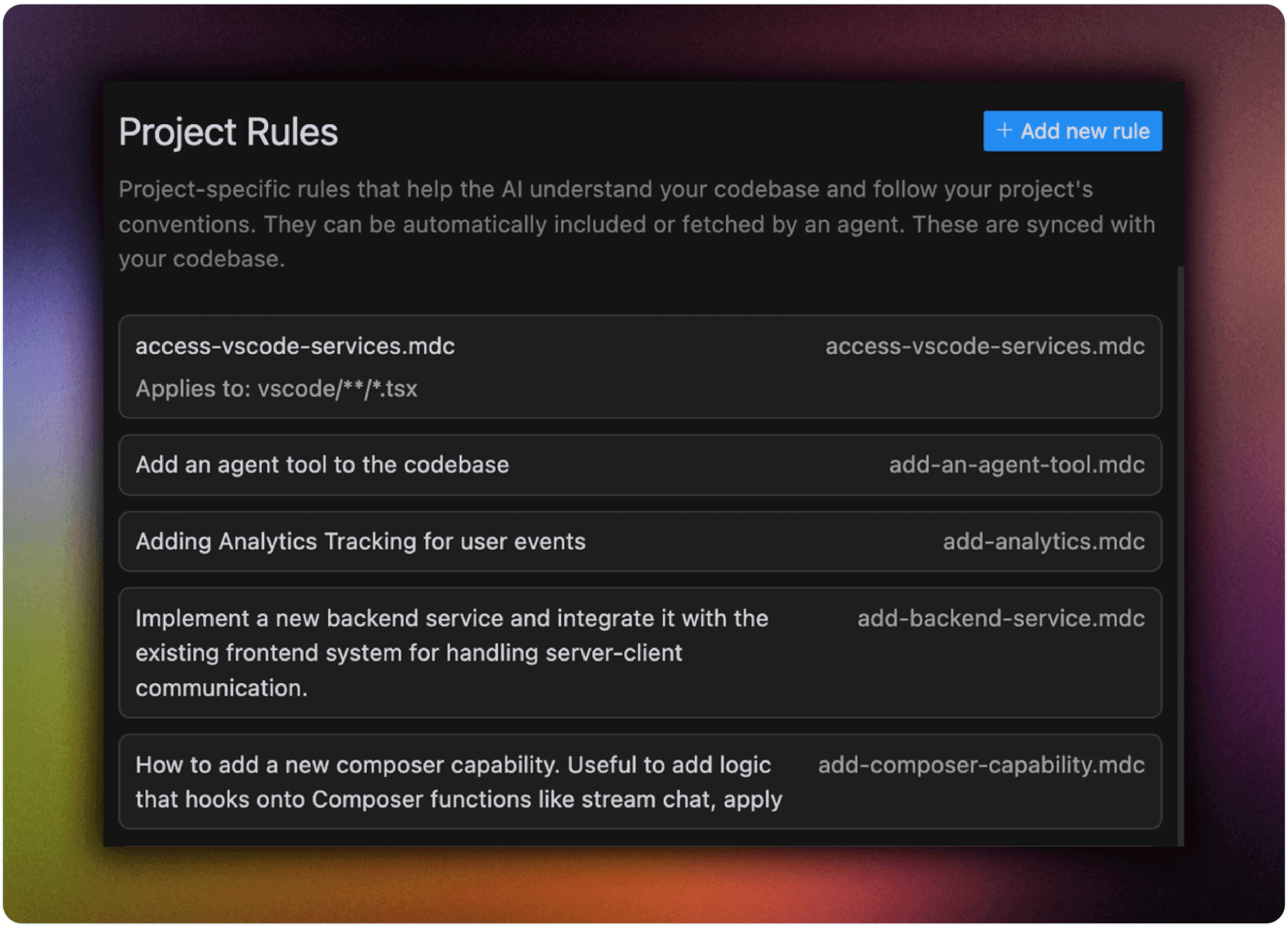
Michael Feldstein
@msfeldstein
10h • 7 tweets • 3 min read • Read on X

One of the things I've been working on at @cursor_ai is beefing up Cursor Rules. We want Agent to be as powerful as the most knowledgable person on your team. Here's how we use them at Cursor. 🧵

