



WiFi Access
SSID: LE Guest
Password: B3OurGu3st



Workshop Resources

Cursor Workshop

Kickoff

Setting the stage with our goals, agenda, and constraints



Workshop Resources



Housekeeping Notes



WiFi Access

SSID: **LE Guest**
Password: **B3OurGu3st**



Slides & Materials

Available at
LeadingEDJE/CursorWorkshop



Stuck?
Confused?
Curious?

There's 3 of us here for a reason



Bathrooms

Outside of our space in the hall area
Floor doors are locked and require a keyfob

Who We Are

At Leading EDJE, we believe technology is the most powerful tool for transforming businesses and unlocking human potential.

As strategists, innovators, and partners, we don't just solve problems — we craft bespoke solutions that positively disrupt your business.



Introductions



Bryce Amacker



Matt Eland



Dave Michels

Learning Goals

Understand concepts and terminology
related to AI assisted development

Proficiency in using Cursor for greenfield
and brownfield development

Apply appropriate solutions for enhancing
Cursor's context in your project in ways
that are compliant with company practices

AI Assisted Development Philosophy

We believe that AI tooling can help developers become more productive when writing code, be more effective at navigating and maintaining code that is new to them, and find specialized solutions in areas that challenge the developer's own expertise.

Ultimately, AI tooling provides developers new options to consider when trying to provide quality technical solutions in timelines that support the needs of the business.

Agenda

Time	Content
8 – 9:15 AM	Introductions, Terminology, and Setup Verification
9:30 – 10:30 AM	Tooling Discussion and Demonstration
10:45 – 11:30 AM	Cursor for Greenfield Development
11:30 AM – 12:30 PM	Catered Lunch
12:30 – 1:15 PM	Context Discussion and Sample App Exploration
1:15 – 2 PM	Cursor for Brownfield Development
2:15 – 3:45 PM	Advanced AI Integration
4 – 5 PM	Final discussions and topics of interest



Our codebase for this afternoon

Terminology

Laying a foundation for the rest of today's discussions and demos

Context

Editor Context

- The current active file is app.js
- Look at all files inside the Services directory

Setting Context

- It is early December in 2025
- We are in Columbus, Ohio
- This codebase is a production codebase that does X

Process Context

- It is critical that the application log error diagnostic information
- Run tests after making any change



Agents

Editor



Selection View Go Run ...

JAIMES AF



Search Agents

New Agent

AWAITING REVIEW

Increase spacing ... 16h

RECENT

Improve loading ... 16h

Enable markdown... 16h

OLDER

Improve chat bub... 16h

Configure ReShar... 16h

Consolidate agen... 16h

Modify GameDto ... 16h

Combine chat res... 16h

Show more

ARCHIVED

Please refactor the code in `C# ChatService.cs` to increase readability and testability

Agent ▾ Auto ▾

@

Local ▾

Shift Tab to plan

✗ ⌂ main* ⌂ ⌂ 0 ⌂ 1 ⌂ 3 ⌂ .NET Core Launch (AppHost) (JAIMES AF) ⌂ Connect

Ln 18, Col 1 Spaces: 4 UTF-8 with BOM CRLF { C# net10.0

Agents in Cursor



Improve loading display layout

Now · +46 -39 · Auto

At present the loading display in `Games.razor` causes the existing layout to flow around the spinner and looks unprofessional. I'd like a more consistent layout during the loading experience that doesn't cause the text on the page to move around when the spinner goes away.

Explored 2 files 1 search

The `if-else` chain causes layout shifts. Reserving space for the content area and overlaying the spinner to prevent movement. Updating the layout:

`Games.razor +46 -39`

```
<div style="position: relative; min-height: 200px;">
    @if (isLoading)
    {
        <div style="display: flex; justify-content: center; align-items: center; min-height: 200px;">
            <MudProgressCircular Indeterminate="true"/>
        </div>
    }
</div>
```

Checking for linting errors:

No linter errors found in `Games.razor`

Updated the loading display to prevent layout shifts.

Changes made:

1. Reserved space container: Wrapped the conditional content in a `div` with `min-height: 200px` to reserve space and prevent shifts.
2. Centered spinner: During loading, the spinner is centered in a flex container within the reserved space.
3. Independent conditions: Replaced the `if-else` chain with separate `if` statements so each state renders independently.

The container maintains a minimum height during loading, so when the spinner disappears and the table or empty state appears, the layout stays stable. The spinner is centered in the reserved area for a consistent look.

The page should now maintain a stable layout during loading transitions.

1 File Edited Review

`Games.razor +46 -39`

Agents in Cursor

Tools

Browser

Browser Automation
Ready (Chrome detected)

Show Localhost Links in Browser
Automatically open localhost links in the Browser Tab

Google Chrome

Installed MCP Servers

G GitKraken

```
c:\Users\matt.eland\AppData\Roaming\Cursor\User\globalStorage\eamodio.gitlens\gk.exe mcp --host=cursor --source=gitlens --scheme=cursor
git_add_or_commit git blame git_branch git_checkout
git_log_or_diff git_push git_stash git_status git_worktree
gitkraken_workspace_list issues_add_comment issues_assigned_to_me
issues_get_detail pull_request_assigned_to_me pull_request_create
pull_request_create_review pull_request_get_comments
pull_request_get_detail repository_get_file_content
code_review_branch
```

L linear

```
npx -y mcp-remote https://mcp.linear.app/sse
list_comments create_comment list_cycles get_document
list_documents get_issue list_issues create_issue update_issue
list_issue_statuses get_issue_status list_issue_labels
create_issue_label list_projects get_project create_project
update_project list_project_labels list_teams get_team list_users
get_user search_documentation
```

+ New MCP Server
Add a Custom MCP Server

Thought for 2s

Checking if IMessagePublisher is registered and updating Program.cs:
Grepped IMessagePublisher in Program.cs
Read Program.cs L180-219
Thought for 4s

Updating Program.cs to register IMessagePublisher and both strategies conditionally:

```
C# Program.cs +51-29
// Register IMessagePublisher for queuing chunks without embeddings
builder.Services.AddSingleton();

// Register chunking strategy based on configuration
string chunkingStrategy = options.ChunkingStrategy ?? "SemanticChunker";
bool useSemanticChunker = string.Equals(chunkingStrategy, "SemanticChunker", StringComparison.OrdinalIgnoreCase);

C# Program.cs +6-2
logger.LogInformation("chunking strategy: {Strategy}", chunkingStrategy);
if (useSemanticChunker)
{
    logger.LogInformation("Ollama Endpoint: {Endpoint}", ollamaEndpoint);
    logger.LogInformation("Ollama Model: {Model}", ollamaModel);
}
```

Read Program.cs L1-25
Thought for 4s

Adding the using statement and checking for compilation errors:
No linter errors found in JAIMES_AF.Workers.DocumentChunking
Read SemanticSliceStrategy.cs
Thought for 3s

Building the solution to check for compilation errors:

Ran command: dotnet

```
$ dotnet build "JAIMES_AF.Workers.DocumentChunking/JAIMES_AF.Workers.DocumentChunking.csproj"
1 Error(s)
```

Time Elapsed 00:00:08.24

Tools and Tool Calling

```

public class ChatService(ChatOptions options, ILogger<ChatService> logger,
IChatHistoryService chatHistoryService) : IChatService
{
    private readonly ChatOptions _options = options ?? throw new
ArgumentNullException(nameof(options));

    private AIAgent CreateAgent(string systemPrompt)
    {
        return new AzureOpenAIclient(
            new Uri(_options.Endpoint),
            new ApiKeyCredential(_options.ApiKey))
            .GetChatClient(_options.Deployment)
            .CreateAIagent(instructions: systemPrompt)
            .AsBuilder()
            .UseOpenTelemetry(sourceName: "agent-framework-source")
            .Build();
    }
}

```

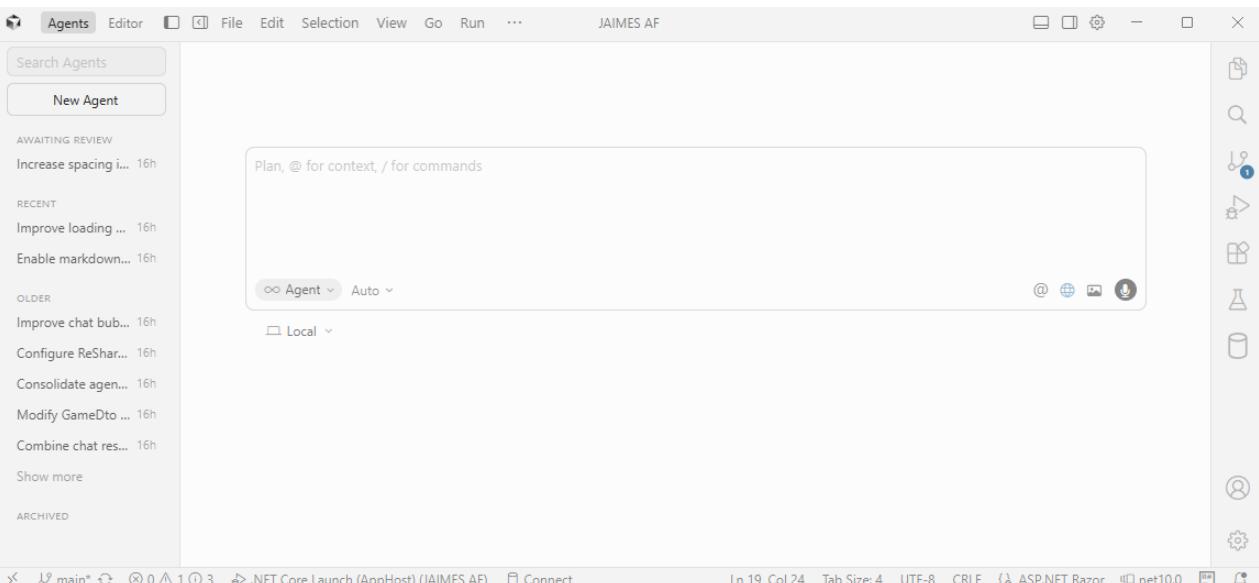


Tokens

<https://santoshkanumuri.github.io/token-visualizer/>

Agent Mode vs Editor Mode

AGENT MODE



EDITOR MODE

The screenshot shows the Editor Mode interface. The top navigation bar includes tabs for 'Agents' and 'Editor'. The main area is a code editor with the file 'ChatService.cs' open. The code is as follows:

```
JAIMES AF.Aagents > Services > ChatService.cs - C# by ReSharper > ChatService
12
13 namespace MattEland.Jaimes.Agents.Services;
14
15 3 references
16 public class ChatService(ChatOptions options, ILogger<ChatService> logger)
17 {
18     3 references
19     private readonly ChatOptions _options = options ?? throw new Argument
20     2 references
21     private AIAGent CreateAgent(string systemPrompt)
22     {
23         return new AzureOpenAIClient(
24             new Uri(_options.Endpoint),
25             new ApiKeyCredential(_options.ApiKey))
26             .GetChatClient(_options.Deployment) // ChatClient
27             .CreateAIAGent(Instructions: systemPrompt) // ChatClientAgent
28             .AsBuilder()
29             .UseOpenTelemetry(sourceName: "agent-framework-source") // AI
30             .Build(); //
```

To the right of the code editor is a 'New Chat' sidebar with the message: "Please refactor this code to increase readability!". Below the code editor, there are several status messages and a 'View All' link.

Multiple Agents in Parallel

Can work locally or in cloud

Locally you can share the same files or use git worktrees

Get Started

Models

Cursor supports all frontier coding models from all major model providers.

Name	Default Context	Max Mode	Capabilities
A\ Claude 4.1 Opus	-	200k	∞ 🎨 ✎
A\ Claude 4.5 Sonnet ⓘ	200k	1M	∞ 🎨 ✎
镝 Composer1	200k	-	∞ ✎
G Gemini 2.5 Pro	200k	1M	∞ 🎨 ✎
🧠 GPT-5.1 ⓘ	272k	-	∞ 🎨 ✎
🧠 GPT-5.1 Codex ⓘ	272k	-	∞ 🎨 ✎
X\ Grok Code	256k	-	∞ 🎨

Show more models

Models

Differences

Effectiveness

Speed

Cost

Context Window

Capabilities

Thinking

Images

Tools

<https://cursor.com/docs/models>

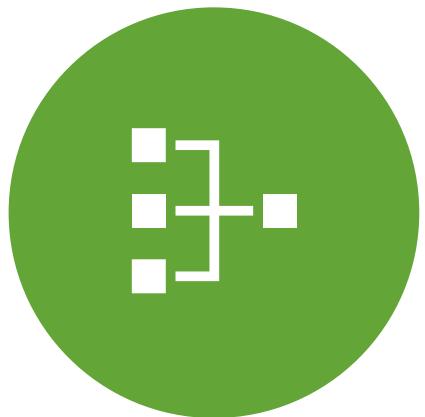
Prompts

Write a method that can take in a string representing a name and return the first, last, and middle name. Create unit tests for this as well.

Write and test a method that takes in a string parameter called name representing the user's full name. Return an object representing the different parts of the name.

For example, "Bruce Wayne" would return an object with a FirstName of "Bruce" a LastName of "Wayne", and an empty string for the MiddleName.

