

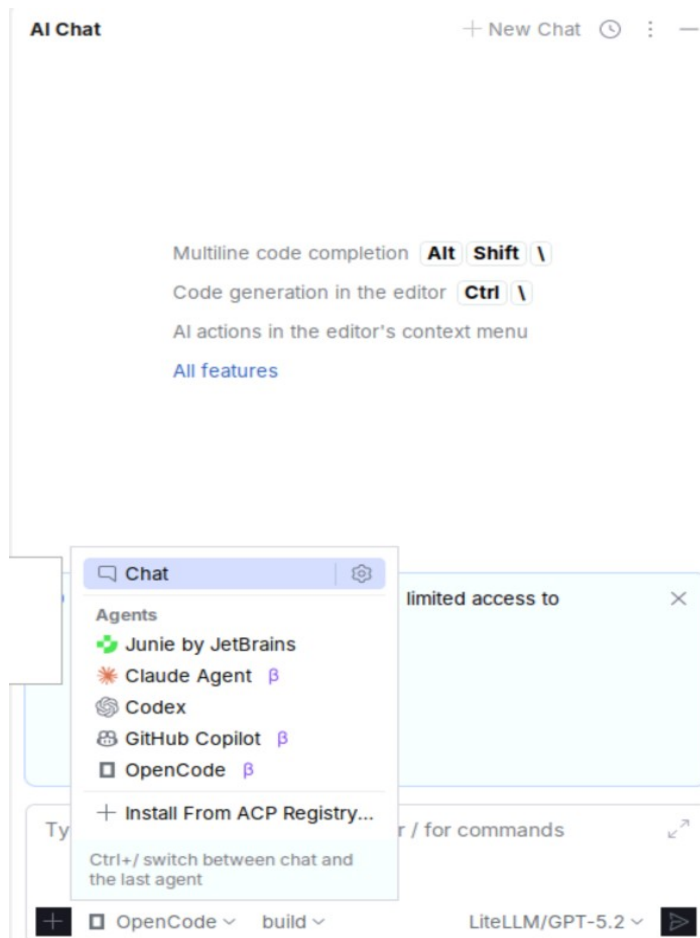
Legacy AI tooling workshop – prerequisites