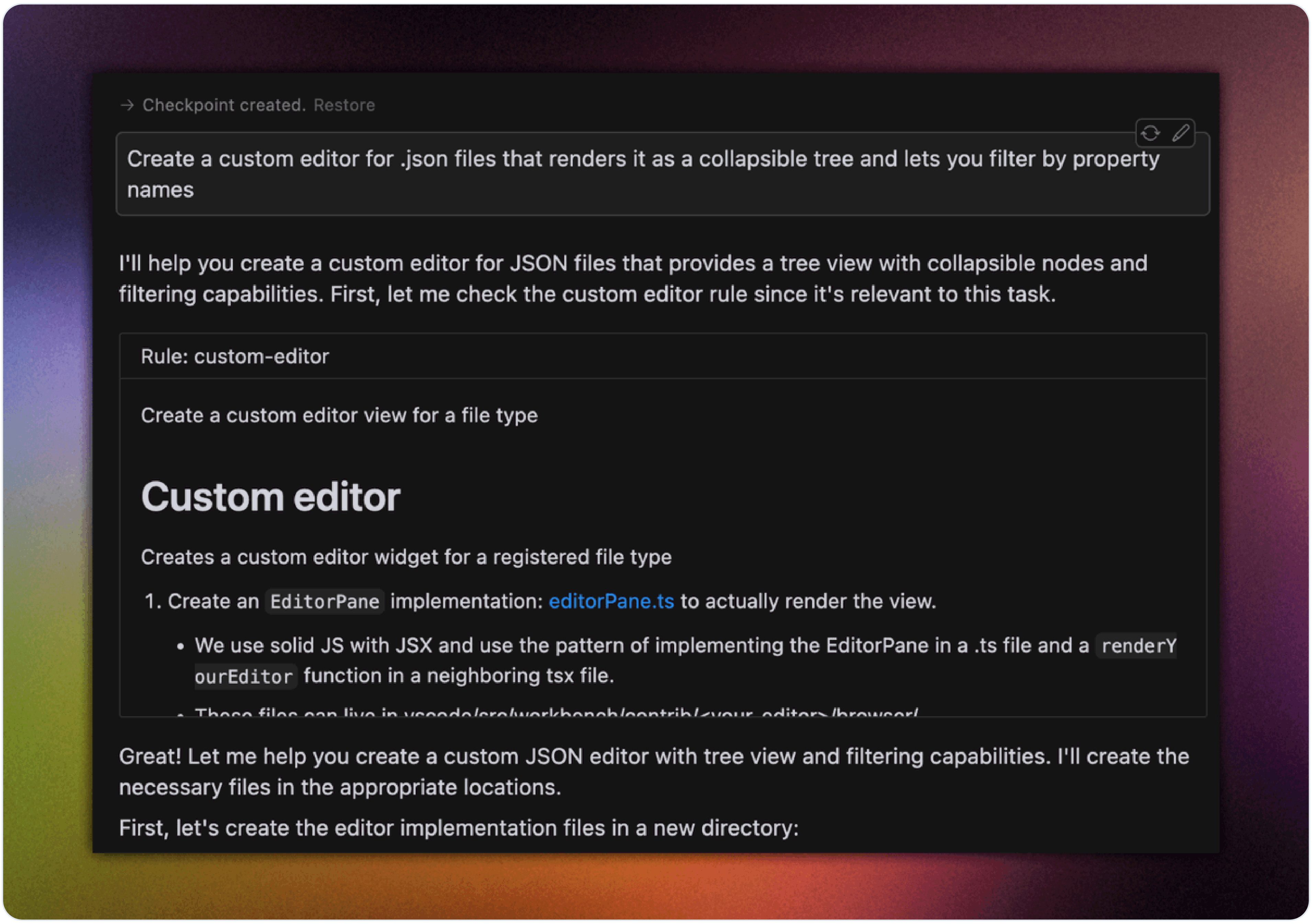


Project Rules

Project-specific rules that help the AI understand your codebase and follow your project's conventions. They can be automatically included or fetched by an agent. These are synced with your codebase.

access-vscode-services.mdc Applies to: vscode/**/*.tsx	access-vscode-services.mdc
Add an agent tool to the codebase	add-an-agent-tool.mdc
Adding Analytics Tracking for user events	add-analytics.mdc
Implement a new backend service and integrate it with the existing frontend system for handling server-client communication.	add-backend-service.mdc
How to add a new composer capability. Useful to add logic that hooks onto Composer functions like stream chat, apply	add-composer-capability.mdc

Project Rules can be auto-attached to any composer/agent request based on filename globbing, or the agent can grab one based on its description. These rules are checked into your codebase so anyone can make use of them.