Resources



SYSTEMS ARCHITECTURE
DOCUMENTATION

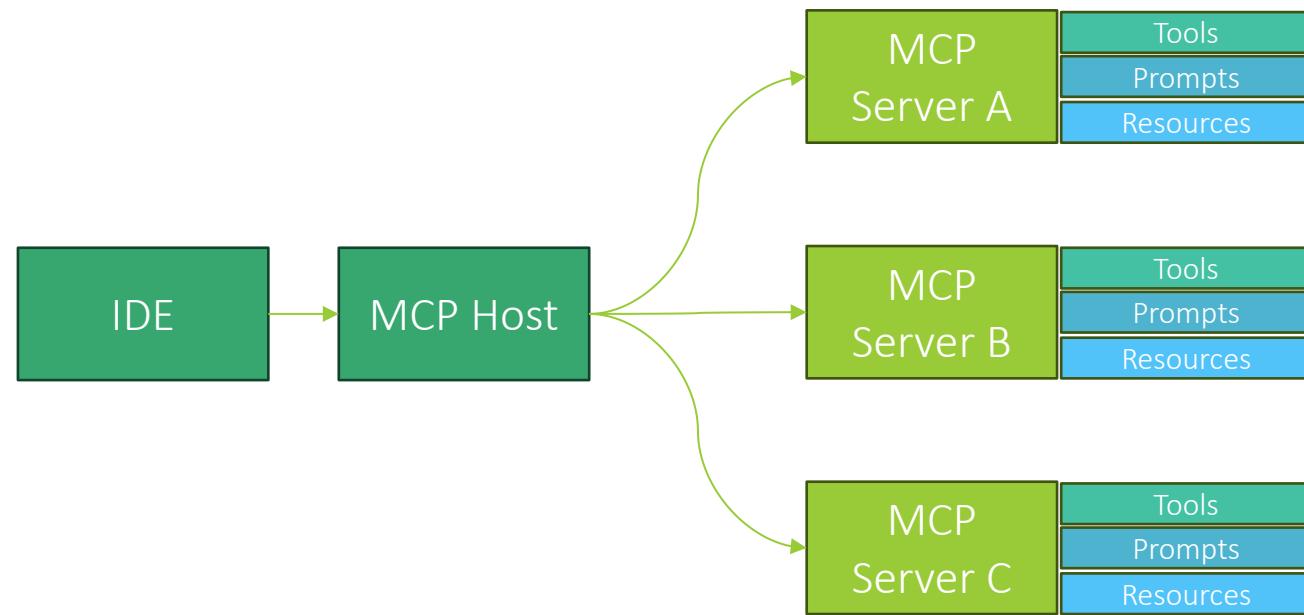


CODE STANDARDS AND
REVIEW PROCESS



FEATURE
SPECIFICATIONS

Model Context Protocol (MCP)



+
•
○

Tooling

AI Developer Productivity Tools and Cursor

Cursor 2.1

What's new in Cursor?

Composer Model

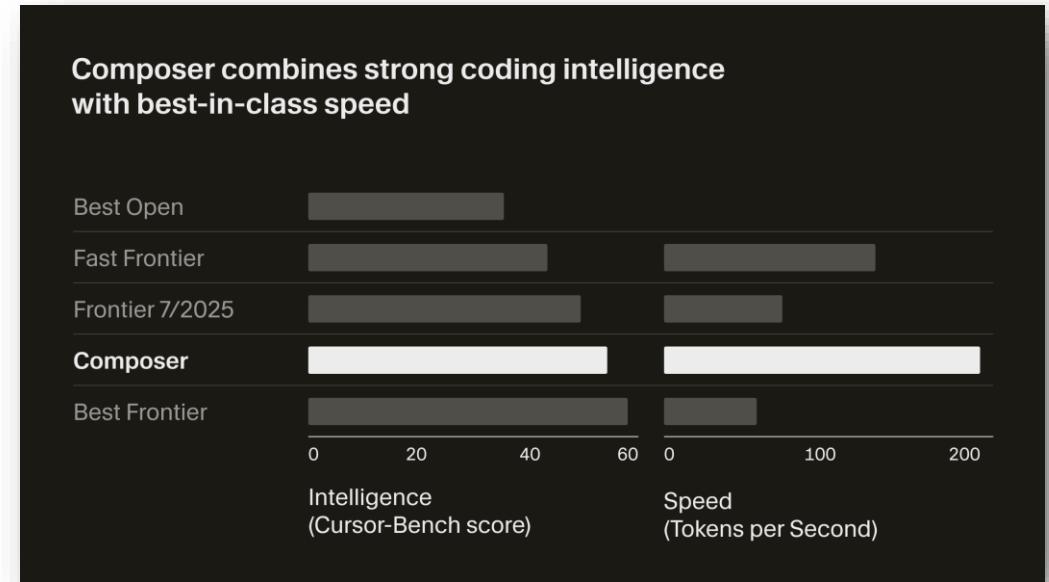
High speed

High quality output

Agent view

Optimized UI for multiple agents

Good for agent-focused workflows



Cursor vs Copilot

CURSOR

- Separate IDE (forked from VS Code)
- Can run in cloud if with per-use billing
- Heavy focus on context
- Has Enterprise editions
- Support for MCP, Agent, Plan, and Ask
- Supports many models
- Generally feels faster

COPilot

- VS Code Extension
- GitHub Cloud Runners
- Heavy focus on GitHub Integration
- Focus on Enterprise-level Features
- Support for MCP, Agents, and Ask
- Supports many models

Other IDEs like Cursor

VS Code with
Extensions

Claude Code

Windsurf

Zed Editor

TabbyML

Void

Tabnine

Business Plans

Teams

\$32/user/mo.

Everything in Pro, plus:

- ✓ Centralized team billing
- ✓ Usage analytics and reporting
- ✓ Org-wide privacy mode controls
- ✓ Role-based access control
- ✓ SAML/OIDC SSO

[Get Teams](#)

Enterprise

Custom

Everything in Teams, plus:

- ✓ Pooled usage
- ✓ Invoice/PO billing
- ✓ SCIM seat management
- ✓ AI code tracking API and audit logs
- ✓ Granular admin and model controls
- ✓ Priority support and account management

[Contact Sales](#)

Cursor Editions

Individual Plans

Hobby

Free

Includes:

- ✓ One-week Pro trial
- ✓ Limited Agent requests
- ✓ Limited Tab completions

[Download](#)

Pro

\$16/mo.

Everything in Hobby, plus:

- ✓ Extended limits on Agent
- ✓ Unlimited Tab completions
- ✓ Background Agents
- ✓ Maximum context windows

[Get Pro](#)

Pro+

Recommended

\$60/mo.

Everything in Pro, plus:

- ✓ 3x usage on all OpenAI, Claude, Gemini models

[Get Pro+](#)

Ultra

\$200/mo.

Everything in Pro, plus:

- ✓ 20x usage on all OpenAI, Claude, Gemini models
- ✓ Priority access to new features

[Get Ultra](#)

Cursor Editions

Model Selection

M matt.eland@gmail.com
Free Plan

Search settings Ctrl+F

General

Agents

Tab

Models

Cloud Agents

Tools & MCP

Rules, Memories, Commands

Indexing & Docs

Network

Beta

Docs 

Models

Add or search model 

Composer 1 

Sonnet 4.5 

GPT-5 Codex 

GPT-5 

Haiku 4.5 

Grok Code 

Gemini 2.5 Flash 

Sonnet 4.5 

GPT-5 Codex High 

GPT-5 Codex Fast 

[View All Models](#)

▼ API Keys 

OpenAI API Key 

You can put in [your OpenAI key](#) to use OpenAI models at cost.

Enter your OpenAI API Key

Override OpenAI Base URL 

Change the base URL for OpenAI API requests.

M matt.eland@gmail.com
Free Plan

Search settings Ctrl+F

General

Agents

Tab

Models

Cloud Agents

Tools & MCP

Rules, Memories, Commands

Indexing & Docs

Network

Beta

Docs

Indexing & Docs

Codebase

Codebase Indexing ?

Embed codebase for improved contextual understanding and knowledge. Embeddings and metadata are stored in the [cloud](#), but all code is stored locally.

100%

146 files

Sync

Delete Index

Index New Folders

Automatically index any new folders with fewer than 50,000 files



Ignore Files in .cursorignore

Files to exclude from indexing in addition to .gitignore. [View included files](#).

Edit

Docs

Crawl and index custom resources and developer docs

+ Add Doc

No Docs Added

Add documentation to use as context. You can also use @Add in Chat or while editing to add a doc.

Add Doc

Document Indexing in Cursor

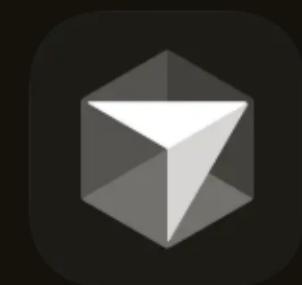
Runs as code is updated

Enables code searching

Enables agents asking questions of the codebase via RAG

Cursor Setup

Installing and Configuring Cursor



Download Cursor

Available for macOS, Windows, and Linux.

[Download for Windows ↓](#)

1.7

Latest

^



Mac (ARM64)



Mac (x64)



Mac Universal



Windows (x64) (System)



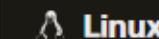
Windows (x64) (User)



Windows (ARM64) (System)



Windows (ARM64) (User)



Linux .deb (ARM64)



Linux .deb (x64)



Linux RPM (ARM64)



Linux RPM (x64)





Welcome to Cursor

The AI Code Editor

Sign Up

Log In

Skip and continue



Sign in

Email

Your email address

Continue

OR

Continue with Google

Continue with GitHub

Continue with Apple

Don't have an account? [Sign up](#)

Cursor and Code Sharing

Data Sharing

Help improve Cursor for everyone

By default, Cursor learns from code to help you get the best experience possible.

- **You're in control.** Turn off anytime in Settings → Privacy.
- **Data sharing is off the first day.** After one day of use, Cursor stores and learns from your prompts, codebase, edit history, and other usage data to improve the product.

More on [Privacy Policy](#) and [Security](#)

I'm fine with Cursor learning from my code or I'll turn it off in Settings

[Continue](#)



Disabling Code Sharing

Cursor Settings X

matt.eland@gm...
Free Plan

Search settings Ctrl+F

General Chat Tab Models

Background Age... Tools & MCP

Rules & Memor... Indexing & Docs Network Beta

Docs

Configure font, formatting, minimap and more Open

Keyboard Shortcuts Configure keyboard shortcuts Open

Import Settings from VS Code Import settings, extensions, and keybindings from VS Code Import

Default Layout The default layout for new windows Editor

Reset "Don't Ask Again" Dialogs See warnings and tips that you've hidden Show

Privacy

✓ Data Sharing Enabled Your codebase, prompts, edits and other usage data will be stored and trained on by Cursor to improve the product Share Data

Data Sharing is paused for you will activate in 24 hours.

Notifications

System Notifications Show system notifications when or needs attention

Privacy Mode (Legacy) No training and no storage. Background Agent and other features that require code storage will be disabled.

System Tray Icon Show Cursor in system tray

Completion Sound Play a sound when Agent finishes responding



New Chat

@Add Context Plan, search, build anything

Ctrl+I Auto

Past Chats New Chat

Other Helpful Settings

The screenshot shows the Code Agent settings interface. On the left, a sidebar lists various settings categories: General, Agents, Tab, Models, Cloud Agents, Tools & MCP, Rules, Memories, Commands, Indexing & Docs, Network, Beta, and Docs. A large green arrow points from the 'General' section towards the main content area.

General

- Manage Account [Open](#)
- Upgrade to Ultra [Upgrade](#)

Preferences

- Default Layout [Agent](#) [Editor](#)
- Editor Settings [Open](#)
- Keyboard Shortcuts [Open](#)
- Import Settings from VS Code [Import](#)
- Reset "Don't Ask Again" Dialogs [Show](#)

Notifications

- System Notifications [Toggle](#)
- System Tray Icon [Toggle](#)
- Completion Sound [Toggle](#)

Privacy

- Privacy Mode [Privacy Mode](#)

Activity Bar

User Workspace

- Workbench (6)
 - Appearance (4)
 - Zen Mode (1)
- Features (3)
 - Testing (1)
 - Source Control (1)
 - Terminal (1)

Workbench > Activity Bar: Icon Click Behavior

Controls the behavior of clicking an Activity Bar icon in the workbench. This value is ignored when Workbench > Activity Bar: Location is not set to default.

toggle

Workbench > Activity Bar: Location

Controls the location of the Activity Bar relative to the Primary and Secondary Side Bars.

default

Workbench > Activity Bar: Orientation

Controls the orientation of the activity bar in the workbench. The vertical orientation is no longer maintained. Quit and reopen for changes to take effect.

vertical

Zen Mode: Hide Activity Bar

✓ Controls whether turning on Zen Mode also hides the activity bar either at the left or right of the workbench.

Workbench > Secondary Side Bar: Show Labels

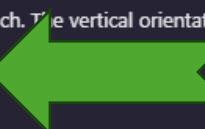
✓ Controls whether activity items in the secondary sidebar title are shown as label or icon. This setting only has an effect when Workbench > Activity Bar: Location is not set to top.

Workbench > Side Bar: Location

Controls the location of the primary side bar and activity bar. They can either show on the left or right of the workbench. The secondary side bar will show on the opposite side of the workbench.

left

Testing: Count Badge



If you want a VS Code-like Activity Bar...
 User Settings > Workbench > Activity Bar > Orientation

API Key Setup

▼ API Keys

OpenAI API Key

You can put in [your OpenAI key](#) to use OpenAI models at cost.



Enter your OpenAI API Key

Override OpenAI Base URL

Change the base URL for OpenAI API requests.



Anthropic API Key

You can put in [your Anthropic key](#) to use Claude at cost. When enabled, this key will be used for all models beginning with "claude-".

Enter your Anthropic API Key

Google API Key

You can put in [your Google AI Studio key](#) to use Google models at-cost.

Enter your Google AI Studio API Key

Azure OpenAI

Configure Azure OpenAI to use OpenAI models through your Azure account.



