

CodeArts Agent

User Guide (Space)

Issue	01
Date	2026-07-30



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2026. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

Contents

1 What Is CodeArts Agent Space?	1
2 Sandbox	6
3 R&D Office	16
4 Multi-chat Parallelism	25
5 Multi-task Parallelism	29
6 Scheduled Task	38
7 Code Review	42
8 Tool Panel	44
9 Settings	50
10 IM Integration	58
10.1 Lark	58
10.2 Slack	62

1 What Is CodeArts Agent Space?

CodeArts Agent Space is a next-generation, agent-centric interactive experience mode launched by CodeArts Agent, specifically designed for multi-agent collaboration and client-cloud integrated scenarios.

Supported OSs

- Windows OS: **Windows 11 (x64) is recommended.** For Windows 10 (x64), the version must be 2019 or later, and upgrading to the latest stable version is advised.
- macOS: macOS 11 or later, compatible with ARM64 (Apple Silicon).
- Linux: Linux (ARM64) architecture, glibc version ≥ 2.31 , glibcxx version $\geq 3.4.26$.

Enabling CodeArts Agent Space

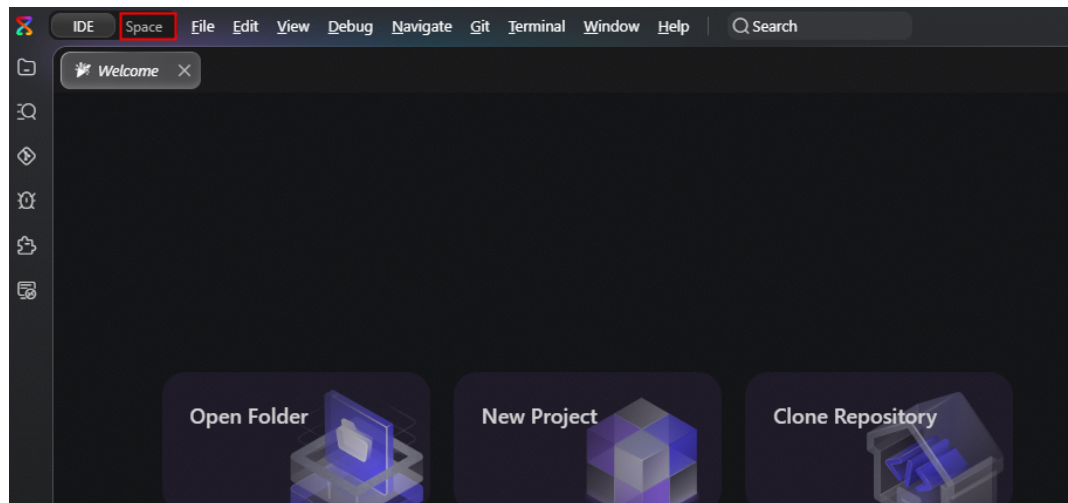
Step 1 Install CodeArts Agent IDE.

Download the CodeArts Agent IDE installation package from [the official website of CodeArts Agent](#) and install it.

Step 2 Switch to CodeArts Agent Space.

In the upper left corner of CodeArts Agent IDE, click **Space** to switch to CodeArts Agent Space.

Figure 1-1 Switching the mode



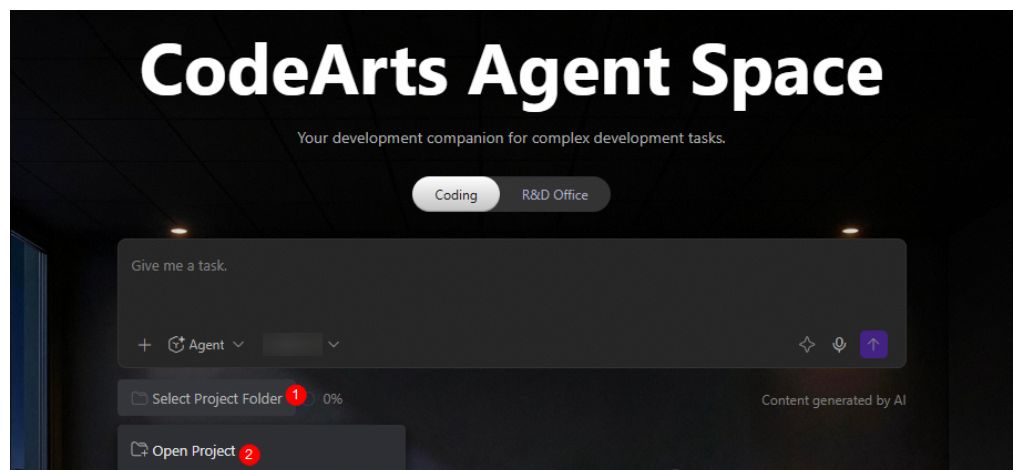
NOTE

- When you open CodeArts Agent IDE for the first time, it enters the IDE mode by default.
- When multiple IDE windows are opened simultaneously and then all closed, the mode of **the last closed window** is automatically recorded. Upon restarting the IDE, it will automatically restore to this mode (IDE mode/Space mode) without requiring manual switching.