→ Checkpoint created. Restore

Create a custom editor for .json files that renders it as a collapsible tree and lets you filter by property names

I'll help you create a custom editor for JSON files that provides a tree view with collapsible nodes and filtering capabilities. First, let me check the custom editor rule since it's relevant to this task.

Rule: custom-editor

Create a custom editor view for a file type

Custom editor

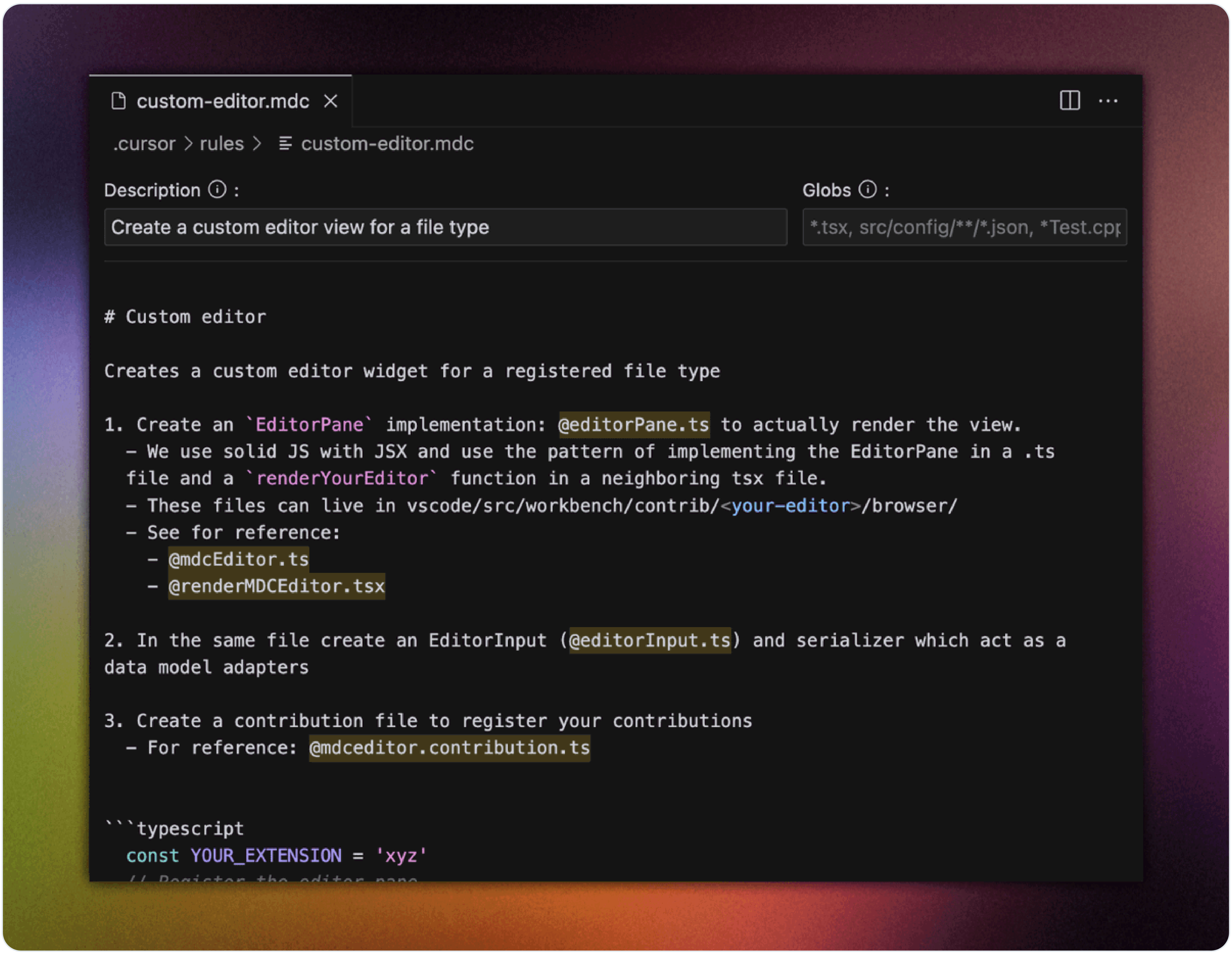
Creates a custom editor widget for a registered file type