Base URL

e.g. my-resource.openai.azure.com

Deployment Name

e.g. gpt-35-turbo

API Key

Enter your Azure OpenAI API Key

AWS Bedrock

Configure AWS Bedrock to use Anthropic Claude models through your AWS account. Cursor Enterprise teams can configure IAM roles to access Bedrock without any Access Keys.



Access Key ID

AWS Access Key ID

Secret Access Key

AWS Secret Access Key

Java-Specific Configuration

Extension Pack for Java



vscjava [microsoft.com](#) | ⚡ 1,800,000 | ★★★★★(1)
Popular extensions for Java development that provides Java IntelliSense, debugging, testing,...

Auto Update 

[DETAILS](#) [FEATURES](#) [CHANGELOG](#)

Extension Pack (7)



Language Support for Java(TM) by Red Hat
Java Linting, Intellisense, formatting, refactor...
Red Hat Installing



Debugger for Java
A lightweight Java debugger for Visual Studio...
vscjava Installing



Project Manager for Java
Manage Java projects in Visual Studio Code
vscjava Installing

Extension Pack for Java

Extension Pack for Java is a collection of popular extensions that can help write, test and debug Java applications in Visual Studio Code. Check out [Java in VS Code](#) to get started.

Extensions Included

By installing Extension Pack for Java, the following extensions are installed:

Marketplace

Identifier	vscjava.vscode- java-pack
Version	0.30.4
Published	2021-09-03, 00:30:16
Last Released	2025-10-26, 23:51:11

Categories

Programming Languages
 Linters Debuggers
 Formatters Snippets
 Extension Packs

Resources

[Marketplace](#)
[Repository](#)
[License](#)

Setting up Java for Cursor

Java Installation

Before setting up Cursor itself, you will need Java installed on your machine.

⚠ Cursor does not ship with a Java compiler, so you need to install a JDK if you haven't already.

- **Windows:** Download and install a JDK (e.g., OpenJDK, Oracle JDK, Microsoft Build of OpenJDK). Set `JAVA_HOME` and add `JAVA_HOME\bin` to your PATH.
- **macOS:** Install via Homebrew (`brew install openjdk`) or download an installer. Ensure `JAVA_HOME` points to the installed JDK.
- **Linux:** Use your package manager (`sudo apt install openjdk-17-jdk` or equivalent) or install via SDKMAN.

To check installation, run:

```
1 java -version  
2 javac -version
```

ⓘ If Cursor does not detect your JDK, configure it manually in `settings.json`. Then, restart Cursor to apply changes.

```
1 {  
2   "java.jdt.ls.java.home": "/path/to/jdk",  
3   "java.configuration.runtimes": [  
4     {  
5       "name": "JavaSE-17",  
6       "path": "/path/to/jdk-17",  
7       "default": true  
8     }  
9   ]  
10 }
```

Java Troubleshooting

<https://cursor.com/docs/configuration/languages/java>

Setup & Troubleshooting

We'll do our rounds and make sure you're set up and ready to go

Feedback & Contact Form



+

•

○

<https://forms.gle/SrTzVzFnsDz4kiA18>

Break

Time	Content
8 – 9:15 AM	Introductions, Terminology, and Setup Verification
9:30 – 10:30 AM	Tooling Discussion and Demonstration
10:45 – 11:30 AM	Cursor for Greenfield Development
11:30 AM – 12:30 PM	Catered Lunch
12:30 – 1:15 PM	Context Discussion and Sample App Exploration
1:15 – 2 PM	Cursor for Brownfield Development
2:15 – 3:45 PM	Advanced AI Integration
4 – 5 PM	Final discussions and topics of interest

•

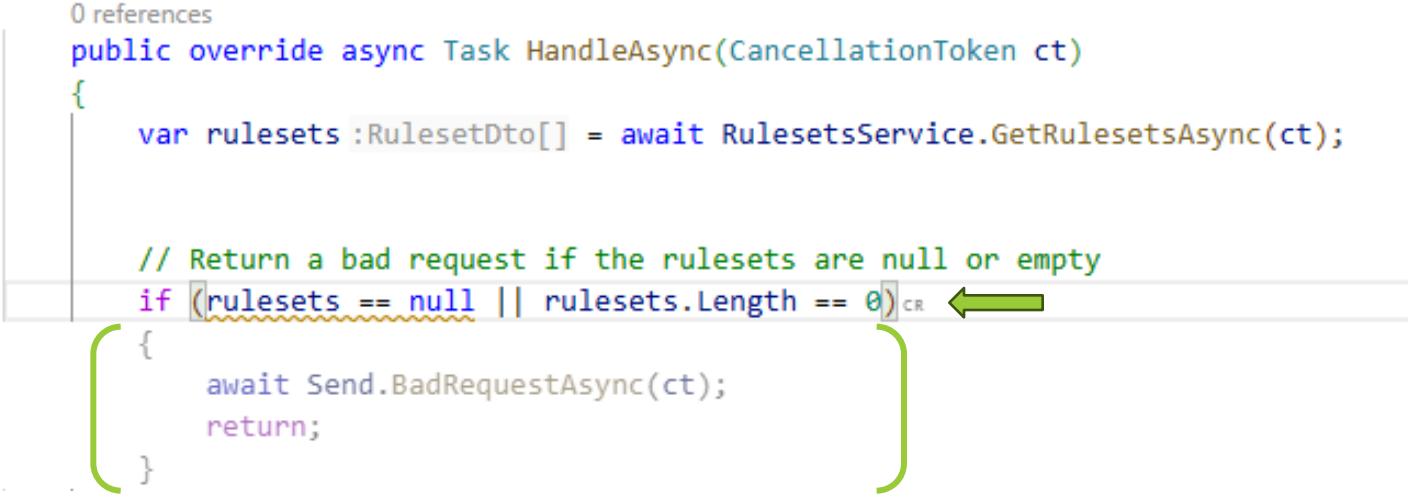
+
•
○

Cursor Basics

Getting oriented in the Cursor IDE

Tab Completions

```
0 references
20 public override async Task HandleAsync(CancellationToken ct)
21 {
22     var rulesets :RulesetDto[] = await RulesetsService.GetRulesetsAsync(ct);
23
24
25     // Return a bad request if the rulesets are null or empty
26     if [rulesets == null || rulesets.Length == 0] CR ←
27     {
28         await Send.BadRequestAsync(ct);
29         return;
30     }
31 }
```



Sidebar Chat (Ctrl+L)

The screenshot shows the Visual Studio Code interface with the Sidebar Chat feature enabled. The main editor area displays the `Program.cs` file for the project `JAIMES-AF`. The sidebar on the right contains a chat window titled "New Chat" with the message: "Refactor this code to reduce the Cyclomatic Complexity and improve the readability". Below the chat window are sections for "Past Chats" and "Recent Activity". The bottom of the screen shows the standard VS Code navigation bar with tabs like "Problems", "Output", "Debug Console", and "Terminal".

```
You, 39 minutes ago | 3 authors (Matt Eland and others)
1  using FastEndpoints;
2  using FastEndpoints.Swagger;
3  using MattEland.Jaimes.Repositories;
4  using MattEland.Jaimes.ServiceDefaults;
5  using System.Diagnostics;
6  using MattEland.Jaimes.ApiService.Helpers;
7  using MattEland.Jaimes.ServiceDefinitions.Services;
8  using MattEland.Jaimes.Services;
9
10 namespace MattEland.Jaimes.ApiService;
11
12 3 references | Matt Eland, last week | 3 authors (Matt Eland and others)
12 public class Program
13 {
14     0 references
14     public static async Task Main(string[] args)
15     {
16         WebApplicationBuilder builder = WebApplication.CreateBuilder(args);
17
18         // Add service defaults & Aspire client integrations.
19         builder.AddServiceDefaults();
20
21         // Add services to the container.
22         builder.Services.AddProblemDetails();
23
24         // Learn more about configuring OpenAPI at https://aka.ms/aspnet/openapi
25         builder.Services.AddOpenApi();
26
27         // Add Swagger services
28         builder.Services.AddEndpointsApiExplorer();
29
30         builder.Services.AddFastEndpoints().SwaggerDocument();
31
32         // Register a shared ActivitySource instance with the same name used by OpenTelemetry
33         builder.Services.AddSingleton(new ActivitySource(builder.Environment.ApplicationName ?? "Jaimes.ApiService"));
34
35         // Configure ChatOptions from configuration and register instance for DI
36         ChatOptions chatOptions = builder.Configuration.GetSection(key: "ChatService").Get<ChatOptions>() ?? throw new Invalid
37         builder.Services.AddSingleton(chatOptions);
38
39         // Add Jaimes repositories and services
40         builder.Services.AddJaimesRepositories(builder.Configuration);
```

Past Chats ▾
Fixing incorrect EF migrations
Application not showing aspire dashboard
Configure .NET Aspire project startup

Ctrl+K to generate command

0 1 7 15 1 C#: Aspire Debug (JAIMES-AF) Combo: 1 Building... 0 Matt Eland, 2 weeks ago Cursor Tab 0 Matt Eland (1 week ago) Ln 63, Col 2 Spaces: 4 UTF-8 CRLF { C# net10.0

Specifying Context @

A screenshot of a code editor's interface, likely Visual Studio Code, showing a context menu open over a file named "Program.cs". The menu includes options for navigating to files like "Endpoints" and "AppHost.cs", as well as browser-related options like "Branch (Diff with Main)" and "Browser".

New Chat + ⏱ ...

Is there anything unusual about the files in my `Endpoints` directory or how they're referenced in `Program.cs` by

Ask Auto

Local

Add files, folders, docs...

- `AppHost.cs` JAAMES AF.AppHost
- `launchSettings.json` ...pHost/Properties
- `launch.json` .vscode
- Files & Folders >
- Docs >
- Terminals >
- Branch (Diff with Main)
- Browser
- What's New

gram.cs 1 C# ListGamesEndpoint.cs C# ListRulesetsEndpoint.cs @ Games.razor @ NewGame.razor M X % plan.plan.md % add-c

JAIMES AF.Web > Components > Pages > @ NewGame.razor

You, 1 second ago | 1 author (You) @page "/games/new" You, 4 minutes ago • Enables game deletion ...

1 @rendermode InteractiveServer
2
3 @pageTitle Create New Game</pageTitle>
4
5 <MudPaper Class="pa-4">
6 <MudText Typo="Typo.h4">Create New Game</MudText>
7 <MudContainer MaxWidth="MaxWidth.Medium" Class="mt-4">
8 <MudPaper Class="pa-4">
9 <MudText Typo="Typo.h4" Class="mb-4">Create New Game</MudText>
10
11 @if (isLoading)
12 {
13 <MudProgressCircular Indeterminate="true" Class="my-4" />
14 }
15 else if (!string.IsNullOrEmpty(errorMessage))
16 {
17 <MudAlert Severity="Severity.Error" Class="mb-4">@errorMessage</MudAlert>
18 }
19
20 @if (!isLoading)
21 {
22 <MudForm>
23 <MudSelect @bind-Value="selectedScenarioId"
24 Label="Scenario"
25 Variant="Variant.Outlined"
26 Class="mb-4"
27 Disabled="isCreating">
28 @foreach (var scenario in scenarios)
29 {
30 <option value="@scenario.Id">@scenario.Name

^ 1 of 3 Undo Ctrl+N Keep Ctrl+Shift+Y

Implement new game page functionality

Implement the New Game page in such a way that it allows the user to specify the details necessary to create a new game. Once the user clicks a Create button the page should call to the [C# NewGameEndpoint.cs](#) and then navigate to [@ GameDetails.razor](#) to display the details of the new game once created.

∞ Agent Auto @ 🌐 🖥 ⬆

@ NewGame.razor +59 -4

@using MattEland.Jaimes.ServiceDefinitions.Responses
@rendermode InteractiveServer
<pageTitle>Create New Game</pageTitle>

Checking for linting errors:
Found 1 error in 1 file
Removing the unnecessary using directive:
@ NewGame.razor -1

@using MattEland.Jaimes.ServiceDefinitions.Responses
@rendermode InteractiveServer

No linter errors found
Checking how buttons are used in other components to ensure consistency:
Grepped MudButton.*OnClick in JAIMES AF.Web
Updating the button to match the pattern used in other components:
@ NewGame.razor +2 -2
 <MudProgressCircular Class="ms-n1" Size="Size.Small" Indeterminate="true" />

Chat: Agent

Cursor Allowlist

M matt.eland@gmail.com
Pro Plan

Search settings Ctrl+F

- General
- Agents**
- Tab
- Models

- Cloud Agents
- Tools & MCP

- Rules, Memories, Commands
- Indexing & Docs
- Network
- >> Beta

- Docs

Auto-Run

Auto-Run Mode

Choose how Agent runs tools like command execution, MCP, and file writes.

