TRGCON 24

AI Code Assistants

Following practical part

github.com/DivergerThinking/TRGCON24/

Overview

Release date: October 2021
Extension para Visual Studio, VSCode, Jetbrain, …
Ownership: Github (Microsoft)

Release date: May 2023
VSCode Fork
Ownership: Cursor (startup)

Features

Autocomplete
AI guesses what you are trying to do and **suggest code without you asking**

Inline chat
Ask AI to modify **specific sections of code**

Chat
Ask AI to explain, modify or create code at **high or low-level**

Code edits / Composer
Similar to Chat, but **focuses on code editing**

Customize instructions
Give **specific instructions** to the AI on how to respond to requests

Autocomplete

Context used:
Each tool will have their own way of handling context, but in general, the context included will be:

- File open and cursor position
- Other files opened in your IDE
- Previous edits (Cursor)
- …

```
class StudentRegistry:
    def __init__(self):
        self.students: Dict[int, Student] = {}

    def get_student(self, student_id: int) -> Optional[Student]:
```

Autocomplete

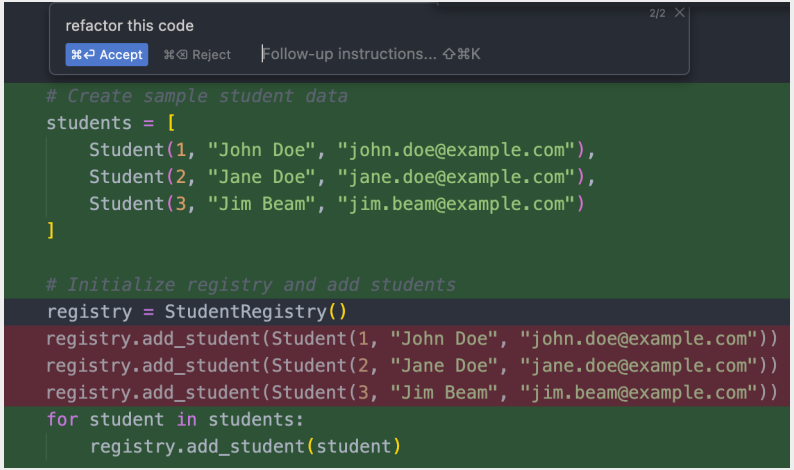
DEMO

Open student.py and add the following to the StudentRegistry class:

```
=====

def calculate_average

=====

def find_students_in_course(self, course_name: str) -> List[Student]:

=====

def sort_students_by_gpa(self) -> List[Student]:
"""
Sort all students in the registry by their GPA in descending order.
Returns a list of students from highest to lowest GPA.
"""

=====

# modify add_course to append_course

=====
```

Inline chat

Scope
generate some changes on specific code selections

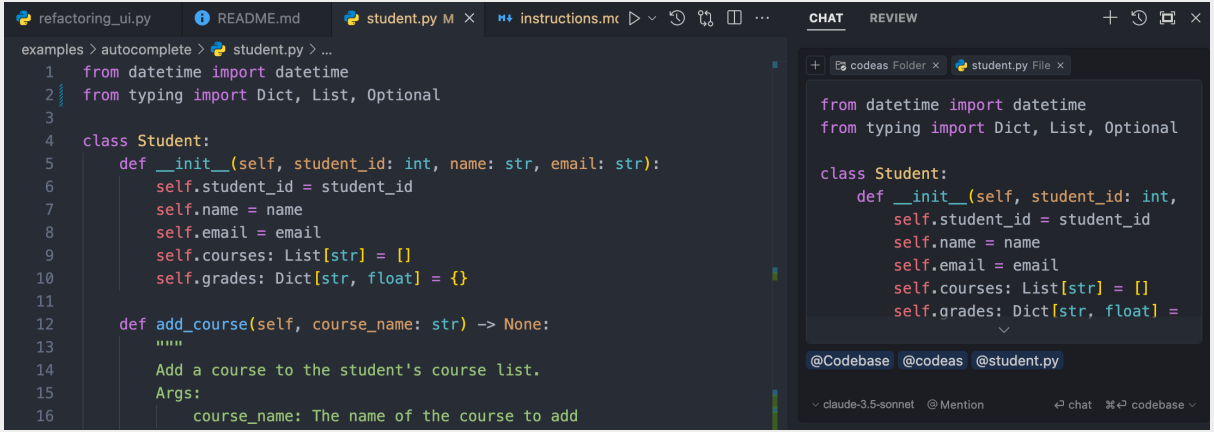


DEMO
Select some code in student.py, trigger inline chat and ask the following:

refactor the following code

Chat

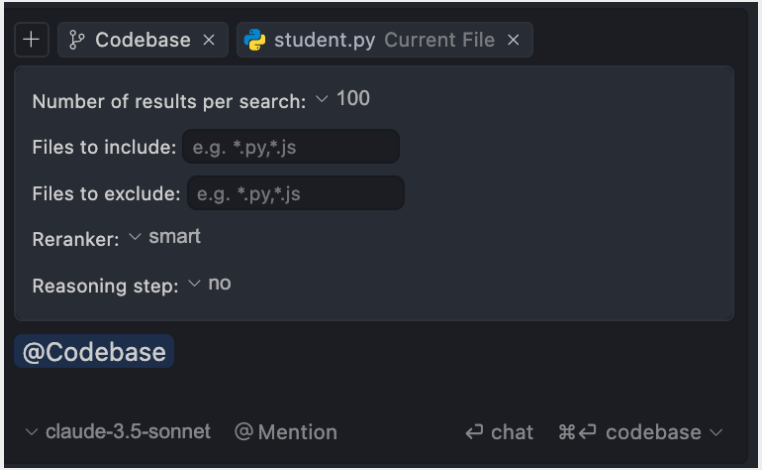
Scope:
AI integrated to your codebase to whom you can ask questions at **high-level** and/or modifications at **low-level**



Context handling:
Project level: @workspace / @codebase
Folder level: @folder (Cursor only)
(Multiple) File level : @files / #files

@workspace (Copilot) / @codebase (Cursor)

Context
Chunks of the codebase retrieved through semantic search



Use cases
Explain the codebase
Search through the codebase
High level questions about the codebase

@workspace (Copilot) / @codebase (Cursor)

DEMO

Open chat interface and ask the following questions about the codeas/ codebase using @codebase (cusor) or @workspace (copilot):

Explain what the project “codeas” does at high level. Be concise and do not reference code.

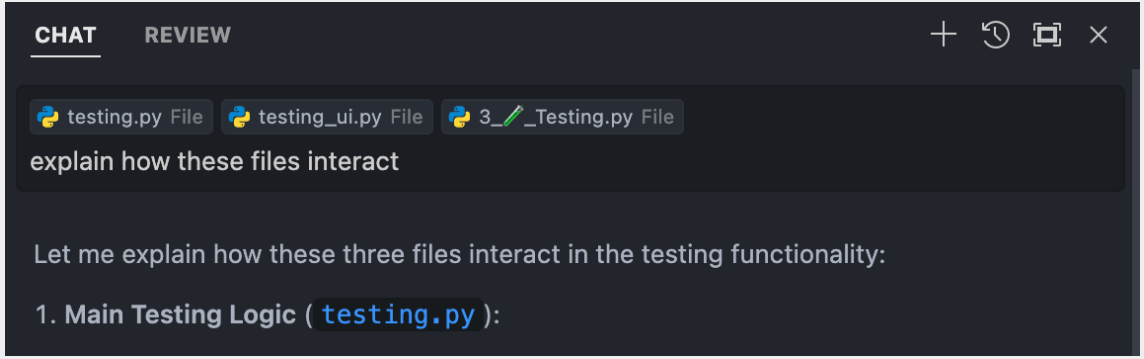
Tell me about the current project structure

How do I run this app

Which part of the codebase is responsable for executing the LLM requests

@files / #files

Context:
Full content of the files added



DEMO
Add testing_ui.py and ask the following:

How can I make this code more readable?

Suggest renaming some of the variables and functions, so that they are more descriptive

Code editor / Composer

New features (!)
These features are quite recent and still seem a little buggy, but it could be that this is the future of programming

Focused on code editing (generate the code diffs immediately, creates new files if needed, etc.)

DEMO
add testing.py, testing_ui and 3_Testing.py and ask:

Create a new use case that's capable of translating a codebase. It should first generate the strategy for translating the codebase, then generate the translations and finally write the translated files to the codebase. Refer to the testing functionality for implementation details, but do not modify it in any way.

Customized rules for your assistant

The LLM behind the chat interface can be “customized” by adding instructions for him to follow for each request:

DEMO
Create the following file inside the repository:
.cursorrules (Cursor)
.github/copilot-instructions.md (Copilot)

Add some instructions for the AI to follow:
Answer me in a single sentence

Examples of custom instructions to use
https://cursor.directory/

Summary

Copilot & Cursor's core functionalities are very similar

There are some **additional features** we didn't cover here:

- Code reviews
- PR reviews (Copilot)
- Git commit generation (Copilot)
- …

Can significantly boost developer productivity but still have **limitations:**

- The AI (autocomplete, chat or code edits) often misses global understanding of the codebase
- Hard to manage context for large complex codebase (@codebase not sufficient)

Codeas

Aims to tackle some of the limitations found in other code assistants

Open Source
github.com/DivergerThinking/codeas

Demo
codeas-diverger.streamlit.app/Documentation

Codeas documentation generation