Step 3 Open a folder to store files generated in the project.

- Method 1: In CodeArts Agent Space, click **Select Project Folder**, choose **Open Project**, and open a folder.

Figure 1-2 Opening a folder



- Method 2: In the dialog box at the bottom of CodeArts Agent Space, click **Open Folder**.

Figure 1-3 Clicking Open Folder in the dialog box



-----End

CodeArts Agent Space User Interface

CodeArts Agent Space features a three-column layout, consisting of the chat management panel, AI chat panel, and tool panel from left to right.

Figure 1-4 CodeArts Agent Space user interface

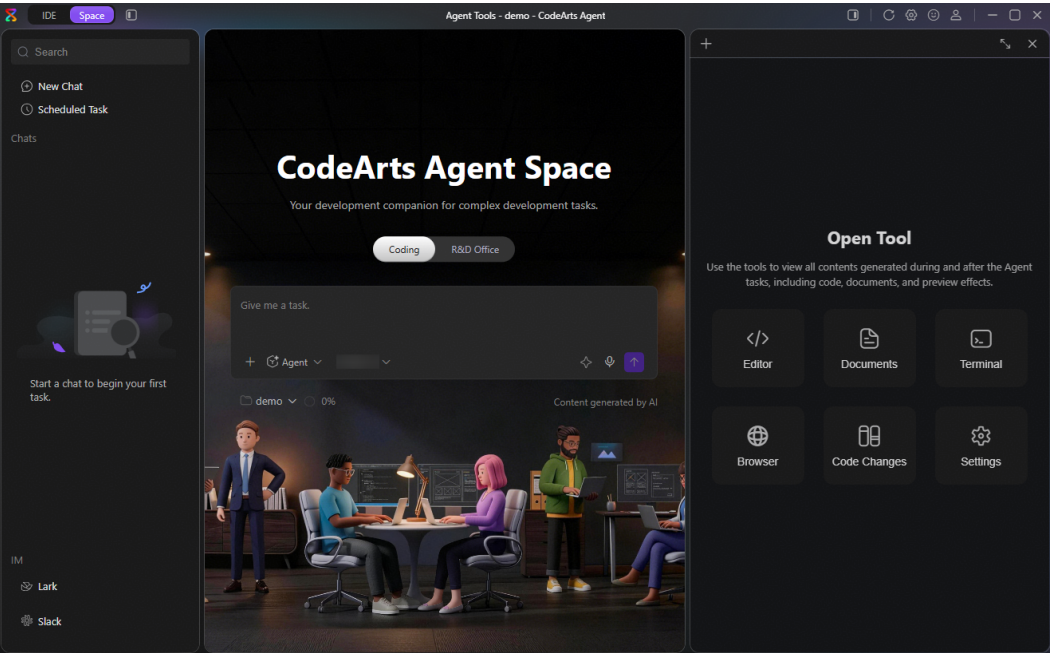


Table 1-1 CodeArts Agent Space user interface functions

Function	Description
Creating a chat	Click New Chat in the upper left corner to open a new chat window. To ensure smooth user experience, it is recommended that no more than five chat windows be opened at the same time. NOTE During the response generation, you can initiate a new chat in the input box of the current chat window, and the new chat will be automatically added to the queue. The queue can hold a maximum of 20 waiting messages; any additional messages will not be accepted.
Scheduled tasks	Create and manage scheduled tasks using natural language, including periodic tasks (such as daily code checks) and one-time tasks (such as tomorrow's submission reminders), achieving flexible task scheduling and control. For details, see Scheduled Task .
Chat history	View the stored memory content formed during the interaction between CodeArts Agent Space and users. It records the chat history between users and the AI, facilitating subsequent task continuity and contextual understanding. For details, see Managing Chats .
IM integration	Integrate Lark chatbot and Slackbot through the remote collaboration capabilities provided by CodeArts Agent Space. This achieves seamless connectivity between CodeArts Agent Space, mobile devices, and collaboration endpoints, effectively supporting the efficient flow of tasks across different platforms. For details, see IM Integration .

Function	Description
Chat interaction	• Text input: Enter your requirements directly in the input box, for example, "Help me generate a technical design document for the user login module." • Voice input: Enable voice input when typing is inconvenient, when you need quick verbal communication, or when structuring text instructions is difficult. This function supports up to 10 minutes of continuous recording each time, eliminating lengthy text descriptions and helping you achieve precise and efficient communication of your needs. For details, see Voice Input. • File upload: You can upload only spreadsheets (.xlsx, .xls, and .csv), documents (.ppt, .pdf, .doc, .docx, .md, and .txt), and images (.jpg, .png, .jpeg, and .webp) for data analysis, document generation, and more. – Images: The size of a single image cannot exceed 10 MB, and the total size of all images cannot exceed 20 MB. – Non-image attachments: A maximum of 20 attachments can be uploaded at a time, with no restrictions on the attachment size. • Mode selection: Currently, only the agent and agent team modes are supported, primarily used for executing tasks and generating content. • Large language model (LLM) support: The LLMs are consistent with those supported by CodeArts Agent IDE to ensure seamless integration with the development environment.
Tool panel	CodeArts Agent Space features a variety of built-in core tools, such as editors, documents, terminals, and browsers. You can flexibly call these tools based on your needs to expand further capabilities. For details, see Tool Panel .