Use Allowlist ▾

Command Allowlist

Commands that can run automatically

npm test X cd X mvn X

MCP Allowlist

MCP tools that can run automatically. Format: 'server:tool', 'server:*' for all tools from a server, '*:tool' for a tool from any server, or '*' for all tools from all servers

e.g., filesystem:read_file

Browser Protection

Prevent Agent from automatically running Browser tools

File-Deletion Protection

Prevent Agent from deleting files automatically

Dotfile Protection

Prevent Agent from modifying dot files like .gitignore automatically

External-File Protection

Prevent Agent from creating or modifying files outside of the workspace automatically

point.cs • C# ListRulesetsEndpoint.cs M @ Games.razor 1, M plan.plan.md add-delete-games-and-new-game-link.plan.md X JAIMES AF.ApiS... Implement game deletion and new game link + ⏱ ...

add-delete-games-and-new-game-link.plan.md

Add Delete Games and New Game Link

Backend Changes

1. Service Layer

- File: JAIMES AF.ServiceDefinitions/Services/IGameService.cs
- Add method: Task DeleteGameAsync(Guid gameId, CancellationToken cancellationToken = default)
- File: JAIMES AF.Services/Services/GameService.cs
- Implement DeleteGameAsync method
- Find game by ID, validate it exists
- Delete game (Messages will cascade delete automatically per EF Core configuration)

2. API Endpoint

- New File: JAIMES AF.Apis/Endpoints/DeleteGameEndpoint.cs
- FastEndpoints endpoint using the SeqNoR's pattern
- Route: DELETE /games/{gameId:guid}
- Call GameService.DeleteGameAsync
- Return 204 No Content on success, 404 if game not found

3. Testing

- New File: JAIMES AF.Tests/Endpoints/DeleteGameEndpointTests.cs
- Test successful deletion
- Test 404 when game doesn't exist
- Follow pattern from ListGamesEndpointTests.cs

Frontend Changes

1. Games List Page

- File: JAIMES AF.Web/Components/Pages/Games.razor

Add "New Game" button/link at top (always visible before the table)

Chat: Plan

Determine a scope of changes for adding the ability to delete existing games and a link to a new games page.

This new game link should be present in all cases and should also be linked to from the (no games) message. It should navigate the browser to /games/new and should not immediately create a game or call the API.

The delete game link should occur at the right edge of each row, use an appropriate icon and color, and require the user to confirm before deletion. Actual deletion should occur via the API using a DELETE verb and a new endpoint.

Plan Auto ↻

Searched What MudBlazor components are used for icons and buttons in the web UI? (J...
Read GameDetails.razor
Grepped MudDialog|Dialog|Confirm in JAIMES AF.Web
Read JaimesDbContext.cs

Checking the Message entity structure and reviewing an endpoint test to understand the testing pattern:

Explored 1 directory 2 files
Creating the plan:

add-delete-games-and-new-game-link.plan.md

Add Delete Games and New Game Link

Add ability to delete games with confirmation dialog, and add a "New Game" link that navigates to /games/new without creating a game. The delete button appears at the right edge of each game row, and the new game link is always visible including in the empty state.

6 To-dos

- Add DeleteGameAsync method to IGameService interface and implement in GameService
- Create DeleteGameEndpoint with DELETE /games/{gameId:guid} route
- Add tests for DeleteGameEndpoint

... 3 more

View Plan Auto ↻ Build Ctrl+d ↻

Chat: Ask

Fixing incorrect EF migrations

I think the ApiService has an incorrect set of migrations. When the application starts I get this exception: "There is already an object named 'Rulesets' in the database.".

I think this comes from a bad series of EF migrations. What should I do to resolve this?

Ask Auto

Investigating the EF migrations issue. Checking migrations and how they're applied.

Explored 1 directory 3 files

Checking the migrations for duplicate table creation:

Explored 1 file 2 searches

Checking other migrations for duplicate table creation:

Explored 2 files 1 search

Checking the migration history and the model snapshot:

Explored 2 files >

The error "There is already an object named 'Rulesets' in the database" usually means the `Rulesets` table exists, but the `__EFMigrationsHistory` table is missing or incomplete, so EF tries to re-run `InitialCreate`.

Diagnosis

Your migrations look correct:

- `20251029024204_InitialCreate.cs` creates the `Rulesets` table
- Later migrations only modify data or add columns; none recreate the table

The issue is likely a mismatch between the database state and the migration history.

Enhance game visuals for reversi

Create PR

Review

28m · +381 -67 · Auto



Okay, this is awesome, but I have a couple of requests. First, can you make the preview dots the same size as the pieces?

Secondly, the orange text when the AI is thinking is hard to read on a white background.



- Added hover tracking with `mouseEntered` and `mouseExited` listeners
- Modified `BoardCell` to support a highlighted state for capture preview
- Updated piece rendering to show purple gradient when highlighted

All linting errors are resolved, and the code is ready to use. The game now provides clearer visual feedback for valid moves and potential captures.



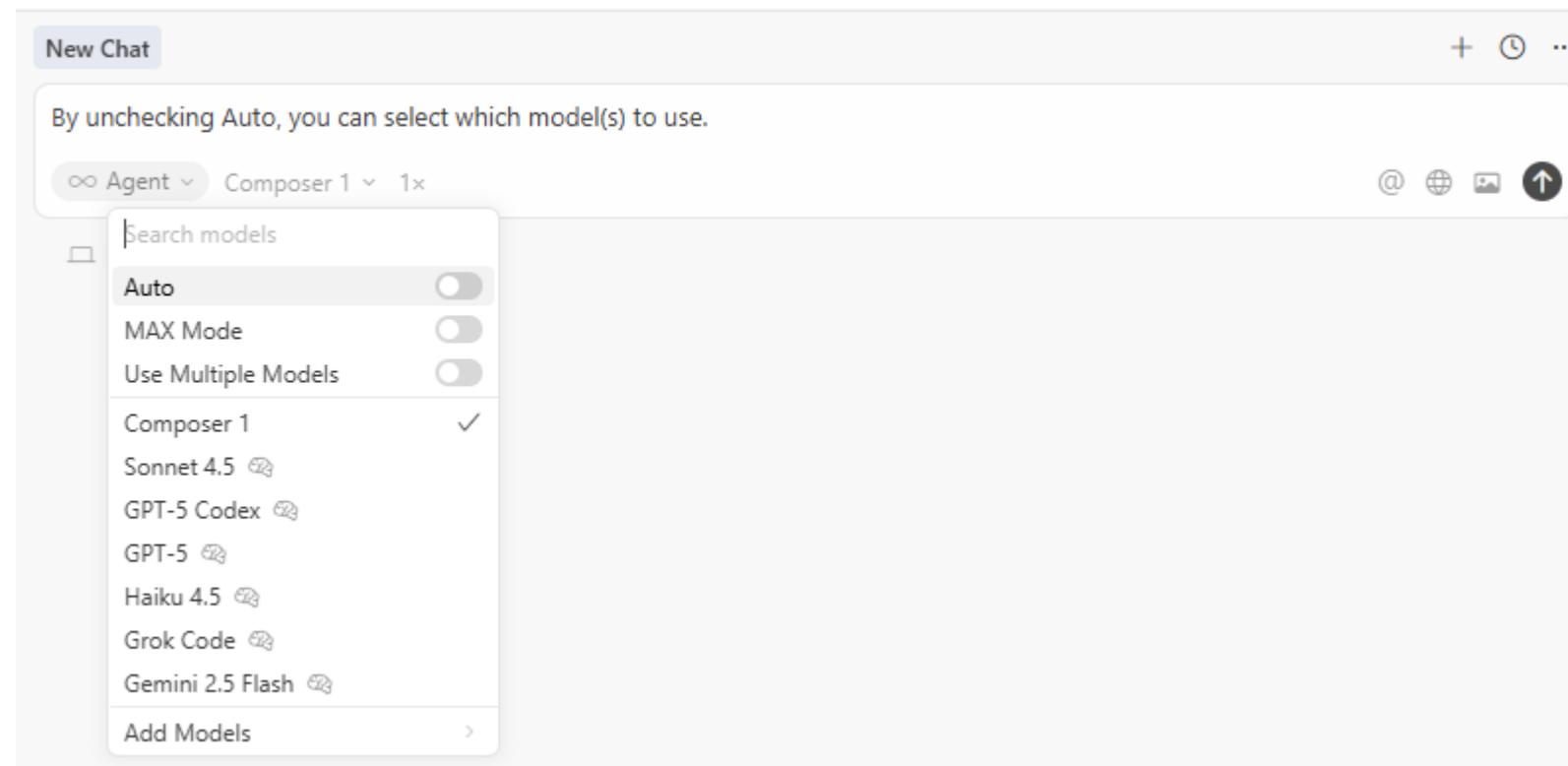
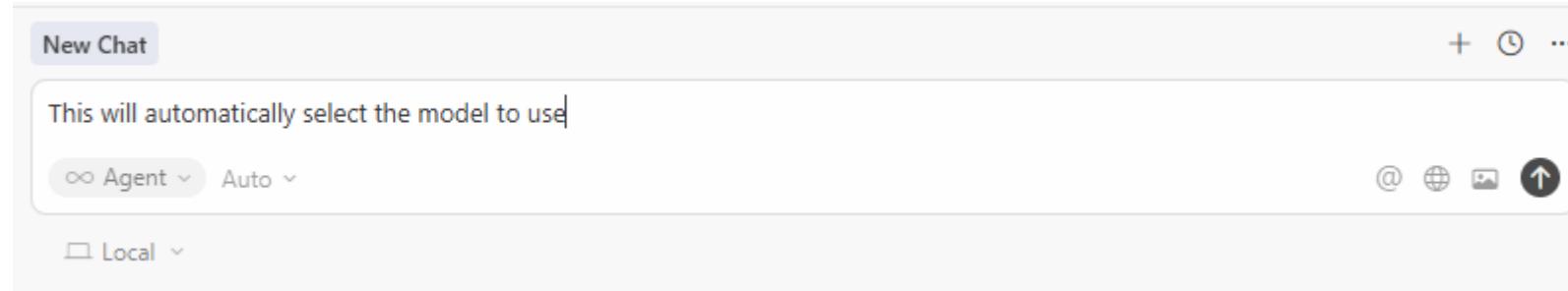
This is great, but can we make the potential move ones not have a gradient, just have a large circle?

∞ Agent ▾ Auto ▾



Chat with Visuals

Model Selection



Model Selection

M matteland@gmail.com
Free Plan

Search settings Ctrl+F

- General
- Agents
- Tab
- Models**
- Cloud Agents
- Tools & MCP
- Rules, Memories, Commands
- Indexing & Docs
- Network
- Beta

Docs

Models

Add or search model

- Composer 1
- Sonnet 4.5
- GPT-5 Codex
- GPT-5
- Haiku 4.5
- Grok Code
- Gemini 2.5 Flash
- Sonnet 4.5
- GPT-5 Codex High
- GPT-5 Codex Fast

[View All Models](#)

▼ API Keys

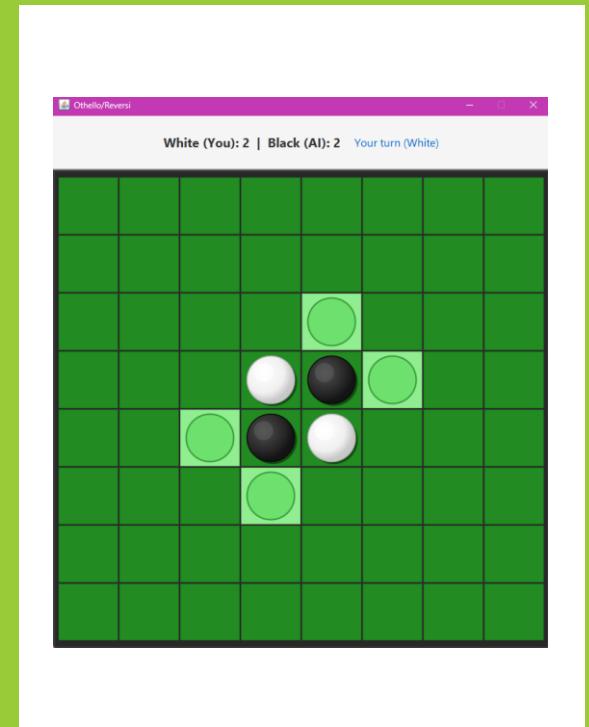
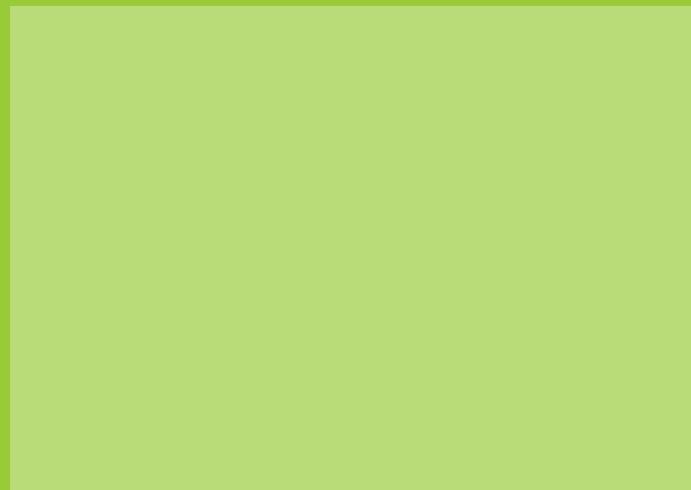
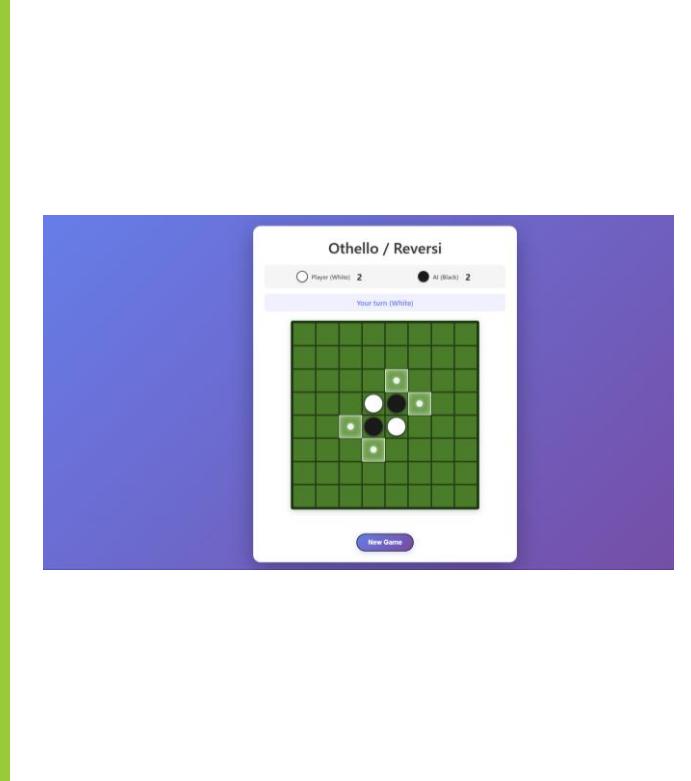
OpenAI API Key
You can put in your OpenAI key to use OpenAI models at cost.