1. Create an `EditorPane` implementation: editorPane.ts to actually render the view.
   - We use solid JS with JSX and use the pattern of implementing the EditorPane in a .ts file and a `renderYourEditor` function in a neighboring tsx file.
   - These files can live in vscode/src/workbench/contrib/<your-editor>/browser/

Great! Let me help you create a custom JSON editor with tree view and filtering capabilities. I'll create the necessary files in the appropriate locations.

First, let's create the editor implementation files in a new directory:

Project rules can be very detailed. The first time i had to write a custom editor UI it was difficult to figure out how to wire it up. But I put that knowledge into a rule and now i just ask agent "Make a custom editor UI that does X"



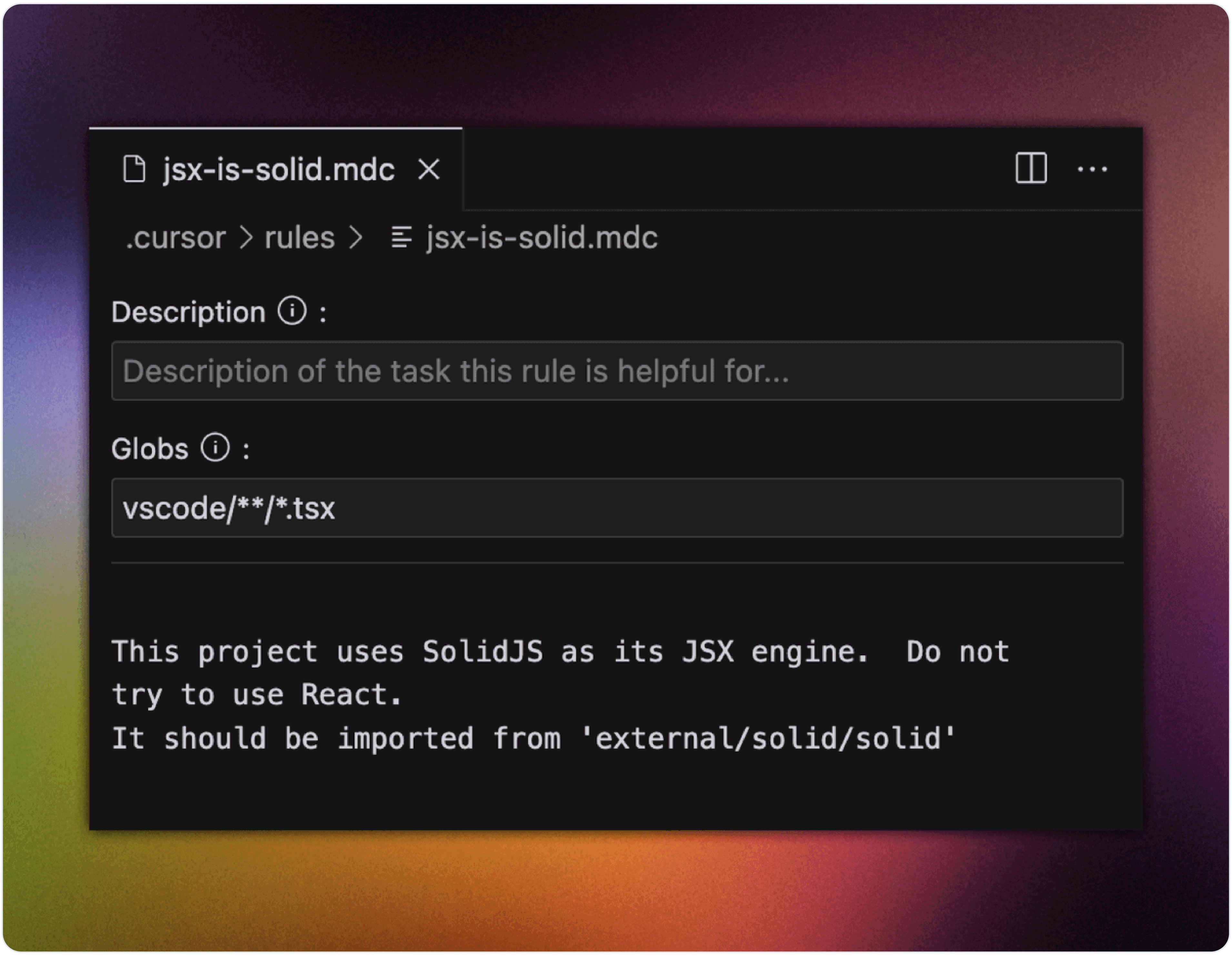
Description ⓘ : Create a custom editor view for a file type