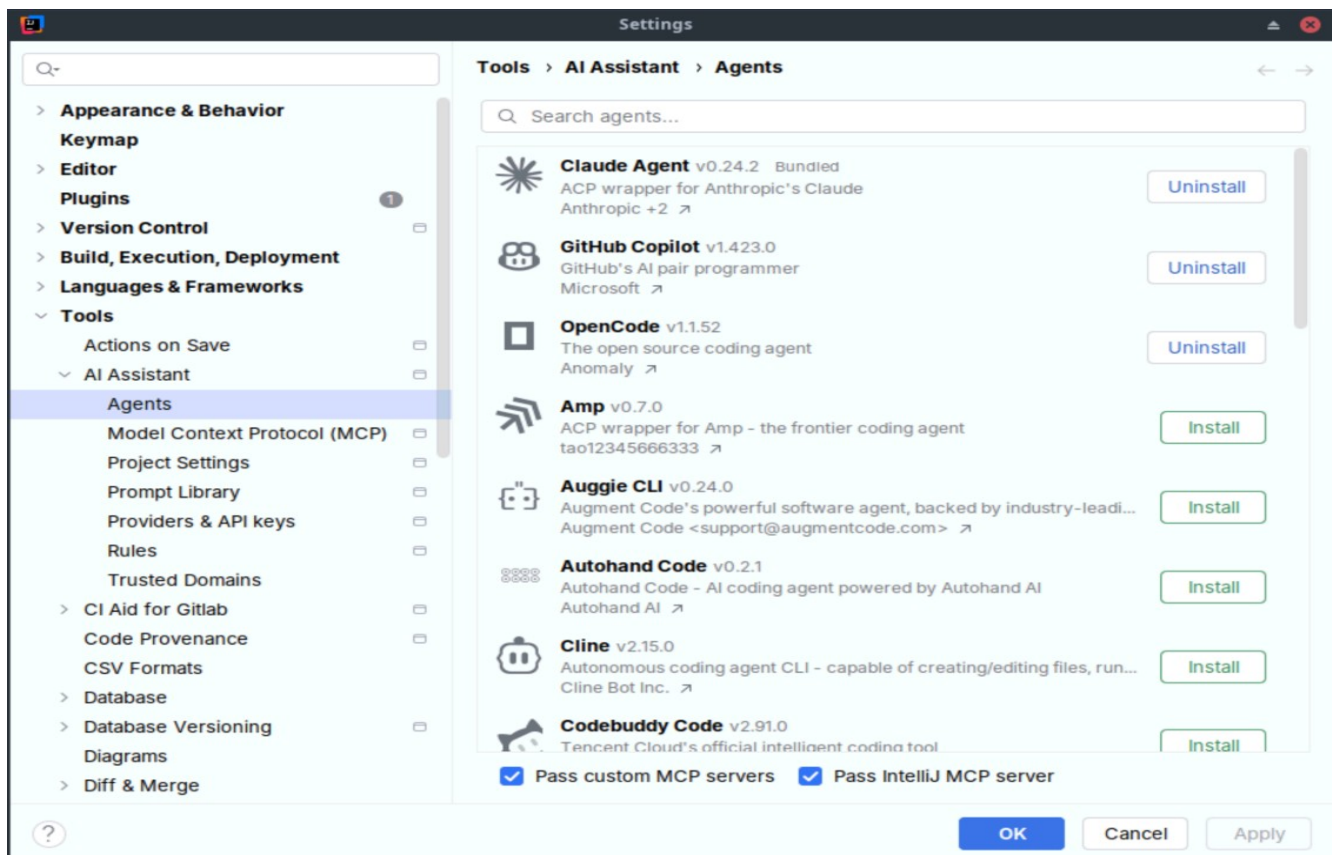
To make sure we don't waste too much time doing the setup, the following steps should be done before.

1. Latest IntelliJ and plugins

Make sure you're running IntelliJ IDEA Ultimate or Community, but with the latest version possible.

Then, once that's installed, go to the "AI chat" feature and click "install from ACP registry". Search for OpenCode and make sure it's installed:





Make sure you enable “pass MCP server” and “custom MCP” servers in the options.

2. Setup OpenCode

We’ll be using OpenCode as part of our AI tooling workshop, so you need to go to <https://opencode.ai/> and install it on your machine (supports Mac, Windows, Linux).

Once you installed opencode, follow the config instructions and create an opencode.json file, at the config location (see <https://opencode.ai/docs/config/#file-based>), on my Linux it’s at `~/.config/opencode/opencode.json`.

```
{
  "$schema": "https://opencode.ai/config.json",
  "mcp": {
    "legacy_docs": {
      "enabled": true,
      "type": "remote",
      "url": "http://localhost:8081/sse"
    }
  },
  "provider": {
    "litellm": {
      "npm": "@ai-sdk/openai-compatible",
      "name": "LiteLLM",
      "options": {
        "baseURL": "https://nginx-proxy-llmproxy.app.cern.ch/v1",
        "apiKey": "XXXX"
      }
    }
  },
  "models": {
```

```
"devstral-medium-latest": {  
  "name": "Devstral Medium"  
},  
"gpt-5.2-codex": {  
  "name": "Gpt 5.2 Codex"  
}  
}  
}
```

The apiKey will be communicated to attendees during the workshop, you can leave it like this for now.

Save the file, then type “opencode” in your terminal or use the GUI if you prefer to do so on Windows. You should be able to interact with some models and ask something in the “plan” phase.

You can also use the default OpenCode Zen models for free, albeit at a slower rate.