.Search(input))
   {
       Console.WriteLine($"Found \"{input}\" in dictionary");
   }
   */
   ```

4. In the file explorer, right click the **Program.cs** file and click **Open in Integrated Terminal**

5. Enter `dotnet run` to run the program.

6. Enter a word to search for such as "follows" or "beans"

7. Verify that your output is similar to the following

   Output

   ```
   Enter a word to search for, or press Enter to exit.
   follows
   Found "follows" in dictionary
   Enter a word to search for, or press Enter to exit.
   beans
   Did not find "beans" in dictionary
   Enter a word to search for, or press Enter to exit.
   ```

If your code displays different results, review your code to find your error and make updates. Run the code again to see if you've fixed the problem. Continue updating and running your code until your code produces the expected results.

8. Comment any method calls and uncomment the `PrefixAutocomplete` method call:

```c#
Trie dictionary = InitializeTrie(words);
// SearchWord();
PrefixAutocomplete();
// DeleteWord();
// GetSpellingSuggestions();
```

9. Enter `dotnet run` to run the program.

10. Enter a prefix to search for such as "pl" or "fo"

    Press Tab to autocomplete the prefix. You can press Tab multiple times to cycle through the suggestions. Press space to accept the suggestion and start a new word.

11. Verify that your output is similar to the following

```Output
The dictionary contains the following words:
as, astronaut, asteroid, are, around, cat, cars, cares, careful, carefully,
for, forgot, follows, from, front, mellow, mean, money, monday,
monster, place, plan, planet, planets, plans, the, their, they, there,
towards,

Enter a prefix to search for, then press Tab to cycle through search
results. Press Enter to exit.
place plan forgot
```

    If your code displays different results, review your code to find your error and make updates. Run the code again to see if you've fixed the problem. Continue updating and running your code until your code produces the expected results.

12. Comment out the `PrefixAutocomplete` method call.

Next unit: Exercise - Prompt Copilot to generate code

‹ Previous Next ›