Enter your OpenAI API Key

Override OpenAI Base URL
Change the base URL for OpenAI API requests.

Demo

Building a Greenfield Application with Cursor



Othello.js Prompt

I want to create a browser-based game that runs on Vanilla JavaScript.

This game should implement the classic game of Othello / Reversi where a player controls the white pieces and can place a piece on the game board, then an AI opponent thinks, then places a black piece on the board. These moves can potentially flip pieces that are surrounded by the color that was placed on horizontal, vertical, or diagonal axes.

The game should display the current score in terms of piece count at all times and should end when all spaces are occupied, announcing which player won.

Tests are important and the game should look polished and professional.

Please scaffold and create this project and its tests.

Othello.js Prompt

I want to create a browser-based Othello/Reversi game using vanilla JavaScript (no frameworks). The game should:

- Run directly in a browser by opening index.html (no build step required)
- Work with `<script>` tags, not Node.js modules
- Include tests that can run with Jest/Node.js separately
- Have clearly visible gridlines separating each cell
- Show valid moves with clear visual indicators
- Display pieces prominently on the board

Othello.js Prompt

Create a browser-based Othello game with:

- Clear visual gridlines so each cell is distinctly visible
- Prominent display of game pieces (white and black circles)
- Highlighted valid moves that are easy to see
- Real-time score display
- Smooth gameplay with click handlers that work after each AI move
- Professional, polished UI with good contrast and visibility

Othello.js Prompt

...and please ensure:

- The game is immediately playable when opening index.html in a browser
- All cells have visible borders/gridlines
- Valid moves are clearly highlighted
- Click handlers work correctly after AI moves
- The code works in browser environment (not just Node.js for tests)

Othello Java 21 Prompt

I want to create an Othello/Reversi game using Java 21

The game should:

- Include a grid with clear gridlines separating each cell
- Allow the player to select which tile they want to move
- Have an AI opponent that selects a valid move with the maximum number of pieces captured
- Flip each piece as it is placed or captured
- Display the current score with the player controlling white and the AI controlling black
- Display a message at the end of the game (when no moves remain)
- Include valid Junit tests

Feedback & Contact Form



+

•

○

<https://forms.gle/SrTzVzFnsDz4kiA18>

Break

Time	Content
8 – 9:15 AM	Introductions, Terminology, and Setup Verification
9:30 – 10:30 AM	Tooling Discussion and Demonstration
10:45 – 11:30 AM	Cursor for Greenfield Development
11:30 AM – 12:30 PM	Catered Lunch
12:30 – 1:15 PM	Context Discussion and Sample App Exploration
1:15 – 2 PM	Cursor for Brownfield Development
2:15 – 3:45 PM	Advanced AI Integration
4 – 5 PM	Final discussions and topics of interest

•

Context Engineering

When you want AI to do things your way

Context Window

The page:

- Loads available scenarios and players on initialization
- Allows selection via dropdowns
- Calls the `NewGameEndpoint` when Create is clicked
- Navigates to `GameDetails.razor` with the new game ID after successful creation
- Handles errors and loading states

Follows the same patterns as `Games.razor` and `GameDetails.razor` for consistency.

This generates an exception at runtime on loading the NewGame page:

> Missing <`MudPopoverProvider` />, please add it to your layout. See <https://mudblazor.com/getting-started/installation#manual-installation>

17.1% context used
Active Rules
AGENTS.md

∞ Agent ▾ Auto ▾



Checking the layout file and adding the missing MudBlazor provider:

Explored 1 file 1 search ▾

Read `MainLayout.razor`

Searched files `**/*Layout*.razor` in JAIMES AF.Web

`MudPopoverProvider` is already in the layout. Checking the app structure and MudBlazor configuration:

Explored 2 files 1 search

Ways of Injecting Context in Cursor

Conversational
prompt

Code and
Comments

Agents.md

Rules

Memories

Team-specific
Context

User-specific
Context

Indexed
Documents

Commands

External tools /
MCP Servers

Matt Eland, 2 weeks ago | 1 author (Matt Eland)

```
1 Agents Orientation
2
3 Purpose
4
5 This file provides minimal, machine-oriented hints and common symbols to help AI assistants (Copilot, GitHub
6 Copilot, and other agents) understand the repository structure and naming conventions.
7
8 Key service interfaces and locations
9 - `IGameService` -> `JAIMES AF.Services/Services/IGameService.cs`
10 - `IScenariosService` -> `JAIMES AF.Services/Services/IScenariosService.cs`
11 - `IPlayersService` -> `JAIMES AF.Services/Services/IPlayersService.cs`
12 - `ChatService` -> `JAIMES AF.Services/Services/ChatService.cs`
13
14 Common DTOs and Mappers
15 - DTOs: `Services/Models/*.cs` (e.g., `GameDto`, `ScenarioDto`, `PlayerDto`)
16 - Mappers: `Services/Services/*Mapper.cs` (e.g., `GameMapper`, `MessageMapper`)
17
18 API endpoints
19 - `JAIMES AF.ApiService/Endpoints/*.cs` - small endpoint classes
20 - Request/Response models in `JAIMES AF.ApiService/Requests` and `Responses`
21
22 EF Core entities
23 - `JAIMES AF.Repositories/Entities/*.cs` - EF Core entities
24 - `JaimesDbContext` - in `JAIMES AF.Repositories/JaimesDbContext.cs`
25
26 Testing
27 - Tests are in `JAIMES AF.Tests/` with folders for endpoints, repositories, and services.
28
29 Development notes
30 - Target: `.NET9`
31 - DI registration: `ServiceCollectionExtensions.cs` in each layer registers services and repositories.
32 - Keep endpoints thin and delegate to services.
33
34 Alternate filenames agents may check
35 - `github/copilot_instructions.md`
```

Agents.md

Single context file in your repository root folder.

PHP	8	You are an expert in Java programming, Spring Boot, Spring Framework, Maven, JUnit, and related Java technologies.
JavaScript	6	
TailwindCSS	5	
Laravel	5	
C#	4	
Web Development	4	
Game Development	4	
Expo	4	
React Native	4	
Flutter	4	
Tailwind	4	
Testing	4	

Pre-built Agent Files

<https://cursor.directory>

<https://agents.md>

User Rules

Manage your custom user rules and preferences

+ Add Rule

Style request, response language, tone...



[Cancel](#) [Done](#)

I prefer avoiding use of the var keyword and using target-typed new syntax wherever possible.

...

User-Specific Context

Centralized Agent Controls

Capability	Individual Plans	Teams	Enterprise
Privacy Mode	User choice	Enforce org-wide	Enforce org-wide
<u>Team Rules</u>		Enforceable + Optional	Enforceable + Optional
<u>Hooks for Logging, Auditing, and more</u>	✓	MDM Distribution	<u>MDM & Server-side distribution</u>

Team-Specific Context

Project-Specific Context

M matt.eland@gmail.com
Pro Plan

Search settings Ctrl+F

- General
- Agents
- Tab
- Models

- Cloud Agents
- Tools & MCP

- Rules, Memories, Commands
- Indexing & Docs
- Network
- Beta

- Docs

Pattern Matched
These rules are applied when working with files that match their patterns

blazor-rules.mdc
.razor

Guidelines for dependency injection registration
dependency-injection.mdc
["**/ServiceCollectionExtensions.cs", "**/Program.cs", "**/RepositoryServiceCollectionExtensions.cs"]

Guidelines for API endpoints using FastEndpoints
endpoints.mdc
["**/Endpoints/*.cs", "**/ApiService/**/*.cs"]

Guidelines for EF Core entity classes
entities.mdc
["**/Entities/*.cs", "**/Repositories/Entities/*.cs"]

Guidelines for Kernel Memory Redis configuration and tag field management
kernel-memory-redis.mdc
["**/Program.cs", "**/ServiceCollectionExtensions.cs", "**/DocumentIndexer.cs", "**/RulesSearchService.cs"]

Guidelines for mapper classes that convert between entities and DTOs
mappers.mdc
["**/Mapping/*.cs", "**/Mappers/*.cs", "**/Services/Mapping/*.cs"]

Guidelines for EF Core migrations and database changes
migrations.mdc
["**/Migrations/*.cs", "**/Repositories/Migrations/*.cs", "**/JaimesDbContext.cs"]

Guidelines for service interfaces and implementations
service-interfaces.mdc
["**/ServiceDefinitions/Services/*.cs", "**/Services/*.cs"]

.cursor/rules/blazor-rules.mdc
.cursor/rules/**/*.razor

Guidelines for dependency injection registration
.cursor/rules/dependency-injection.mdc
.cursor/rules/["**/ServiceCollectionExtensions.cs", .cursor/rules/**/Program.cs", ".cursor/rules/**/RepositoryServiceCollectionExtensions.cs"]

Guidelines for API endpoints using FastEndpoints
.cursor/rules/endpoints.mdc
.cursor/rules/["**/Endpoints/*.cs", .cursor/rules/**/ApiService/**/*.cs"]

Guidelines for EF Core entity classes
.cursor/rules/entities.mdc
.cursor/rules/["**/Entities/*.cs", .cursor/rules/**/Repositories/Entities/*.cs"]

I'm getting started using Cursor on my codebase and want to generate an agents.md file and relevant context files for the major areas of my application. These files should include details on how to work with each area, things to watch out for, and the best way to work with these files so Cursor can be successful in the future.

∞ Agent ▾ Auto ▾

@ 🌐 🖼 ⬆

Local ▾

Shift Tab to plan

Cursor can generate Context Files

Commands

Project Commands

Commands specific to this workspace

+ Add Command

build

Verify the solution builds by running dotnet build.

runtests

Test the solution by running "dotnet test" and verifying that the solution build...

test

Write tests to support your work and ensure the tests pass by using the dotnet t...

Command Examples

/test – ensure you add relevant tests for these changes and these tests pass

/tdd – follow test-driven development (TDD) while making these changes

/docs – please update any relevant documentation

/context – please update any relevant context files to reflect our decisions here

/maintain – ensure code is simple and maintainable with low cyclomatic complexity

/dry – look for places where there is code duplication and consolidate when possible

Commands

Project Commands
Commands specific to this workspace

+ Add Command

test-command
test-command

User Commands
Your personal commands from ~/cursor/commands and ~/claude/commands

+ Add Command

No User Commands
Create personal commands to reuse frequently

Add Command

.cursor > commands > test-command.md > # test-command

```
1 # test-command
2
3 Make your answer as silly as possible.
4
```

Enhance content for GameDetails page ...

I need more content on the GameDetails.razor page. /test-command

Reviewing the GameDetails page and its code-behind to understand the structure before adding more content.

Explored 3 files 2 searches

Adding more content to the GameDetails page with a fun, silly twist.

@ GameDetails.razor +40 -1

```
<MudPaper Elevation="2" Class="pa-4 mb-4">
    <MudText Typo="Typo.h6" Class="mb-3">🎮 Game Information Card 🎮</MudText>
    <MudGrid>
        <MudItem xs="12" sm="6" md="4">
            <MudText Class="mb-1"><b>ID</b></MudText>
            <MudText Class="mb-2" Style="font-family: monospace; font-size: 1em; padding: 0 10px;">
```

Adding an empty state for when there are no messages and enhancing the message input area:

File-Specific Context

Apply to Specific Files

Matt Eland, 6 days ago | 1 author (Matt Eland)

```
1  ---
2  globs: ["**/ServiceCollectionExtensions.cs", "**/Program.cs", "**/RepositoryServiceCollectionExtensions.cs"]
3  alwaysApply: false
4  description: Guidelines for dependency injection registration
5  ---
6
7  # Dependency Injection Guidelines
8
9  When working with DI registration:
10
11 ## Registration Location
12 - Each project has its own `ServiceCollectionExtensions.cs` (or similar)
13 - Services are registered in `Program.cs` using extension methods
14 - Keep registration logic organized by layer
15
16 ## Service Lifetimes
17 - **Scoped**: Most application services (created once per request)
18 - **Singleton**: Configuration, shared services (created once for application lifetime)
19 - **Transient**: Lightweight, stateless services (created each time requested)
20
21 ## Registration Patterns
22 - Register interfaces, not concrete types (when using interfaces)
23 - Use extension methods to organize registrations
24 - Group related registrations together
25
26 ## Example Pattern
27 ```csharp
28 // ServiceCollectionExtensions.cs
29 public static class ServiceCollectionExtensions
30 {
31     public static IServiceCollection AddApplicationServices(this IServiceCollection services)
32     {
33         services.AddScoped<IGameService, GameService>();
34         services.AddScoped<IScenariosService, ScenariosService>();
35         services.AddScoped<IPlayersService, PlayersService>();
36
37         return services;
38     }
39 }
```

Directory-Specific Context

Apply to Specific Files

["**/Mapping/*.cs" x]

["**/Mappers/*.cs" x]