Globs ⓘ : *.tsx, src/config/**/*.json, *Test.cpp

~~~
# Custom editor

Creates a custom editor widget for a registered file type

1. Create an `EditorPane` implementation: @editorPane.ts to actually render the view.
  - We use solid JS with JSX and use the pattern of implementing the EditorPane in a .ts
  file and a `renderYourEditor` function in a neighboring tsx file.
  - These files can live in vscode/src/workbench/contrib/<your-editor>/browser/
  - See for reference:
    - @mdcEditor.ts
    - @renderMDCEditor.tsx

2. In the same file create an EditorInput (@editorInput.ts) and serializer which act as a
data model adapters

3. Create a contribution file to register your contributions
  - For reference: @mdceditor.contribution.ts


```typescript
  const YOUR_EXTENSION = 'xyz'
  // Register the editor pane
~~~

They can also be simple. We use SolidJS instead of react, but the AI would often give us React code. We can't use the original .cursorrules which get included with every request, especially since our dashboards _do_ use react.

This simple rule gets included with all front-end tsx files, and i haven't seen unwanted react since.



Description ⓘ :

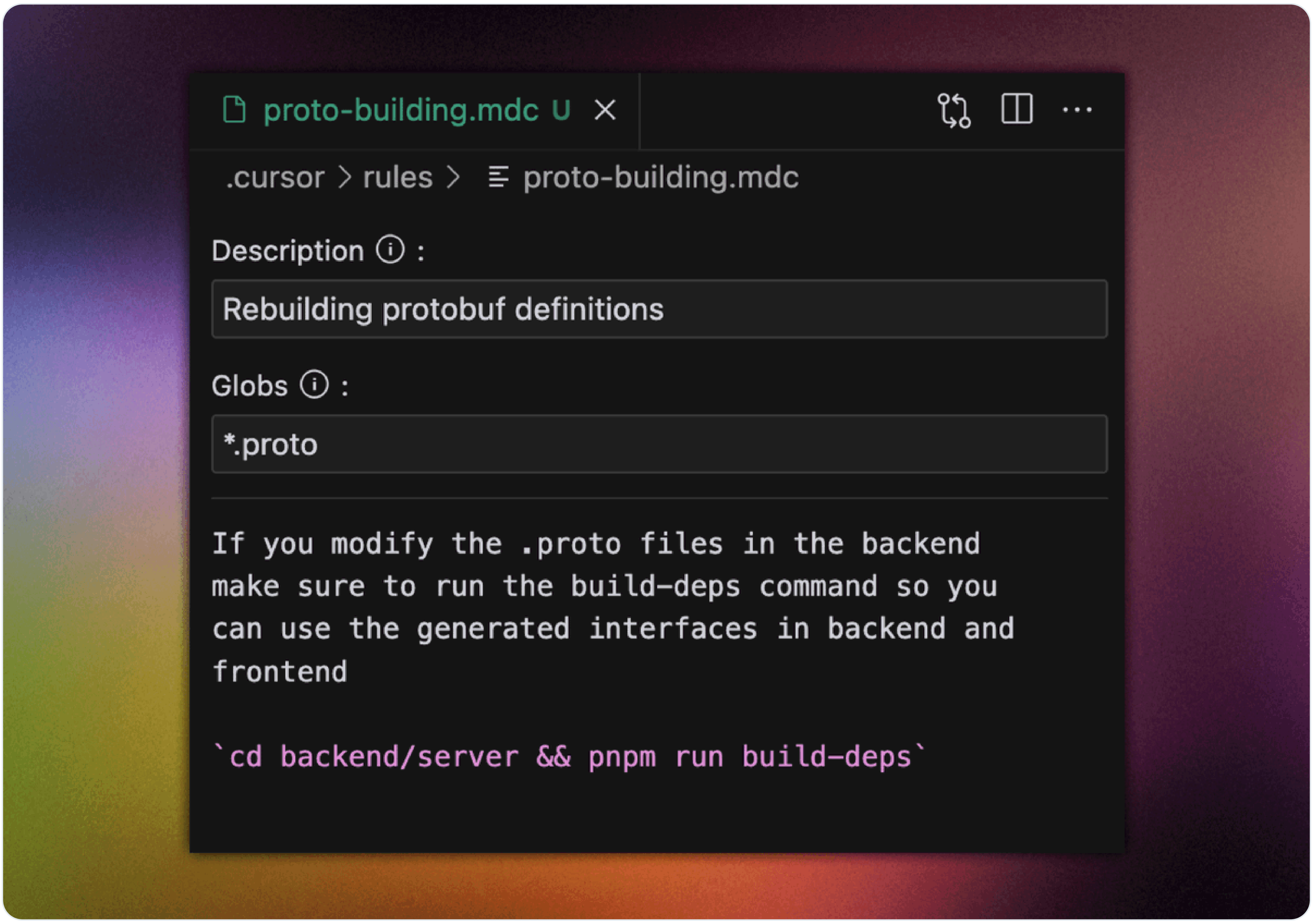
Description of the task this rule is helpful for...

Globs ⓘ :

vscode/**/*.tsx