["**/Services/Mapping/*.cs"] x

Matt Eland, 6 days ago | 1 author (Matt Eland)

```
1 --- Matt Eland, 6 days ago • Add guidelines for various project layers incl...
2 globs: ["**/Mapping/*.cs", "**/Mappers/*.cs", "**/Services/Mapping/*.cs"]
3 alwaysApply: false
4 description: Guidelines for mapper classes that convert between entities and DTOs
5 ---
6
7 # Mapper Guidelines
8
9 When working with mapper classes:
10
11 ## Purpose
12 Mappers convert between:
13 - Repository entities (from `JAIMES AF.Repositories/Entities`)
14 - DTOs (from `MattEland.Jaimes.Domain`)
15 - Response models (from `JAIMES AF.ServiceDefinitions/Responses`)
16
17 ## Mapperly (Preferred Approach)
18
19 **Always prefer Mapperly** for mapping when possible. Mapperly generates compile-time code that is fast and type-safe.
20
21 ### When to Use Mapperly
22 - When source and target properties have matching names (or can be mapped with `[MapProperty]`)
23 - When mapping between classes with similar structures
24 - When you need collection mapping (automatically handled)
25
26 ### When NOT to Use Mapperly
27 - Generally, Mapperly can handle most cases, even complex ones, using `[UserMapping]` for custom implementations
28
29 **Note**: If a DTO is currently a record with positional parameters, consider converting it to a class to enable Mapperly maintaining manual mappers.
30
31 **Note**: For complex navigation property mappings (e.g., `Game.Ruleset.Name` → `GameDto.RulesetName`), use `[UserMapping]` while still benefiting from Mapperly's collection mapping and other features.
32
33 ### Mapperly Best Practices
34
35 1. **Use `'[Mapper]'` attribute** on static partial classes
36 2. **Use `'[MapProperty]'** for property name mismatches:
37     ```csharp
38     [MapProperty(nameof(Source.Id), nameof(Target.SourceId))]
39     ```


```

Apply Intelligently ▾

Governs usage of the C# var keyword

```
...
1 ---
2 alwaysApply: false
3 description: Governs usage of the C# var keyword
4 ---
5 The C# var keyword is only acceptable in foreach statements. For all other variable declarations, always specify the created type. Use the C# target-typed new feature when possible to reduce repetition in code.
```

Auto-Applied Context

Context Inclusion Modes



Always Apply



Apply Intelligently if agent decides to



Apply to Specific Files using patterns



Apply Manually using @ mentions

Context Recommendations

- Rules embedded in code are **easy to maintain**
- Use **example Agent files** with tweaks as needed or have Cursor draft it
- Have Cursor **generate context files** for you
 - Context for major areas** – APIs, front-end, tests, persistence layer, etc.
 - Context for key files** of interest or challenge
- When you have struggles and overcome challenges, ask **Cursor to improve** your context or agent files based on what it learned

Memories

Memories
Saves useful context across Chats

Memories
Learn your preferences automatically based on Chats 

Saved Memories
Review or remove individual Memories 

No Memories Yet
Memories will appear as you use the Agent

Memories

Memories are automatically generated rules based on your conversations in Chat. These memories are scoped to your project and maintain context across sessions.

You can manage memories from Cursor Settings → Rules.

How memories are created

1. **Sidecar observation:** Cursor uses a sidecar approach where another model observes your conversations and automatically extracts relevant memories. This happens passively in the background as you work. Background-generated memories require user approval before being saved, ensuring trust and control over what gets remembered.
2. **Tool calls:** Agent can directly create memories using tool calls when you explicitly ask it to remember something or when it notices important information that should be preserved for future sessions.

```
2025-11-11 15:45:59.565 [info] Finishing sync (hasUpdates: false, isError: false)
2025-11-11 15:45:59.609 [info] Finished indexing repository.
2025-11-11 15:45:59.609 [info] Indexing job successfully done or aborted.
2025-11-11 15:49:20.148 [error] Error getting submodules: A system error occurred (ENOENT: no such file or directory,
open 'c:\Dev\JAIMES-AF\.gitmodules')
2025-11-11 15:49:55.515 [info] Get user commits took 76.1935000005215ms for 10 commits
2025-11-11 15:50:47.509 [info] multiCodebaseIndexingJob dispose
2025-11-11 15:50:47.509 [info] Aborting indexing job.
2025-11-11 15:50:47.509 [info] Setting global status from status tracker: synced
2025-11-11 15:50:47.509 [info] Doing a startup handshake.
2025-11-11 15:50:47.517 [info] Finished computing merkle tree in 2.3660999992862344 ms.
2025-11-11 15:50:47.518 [info] Doing the initial handshake with hash:
ee8a3b5430d186213363ab47fd203d0805b3a0454cf581e41c932767da134ac8
2025-11-11 15:50:47.519 [info] Computed simhash vector of length: 256
2025-11-11 15:50:47.519 [info] Path key hash: 0b0718901b86cdff8022640a6911753a08c9a8f3a1d4727d37ac63e5aade6a30
2025-11-11 15:50:47.519 [info] Handshake start
2025-11-11 15:50:47.660 [info] Handshake timing: 142.035000000149, response: {"status": "STATUS_SUCCESS", "codebases": [{"codebaseId": "17505526-9e8b-43db-9335-77c9c68a47af", "status": "STATUS_OUT_OF_SYNC", "copyTaskHandle": ""}]}
2025-11-11 15:50:47.660 [info] Handshake result: {"status": "STATUS_SUCCESS", "codebases": [{"codebaseId": "17505526-9e8b-43db-9335-77c9c68a47af", "status": "STATUS_OUT_OF_SYNC", "copyTaskHandle": ""}]}
2025-11-11 15:50:47.660 [info] Starting fast remote sync.
2025-11-11 15:50:47.661 [info] Total num embeddable files: 146
2025-11-11 15:50:47.662 [info] Root hash: ee8a3b5430d186213363ab47fd203d0805b3a0454cf581e41c932767da134ac8
2025-11-11 15:50:47.662 [info] In the out of sync case.
2025-11-11 15:50:47.664 [info] Waiting on semaphore to be released 1
2025-11-11 15:50:47.877 [info] Waiting on semaphore to be released 1
2025-11-11 15:50:48.080 [info] Waiting on semaphore to be released 1
2025-11-11 15:50:48.283 [info] Waiting on semaphore to be released 1
2025-11-11 15:50:48.283 [info] [startSync]: sending sync merkle subtree batch of size 2
2025-11-11 15:50:48.493 [info] setting numJobsToGo to 2
2025-11-11 15:50:48.493 [info] [startSync]: numJobs: 2
2025-11-11 15:50:48.494 [info] Uploading 2 files.
2025-11-11 15:50:48.494 [info] Total number of files to embed: 146
2025-11-11 15:50:48.494 [info] Not aborted
2025-11-11 15:50:48.494 [info] Starting while loop.
2025-11-11 15:50:48.499 [info] syncing batch of 2 files and 3005 bytes
2025-11-11 15:50:50.778 [info] Completed job successfully: 17505526-9e8b-43db-9335-77c9c68a47af .\JAIMES AF.
Web\Components\Layout\MainLayout.razor
2025-11-11 15:50:50.778 [info] Completed job successfully: 17505526-9e8b-43db-9335-77c9c68a47af .\JAIMES AF.
Web\Components\Pages\NewGame.razor
2025-11-11 15:50:50.821 [info] Finishing sync (hasUpdates: true, isError: false)
2025-11-11 15:50:51.070 [info] Finished indexing repository.
2025-11-11 15:50:51.070 [info] Indexing job successfully done or aborted.
2025-11-11 15:51:51.268 [error] Error getting submodules: A system error occurred (ENOENT: no such file or directory,
```

Indexing

Indexing & Docs

Indexing & Docs

Codebase

Codebase Indexing [?](#)

Embed codebase for improved contextual understanding and knowledge. Embeddings and metadata are stored in the [cloud](#), but all code is stored locally.

100%

381 files



Sync



Delete Index

Index New Folders

Automatically index any new folders with fewer than 50,000 files



Ignore Files in .cursorignore

Files to exclude from indexing in addition to .gitignore. [View included files](#).



Index Repositories for Instant Grep BETA

Automatically index repositories to speed up Grep searches. All data is stored locally.



Docs

Crawl and index custom resources and developer docs

+ Add Doc

- AG-UI Docs

Indexed 12/5/25, 2:17 PM



Tools & MCP Servers

The screenshot shows a software interface with a sidebar and a main panel.

Top Bar: Shows a profile icon (M), the email address matt.eland@gmail.com, and a "Free Plan".

Search Bar: "Search settings Ctrl+F"

Left Sidebar: A navigation menu with the following items:

- General
- Agents
- Tab
- Models
- Cloud Agents
- Tools & MCP** (highlighted)
- Rules, Memories, ...
- Indexing & Docs
- Network
- Beta

Right Panel:

Tools

Browser:

- Browser Automation**
- Ready (Chrome detected)

Installed MCP Servers:

No MCP Tools

Add a custom MCP tool here or configure project-specific tools in `<project-root>/.cursor/mcp.json`

Add Custom MCP

Tips for Improving Context

Focus	New	Filter	Improve
Keep things focused on the task you want the agent to accomplish	Start new conversations when ready to move on to new focuses	Ignore irrelevant file types and directories so they're not indexed	If you find certain areas are tricky, you may need more rules or tweaks to agent files



Demo

Exploring Context in a Sample Project

Feedback & Contact Form



+

•

○

<https://forms.gle/SrTzVzFnsDz4kiA18>

Break

Time	Content
8 – 9:15 AM	Introductions, Terminology, and Setup Verification
9:30 – 10:30 AM	Tooling Discussion and Demonstration
10:45 – 11:30 AM	Cursor for Greenfield Development
11:30 AM – 12:30 PM	Catered Lunch
12:30 – 1:15 PM	Context Discussion and Sample App Exploration
1:15 – 2 PM	Cursor for Brownfield Development
2:15 – 3:45 PM	Advanced AI Integration
4 – 5 PM	Final discussions and topics of interest

•

Brownfield Development

Let's see what happens if we just work on production code without any added context



This afternoon's codebase

What do we get out of the box?

Switch to the 'workspace/no-rules' branch and start a new Agent session with the 'step6-historical-feature.md' prompt.

This branch is from a previous point in time for our repo, before the feature described in the prompt was implemented.

You can either commit changes locally or git stash these results to compare them with an implementation with additional context we'll do later

Demo

Let's try AI assisted development with zero changes!

Discussion

What happened?

What was good?

What wasn't?

What ways of trying to fix this might work?

Which are the best options to try first?

Feedback & Contact Form



+

•

○

<https://forms.gle/SrTzVzFnsDz4kiA18>

Break

Time	Content
8 – 9:15 AM	Introductions, Terminology, and Setup Verification
9:30 – 10:30 AM	Tooling Discussion and Demonstration
10:45 – 11:30 AM	Cursor for Greenfield Development
11:30 AM – 12:30 PM	Catered Lunch
12:30 – 1:15 PM	Context Discussion and Sample App Exploration
1:15 – 2 PM	Cursor for Brownfield Development
2:15 – 3:45 PM	Advanced AI Integration
4 – 5 PM	Final discussions and topics of interest

•

Advanced AI Integration

Time to add in some contextual sprinkles to see how we can improve things



Project-specific Scaffolding and Prompts

Writing a good AGENTS/Rules.md

Every time you use an Agent, it is seeing your codebase for the first time. This means you should tell it the:

- **What** – is this codebase? What are the features? Where are they defined?
- **Why** – do we have this codebase? What is its purpose? Why are things organized the way they are?
- **How** – do we use/build it? How do you style code? How do you run tests? How do you validate changes?

These rule files are included in *every single chat*, so we want them to be universally applicable to the project. High level, curated rules beats targeted 'hotfix' instructions that might only be applicable in narrow cases.

Bad AGENTS/Rules examples

1. Database Schema Instructions

When creating new database tables:

- Always include `created_at` and `updated_at` timestamps
- Use UUID primary keys, not auto-increment integers
- Add indexes on foreign keys
- Follow naming pattern: `user_profiles`, *not userProfiles*

Why bad: Only relevant when doing database work, wastes context otherwise.

3. One-Time Setup Instructions

Setting up the development environment:

1. Install Node.js 18+
2. Run ``npm install``
3. Copy `.env.example` to `.env`
4. Set up local PostgreSQL
5. Run migrations with ``npm run migrate``

Why bad: Setup happens once, not in every session. Bloats every conversation.

2. Detailed Code Style Guidelines

Code Style:

- Use 2 spaces for indentation
- Max line length 80 characters
- Prefer `const` over `let`
- [50 more formatting rules...]

Why bad: Linters do this better/faster, bloats context.

Teach the LLM to learn from examples.

Better AGENTS/Rules examples

1. Project Structure Overview

Project Structure:

- /apps/web – Next.js customer dashboard
- /apps/api – Go API server
- /apps/admin – Internal admin panel
- /packages/ui – Shared React components
- /packages/db – Database schemas and migrations

This is a pnpm monorepo. Always work from the root directory.

Why good: Helps Claude navigate and understand scope, relevant to all tasks.

3. Technology Stack

Tech Stack:

- Frontend: Next.js 14, TypeScript, TailwindCSS
- Backend: Go 1.21, PostgreSQL 15
- Infrastructure: Docker, deployed on AWS ECS
- Package manager: pnpm (not npm/yarn)

Why good: Informs all technical decisions, always relevant.

2. Progressive Disclosure Pointers

Documentation:

- Database conventions: see [agent_docs/database_design.md](#)
- API patterns: see [agent_docs/api_guidelines.md](#)
- Testing approach: see [agent_docs/testing_strategy.md](#)
- Deployment: see [agent_docs/deployment.md](#)

Read relevant docs before starting work on those areas.

Why good: Agent only reads what it needs, when it needs it.

Building Rules for your codebase

Create a high-level overview

Break down individual components

Consolidate into single artifact

Generate rules with supplemented best practices document + artifact

Evaluate rules by re-creating existing features. Refine as necessary.

Building Rules for your codebase

Start by building a comprehensive architecture document of your codebase. This can be done manually, or by utilizing Agents to break down your codebase piece by piece.

✉ product-mfe Give me a high-level overview of this repository. I want you to identify the primary components of the system. Determine what it is meant to do, how it does it. Identify common patterns used in the repo. Identify the main services and describe what they do. Create an architecture diagram of the repo.



Great, now I want you to dig into each one of the components that you identified, and break them out in more granular analysis. Collate this more granular investigation back into the initial high level review you made. Basically dig into the smaller parts and add them back in. Also, be sure to 'escape' mermain components with parenthesis like this: "Fragments (styles/content/scripts)"|

Ok, I now want you to convert this to a markdown file in the root of the repo, named 'ARCHITECTURE.md'. This should include the diagrams and findings you've made. This should have a consistent structure for all services. Include the mermaid diagrams, but also include a simple ascii tree diagram of the different components and their purpose.|



Step 0: Make a branch

There are 2 prepared branches for this exercise:

- workshop/no-rules
- workshop/with-rules

We will switch to those branches for a prepared demo, but you are encouraged to create a new branch and run these commands yourself as we build our Cursor Rules.

Step 1: High-Level Overview

This is an example prompt for starting your ARCHITECTURE.md document. This example is very generic and provides no repo-specific details, but the more direction you can give here, the better.

Start a new Agent Session and paste the 'step1-high-level-overview.md' prompt from the 'prompts' directory.

```
@product-mfe Give me a high-level overview of this repository. I want you to identify the primary components of the system. Determine what it is meant to do, how it does it. Identify common patterns used in the repo. Identify the main services and describe what they do. Create an architecture diagram of the repo.
```

Step 2: Granular Component Analysis

In the same Agent Chat session, continue to breakdown the analysis into lower level, component specific investigations.

Paste 'step2-granular-component-analysis.md' into the same Session.

Again, this example is generic, but more detailed instructions here will yield more accurate results.

Great, now I want you to dig into each one of the components that you identified for this codebase, and perform a detailed, granular analysis for each one. Collate this more granular investigation back into the initial high level review you made. Do this for every component/subsystem/service you can identify.

Step 3: Refine and save artifact

Once the Agent has finished reviewing the codebase, use this as an opportunity to review its findings. LLMs **can** and **will** make mistakes!

If you find incorrect information in the final analysis, you should correct the Agent in the same session. Once satisfied with its review, ask it to save the review in an ARCHITECTURE.md file for future reference for rule building.

Paste 'step3-save-architecture.md' into the same Session.

Step 4: Learning Rule-Building Principles

LLMs are stateless, they see your codebase for the first time every chat.

We want to give them enough context to start building the way we expect them to, but we don't want them to read the entire codebase every single chat.

- Teach the LLM *how to learn things* about the repo on its own, instead of giving the answer
- Keep Rules concise, reference existing code instead of hard-coding examples.
- Build progressive rules, link to other rules instead of fitting it all in one place.
- [This blog post](#) is a great resource for how to build effective rules.

Step 5: Create Initial Rules

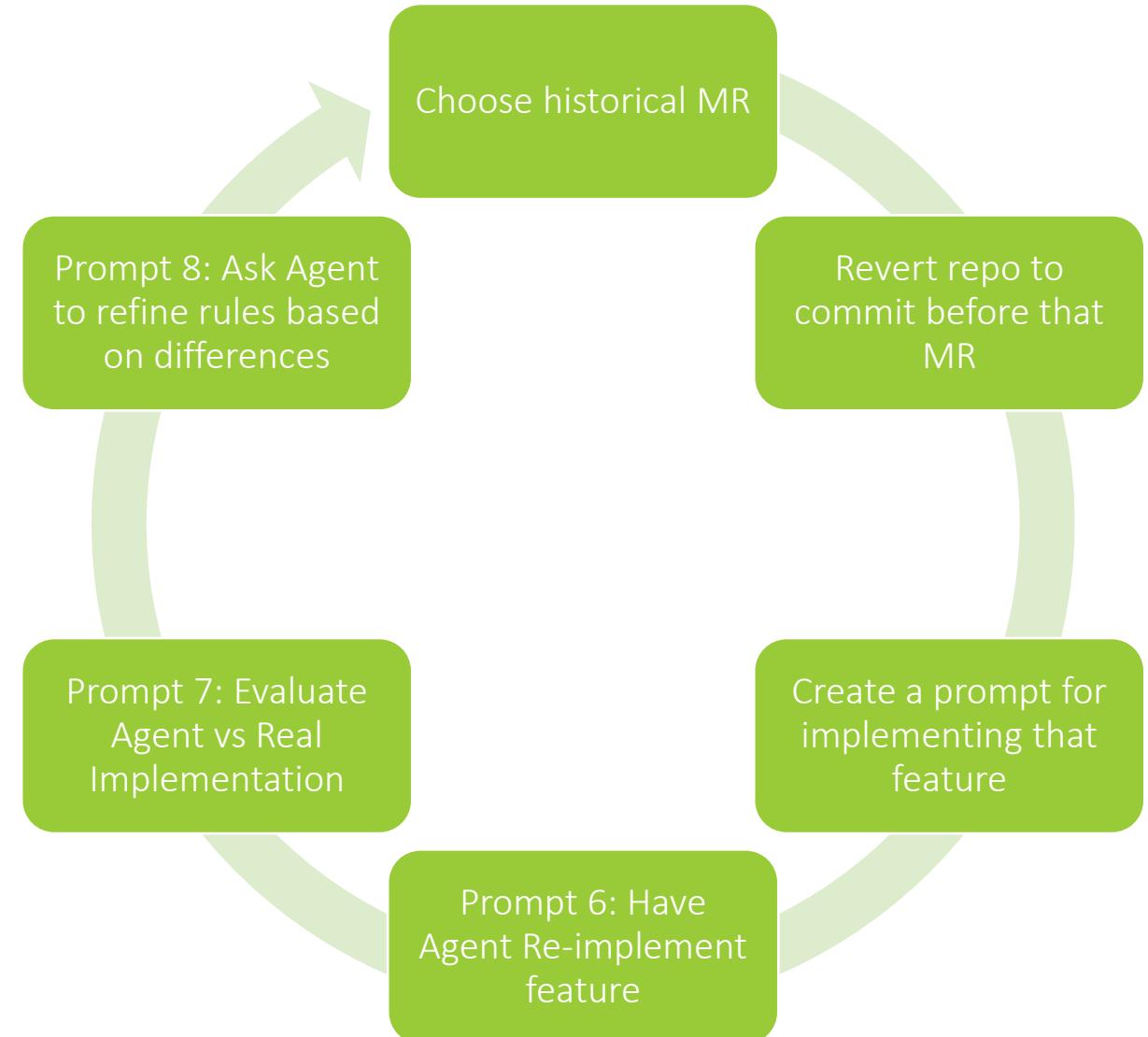
We've got 2 great pieces of context for building Rules, the `ARCHITECTURE.md` for our repo and best practices defined in the `writing-good-rules.md` document.

- Create a new Agent Session with prompt '`step5-initial-rules-creation.md`'
- This will give us the initial set of rules for our codebase, using the initial analysis as a starting point and the best practices document as a guide.
- This is a good starting point, but we will need to review and curate these rules for best results.
- Once you have finished, make a backup of the `.cursor/rules` outside of your repo (easier than dealing with git commands with a whole room of people)

Step 6: Validate Rules with Known Implementation

How do we know if these rules are good?

Test them out on something we know the ‘right’ answer to!



Step 6: Let's try this with Rules

Now that we know what results look like with no guidance, let's see how our Cursor Rules change the implementation.

Switch to the 'workshop/with-rules' branch. If you want to use the Rules you just created, overwrite the .cursor/rules folder in this branch with your backup copy. Otherwise this branch will use the ones previously created for this workshop.

Repeat the same 'step6-historical-feature.md' prompt as before in a new session.

This version will automatically follow the generated rules built from the previous Step 1 – 5.

Step 7: Comparing Human vs Agent MRs

Cursor Agents can use git to compare their implementation with the real merge commit.

This gives us a workflow for improving our rules to enforce code that is more similar to historical MRs.

Run prompt 'step7-compare-with-real-feature.md' in the same session as step 6-b

Comparison: My Implementation vs. Commit b497ebd

Files Comparison

File	My Implementation	Real Commit
ANFStoreAttrDeleteService.java	<input checked="" type="checkbox"/> Created	<input checked="" type="checkbox"/> Created
ANFStoreAttrValDeleteService.java	<input type="checkbox"/> Not created	<input checked="" type="checkbox"/> Created
ANFStoreAttrValRepository.java	<input checked="" type="checkbox"/> Added findByStoreAttr_StoreAttrId()	<input type="checkbox"/> No changes
Controller	Added 1 endpoint	Added 2 endpoints
Tests	Service + Controller tests	Service + Controller tests

Step 8: Rule Refinement

Now that we have identified the differences between the Agent and Human MRs, we can analyze our rule set to identify what areas we can improve.

We want to maintain the same best practices we used before and avoid 'hot-fixing' our rules.

Run the prompt in 'step8-analyze-rules-from-implementation.md' in the same session as Step 7. This will let the Agent compare its own analysis against the real feature and where the rules came up short for its version of the feature.

Context

Thought for 3s

Checking the SemanticSlicer package namespace and fixing the import:

Searched web drittich.SemanticSlicer NuGet package namespace C#

Auto-Ran command: dotnet

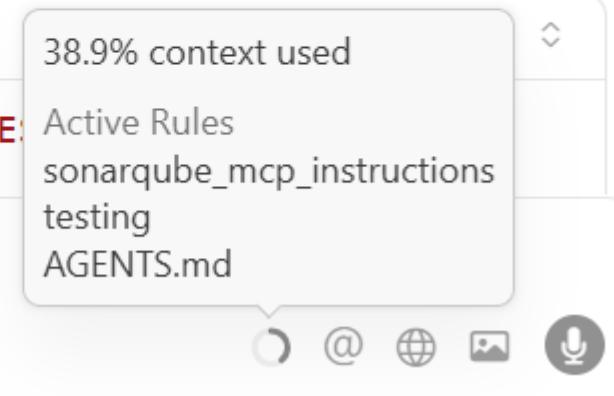
```
$ dotnet list "JAIMES_AF.Workers.DocumentChunking/JAIME:DocumentChunking.csproj" package
```

Plan, @ for context, / for commands

∞ Agent ▾ Auto ▾ 1×

38.9% context used

Active Rules
sonarqube_mcp_instructions
testing
AGENTS.md



Tools

Browser

Browser Automation
Ready (Chrome detected)

Show Localhost Links in Browser
Automatically open localhost links in the Browser Tab

Google Chrome

Installed MCP Servers

G GitKraken

```
c:/Users/matt.eland/AppData/Roaming/Cursor/User/globalStorage/eamodio.git-lens/gk.exe mcp --host=cursor --source=gitlens --scheme=cursor
git_add_or_commit git_branch git_checkout
git_log_or_diff git_push git_stash git_status git_worktree
gitkraken_workspace_list issues_add_comment issues_assigned_to_me
issues_get_detail pull_request_assigned_to_me pull_request_create
pull_request_create_review pull_request_get_comments
pull_request_get_detail repository_get_file_content
code_review_branch
```

L linear

```
npx -y mcp-remote https://mcp.linear.app/sse
list_comments create_comment list_cycles get_document
list_documents get_issue list_issues create_issue update_issue
list_issue_statuses get_issue_status list_issue_labels
create_issue_label list_projects get_project create_project
update_project list_project_labels list_teams get_team list_users
get_user search_documentation
```

+ New MCP Server
Add a Custom MCP Server

Thought for 2s

Checking if IMessagePublisher is registered and updating Program.cs:
Grepped IMessagePublisher in Program.cs
Read Program.cs L180-219
Thought for 4s

Updating Program.cs to register IMessagePublisher and both strategies conditionally:

```
C# Program.cs +51-29
// Register IMessagePublisher for queuing chunks without embeddings
builder.Services.AddSingleton<IMessagePublisher, MessagePublisher>();

// Register chunking strategy based on configuration
string chunkingStrategy = options.ChunkingStrategy ?? "SemanticChunker";
bool useSemanticChunker = string.Equals(chunkingStrategy, "SemanticChunker", StringComparison.OrdinalIgnoreCase);

C# Program.cs +6-2
logger.LogInformation("chunking strategy: {Strategy}", chunkingStrategy);
if (useSemanticChunker)
{
    logger.LogInformation("Ollama Endpoint: {Endpoint}", ollamaEndpoint);
    logger.LogInformation("Ollama Model: {Model}", ollamaModel);
}
```

Read Program.cs L1-25
Thought for 4s

Adding the using statement and checking for compilation errors:
No linter errors found in JAIMES_AF.Workers.DocumentChunking
Read SemanticSliceStrategy.cs
Thought for 3s

Building the solution to check for compilation errors:

Ran command: dotnet

```
$ dotnet build "JAIMES_AF.Workers.DocumentChunking/JAIMES_AF.Workers.DocumentChunking.csproj"
1 Error(s)
```

Time Elapsed 00:00:08.24

Tools and Tool Calling

MCP Servers

Installed MCP Servers

G

GitKraken

c:\Users\matt.eland\AppData\Roaming\Cursor\User\globalstorage\eamodio.gitlens\gk.exe mcp --host=cursor --source=gitlens --scheme=cursor

git_add_or_commit git blame git_branch git_checkout

git_log_or_diff git_push git_stash git_status git_worktree

gitkraken_workspace_list issues_add_comment issues_assigned_to_me

issues_get_detail pull_request_assigned_to_me pull_request_create

pull_request_create_review pull_request_get_comments

pull_request_get_detail repository_get_file_content

code_review_branch

Show less ▾

L

linear

npx -y mcp-remote https://mcp.linear.app/sse

Show less

+

New MCP Server

Add a Custom MCP Server

Browser MCP Support

Tools

Browser

Browser Automation

Google Chrome ▾

Ready (Chrome detected)

Show Localhost Links in Browser

Automatically open localhost links in the Browser Tab



Featured MCPs

Browse MCPs or post a MCP to reach 250,000+ monthly active developers.