```
This project uses SolidJS as its JSX engine.  Do not
try to use React.
It should be imported from 'external/solid/solid'
```

Some of our code is auto-generated, and if the AI modifies a .proto file, it used to need human intervention to generate the code before it can continue working, with a simple rule added to all .proto modifications, it can just keep moving.



Description ⓘ :

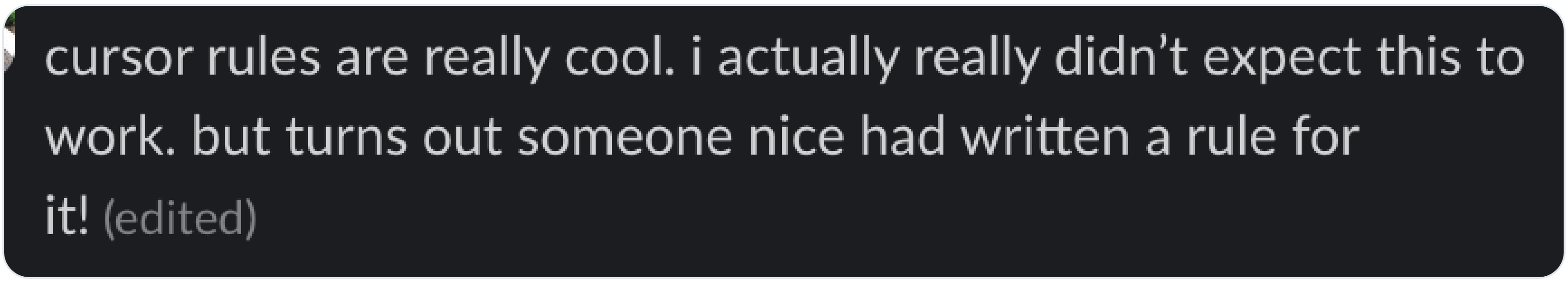
Rebuilding protobuf definitions

Globs ⓘ :

*.proto

```
If you modify the .proto files in the backend
make sure to run the build-deps command so you
can use the generated interfaces in backend and
frontend

`cd backend/server && pnpm run build-deps`
```

We have a _ton_ of ideas how to make these easier to author, easier to use, more powerful, but we're already seeing our agent do things it definitely couldn't before. More to come!



cursor rules are really cool. i actually really didn’t expect this to work. but turns out someone nice had written a rule for it! (edited)

also we're hiring

https://anysphere.inc/

• • •

Missing some Tweet in this thread? You can try to force a refresh