Add new + ⏪ ⏴ ⏵



Atono

Atono's MCP server connects your AI coding assistant directly to your product backlog, pulling your...



Endgame

Deployment superpowers for Cursor. Fast + Self-healing + Free - endgame.dev...



Mailtrap Email Sending

Integrates with Mailtrap Email Platform to enable sending transactional emails via API.



MailerSend MCP

Access the transactional messaging MailerSend API and services directly from your AI coding environment or...

Search 1800+ MCPs



Ashra MCP Server

Extract structured data from any website with a simple SDK call. No scraping code, no...



Compresto MCP

A Model Context Protocol (MCP) server for Compresto, providing AI assistants with...



Caiyun Weather

Caiyun Weather is a weather data service that provides real-time weather...



Encore

Encore turns Cursor into more than a coding tool. The open source framework lets you...



urlDNA

Scan and analyze potentially malicious URLs using the urlDNA



Pearl

The Pearl MCP Server expands the ecosystem of MCP-compatible tools by...



AWS Knowledge MCP

A fully managed remote MCP server that provides up-to-date documentation, code...



AWS API MCP Server

The AWS API MCP Server enables AI assistants to interact with AWS services...

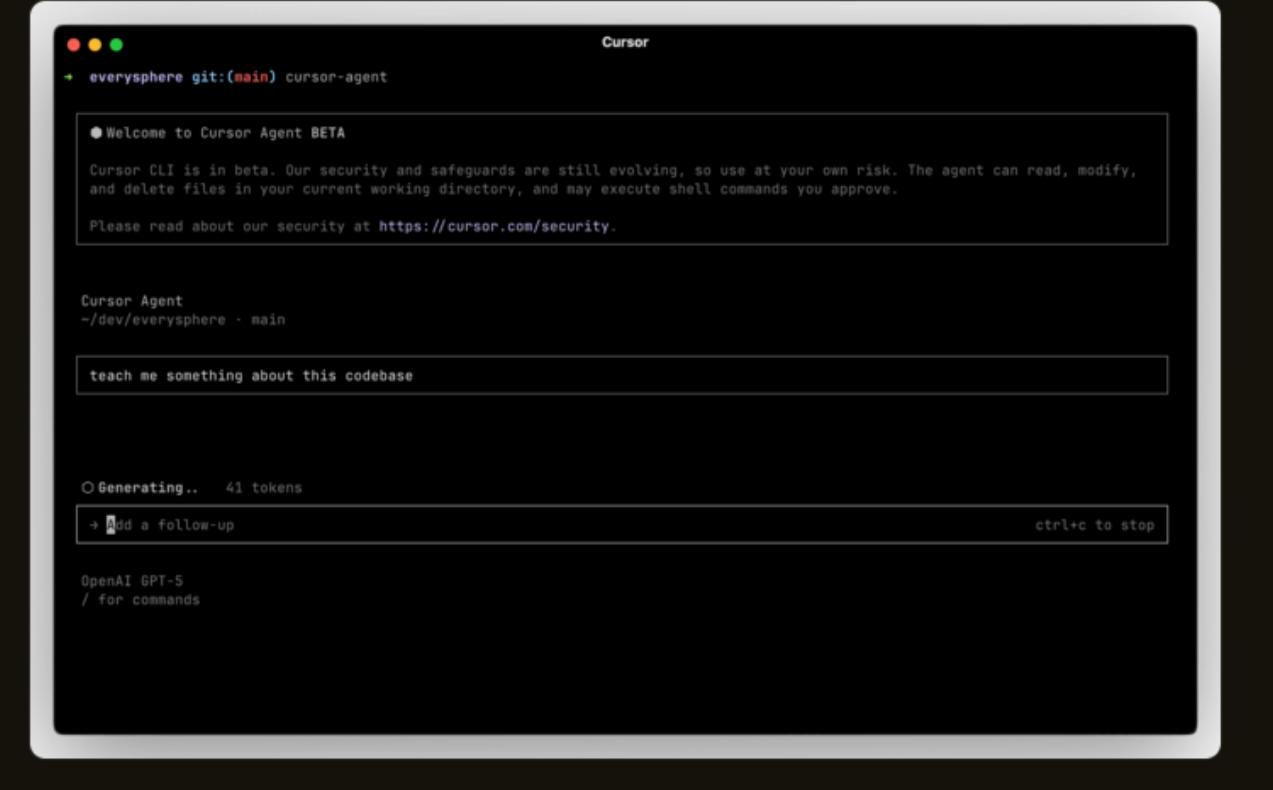
External MCP Servers

<https://cursor.directory/mcp>

Cursor CLI

Getting started

```
1 # Install
2 curl https://cursor.com/install -fsS | bash
3
4 # Run interactive session
5 cursor-agent
```



The screenshot shows the Cursor Agent BETA interface. At the top, it displays the command: `everyisphere git:(main) cursor-agent`. Below this, a welcome message reads: "Welcome to Cursor Agent BETA. Cursor CLI is in beta. Our security and safeguards are still evolving, so use at your own risk. The agent can read, modify, and delete files in your current working directory, and may execute shell commands you approve. Please read about our security at <https://cursor.com/security>". The main area shows the output of the command: "teach me something about this codebase". The response is: "Generating.. 41 tokens". A button labeled "Add a follow-up" is visible. At the bottom, it says "ctrl+c to stop". The interface has a dark theme with light-colored text and a header bar.

The screenshot shows the Cursor documentation website. The top navigation bar includes links for 'CURSOR', 'Docs' (which is highlighted in orange), 'API', and 'Learn'. A search bar and an 'Ask AI' button are also present. The left sidebar contains a hierarchical menu with sections like 'Get Started', 'Welcome', 'Quickstart', 'Concepts', 'Models', 'Pricing', 'Core', 'Tab', 'Agent', 'Cloud' (which is expanded to show 'Overview', 'Web & Mobile', and 'API'), 'CLI', 'Inline Edit', 'Rules', 'Bugbot', 'Context', 'Codebase Indexing', 'Ignore files', 'Model Context Protocol (MCP)' (which is expanded to show '@ Symbols'), 'Integrations', 'Slack', 'Linear', 'GitHub', 'Git', and 'Deeplinks'. The main content area is titled 'Cloud Agents'. It starts with a brief introduction: 'With cloud agents, spawn asynchronous agents that edit and run code in a remote environment. View their status, send follow-ups, or take over anytime.' It then explains that 'Cloud agents leverage the same agent fundamentals but run autonomously in isolated environments.' Below this, there are two sections: 'How to Use' and 'Setup'. The 'How to Use' section describes how to access cloud agents via the sidebar or mode keybinding, and how to select an agent after submission. It also notes that 'Cloud agents require data retention on the order of a few days.' The 'Setup' section details the default setup where agents run in an isolated Ubuntu-based machine with internet access and package installation capabilities. It also mentions GitHub connection and base environment setup.

Get Started

Welcome

Quickstart

Concepts

Models

Pricing

Core

Tab

Agent

Cloud

- Overview
- Web & Mobile
- API

CLI

Inline Edit

Rules

Bugbot

Context

Codebase Indexing

Ignore files

Model Context Protocol (MCP)

- @ Symbols

Integrations

Slack

Linear

GitHub

Git

Deeplinks

Cloud Agents

With cloud agents, spawn asynchronous agents that edit and run code in a remote environment. View their status, send follow-ups, or take over anytime.

Cloud agents leverage the same agent fundamentals but run autonomously in isolated environments.

How to Use

You can access cloud agents in two ways:

1. **Cloud Agent Sidebar:** Use the cloud agent tab in the native Cursor sidebar to view all cloud agents associated with your account, search existing agents, and start new ones.
2. **Cloud Agent Mode:** Hit `Ctrl E` to trigger cloud agent mode in the UI.

After submitting a prompt, select your agent from the list to view status and enter the machine.

ⓘ Cloud agents require data retention on the order of a few days.

Setup

Cloud agents run in an isolated Ubuntu-based machine by default. Agents have internet access and can install packages.

GitHub connection

Cloud agents clone your repo from GitHub and work on a separate branch, pushing to your repo for easy handoff.

Grant read-write privileges to your repo (and any dependent repos or submodules). We'll support other providers (GitLab, BitBucket, etc) in the future.

Base Environment Setup

How to Use

Setup

- GitHub connection
- Base Environment Setup
- Maintenance Commands
- Startup Commands
- The 'environment.json' Spec
- Using AWS IAM Roles

Models

Pricing

Security

Dashboard settings

Defaults Settings

Security Settings

Naming History

Copy page

Share feedback

Explain more

Cloud Agents

Bugbot

Cursor bot reviewed 2 days ago

```
backend/server/src/bugbotGithub/applyBugbotPRReview.ts
```

228 - const reviewCommentsResponse = await octokit.request('GET /repos/{owner}',
228 + const reviewCommentsResponse = await octokit.request('GET /repos/{owner}',
229 owner: repository.owner,
230 repo: repository.repo,
231 pull_number: pullRequest.issueNumber,
232 - review_id: commentResponse.data.id,
232 + since: new Date(Date.now() - 10000).toISOString(), // Only get recent comments
233);

Cursor bot 2 days ago

Bug: PR Comments Fetching Misses New Inline Comments (Potential edge case)
By switching to all PR comments and relying on a fixed 10s since filter without pagination, Bugbot can miss new inline comments — either due to timing delays or because they aren't included in the first batch of results.

 [Fix in Cursor](#) [Fix in Web](#)

Bugbot Add-on

Free

\$0

Includes:

- ✓ Limited code reviews each month
- ✓ Unlimited access to Cursor Ask
- ✓ Cursor connection to auto-fix bugs
- ✓ GitHub integration

[Try Bugbot](#)

Pro

\$32/user/mo.

Everything in Free, plus:

- ✓ 14 day individual trial
- ✓ Unlimited reviews on up to 200 PRs/month
- ✓ Access to Bugbot Rules

[Get Bugbot Pro](#)

Teams

\$32/user/mo.

Everything in Pro, plus:

- ✓ 14-day team trial
- ✓ Unlimited code reviews on all PRs
- ✓ Analytics and reporting dashboard
- ✓ Advanced rules and settings

[Get Bugbot Teams](#)

Enterprise

Custom

Everything in Teams, plus:

- ✓ 30-day org-wide trial
- ✓ Advanced analytics and reporting
- ✓ Priority support and account management

[Contact Sales](#)

Bugbot Editions

Hands-on Exercise

Add targeted context, then repeat some of our steps from earlier

Discussion

How did this go?

What was most effective?

Did anything not help as much as you thought it might?

Feedback & Contact Form



+

•

○

<https://forms.gle/SrTzVzFnsDz4kiA18>

Break

Time	Content
8 – 9:15 AM	Introductions, Terminology, and Setup Verification
9:30 – 10:30 AM	Tooling Discussion and Demonstration
10:45 – 11:30 AM	Cursor for Greenfield Development
11:30 AM – 12:30 PM	Catered Lunch
12:30 – 1:15 PM	Context Discussion and Sample App Exploration
1:15 – 2 PM	Cursor for Brownfield Development
2:15 – 3:45 PM	Advanced AI Integration
4 – 5 PM	Final discussions and topics of interest

•

Demo

Generating and Improving Test Coverage

Bonus Content?

As time allows

+
•
○

Conclusion

Final thoughts, recommendations, and next steps / resources

Recommendations for Long-Term Success

1

Store context in project-specific rules

2

Bootstrap your initial context, then review it

3

Start in a narrowly scoped manner and expand

4

Update documentation and context as things change

5

Get Cursor's help flagging weak areas

Injecting Context in Cursor

Conversational
prompt

Code and
Comments

Agents.md

Rules

Memories

Commands

Team-specific
Context

User-specific
Context

Indexed
Documents

External tools /
MCP Servers

Adopting AI Tooling

01

Start with a specific scope in mind

- Team / Project
- Codebase
- Tooling

02

Pick work items that are well-defined, speced out, and deal with common technologies

03

Plan in advance for testing and code review

04

Provide relevant context and instructions for how to operate (tests, docs, building code, etc.)

05

Consider pre-built Agent files as a starting point

Onboarding new developers with AI in mind



AI USE IS EXPECTED,
BUT NOT A SUBSTITUTE
FOR COMMON SENSE



TESTING IS CRITICAL



CODE REVIEWS BECOME
EVEN MORE IMPORTANT



THE WHOLE TEAM
ALWAYS OWNS
SOFTWARE QUALITY



DON'T TRUST THE AI

Additional Resources

<https://blog.LeadingEDJE.com>

<https://cursor.com/docs>

<https://cursor.directory/learn>

<https://Agents.md>

Get In Touch



Bryce Amacker



Matt Eland



Dave Michels



Open Q&A



Feedback & Contact Form



<https://forms.gle/SrTzVzFnsDz4kiA18>