2 Sandbox

A sandbox provides an **isolated and restricted execution space** for instructions generated by agents. Through strict permission control, it ensures that commands run within a securely isolated environment. This effectively blocks access to unauthorized resources (such as files and networks) or the execution of high-risk commands (such as permission modification commands). For intercepted commands, the agent will issue a risk prompt within a chat. They will only be executed outside the sandbox after a secondary confirmation by the user.

Using a sandbox offers the following advantages:

- **Preventing data leakage:** Based on sandbox technology, system-level and user-level sensitive directories are isolated, blocking secret leakage risks at the source and comprehensively protecting core assets.
- **Preventing direct execution of high-risk commands:** Commands are executed in a sandbox environment, where potentially dangerous operations (e.g., deleting all files on the D drive or modifying user permissions) can be identified and blocked in advance. A prompt is also generated in the chat flow for user decision-making.
- **Supporting enterprise compliance requirements:** This mechanism meets stringent regulatory requirements for development environment file isolation in industries such as finance, healthcare, and government, reducing security concerns and accelerating enterprise procurement decisions.

Constraints

Table 2-1 Constraints

Category	Description
OS	• Windows OS: Windows 11 (x64) is recommended. For Windows 10 (x64), the version must be 2019 or later, and upgrading to the latest stable version is advised. • macOS: macOS 11 or later, compatible with ARM64 (Apple Silicon). • Linux OS: Those on the x64 and ARM64 architectures are supported.

Category	Description
Controlled object	• Windows: Only Bash and PowerShell commands are controlled. • macOS: Only Shell commands are controlled. • Linux: Only Shell commands are controlled.

File Access Control

After sandbox is enabled, CodeArts Agent Space configures file directory access permissions as described below. You can also configure the **sandbox.json** file to customize the file and network access scope of processes in the sandbox environment of the current project. For details, see [Step 4](#).

Table 2-2 File access control

OS	Permission Type	Directory Type	Directory List
Windows	Read-only	-	All directories are readable except critical Windows system directories and sensitive user directories .
	Read/Write	Project directories and their subdirectories	-
	No read/No write	Critical Windows system directories	• C:\Windows\System32\config • C:\Windows\System32\drivers\etc • C:\Windows\SysWOW64\config • C:\ProgramData\Microsoft\Crypto • C:\Windows\System32\GroupPolicy • C:\Windows\System32\GroupPolicyUsers • C:\ProgramData\Microsoft\Windows\WER
		Sensitive user directories	• C:\Users*\AppData\Local\Microsoft\Credentials • C:\Users*\AppData\Roaming\Microsoft\Credentials • C:\Users*\AppData\Local\Microsoft\Protect • C:\Users*\NTUSER.DAT • C:\Users*\ntuser.dat.LOG

OS	Permission Type	Directory Type	Directory List
macOS	Read-only	-	All directories are readable except sensitive system directories .
	Read/Write	Workspace and additionally configured directories	• Temporary directories: /tmp, /var/ folders, and TMPDIR environment variable paths • Cache directories: ~/Library/Caches and ~/.cache • Dependency directories of common tools: ~/.local/lib, ~/.local/bin, and ~/.local/share • Toolchain and dependency directories of common development languages (Go, Java, Python, Node.js, Rust, and Ruby)
	No read/No write	Documents/ Desktop/ Downloads (privacy-related)	• ~/Desktop • ~/Documents • ~/Downloads • ~/Pictures • ~/Movies • ~/.ssh • ~/.zsh_history
		Password/ Wallet/ Keychain-related	~/Library/Keychains
		System-level sensitive configuration	• /etc/passwd • /private/etc/passwd • /etc/group • /private/etc/group • /etc/hosts • /private/etc/hosts • /etc/resolv.conf • /private/etc/resolv.conf • /etc/pam.d • /private/etc/pam.d
Linux	Read-only	-	All directories are readable except sensitive system directories .

OS	Permission Type	Directory Type	Directory List
	Read/Write	Workspace and additionally configured directories	• Temporary directories: /tmp and TMPDIR environment variable paths • Cache directories: ~/.cache and XDG_CACHE_HOME environment variable paths • Dependency directories of common tools: ~/.local/lib, ~/.local/bin, and ~/.local/share • Toolchain and dependency directories of common development languages (Go, Java, Python, Node.js, Rust, and Ruby)
	No read/No write	Critical Linux system directories	• /etc/shadow • /etc/passwd • /etc/group • /etc/gshadow • /etc/resolv.conf • /etc/sudoers

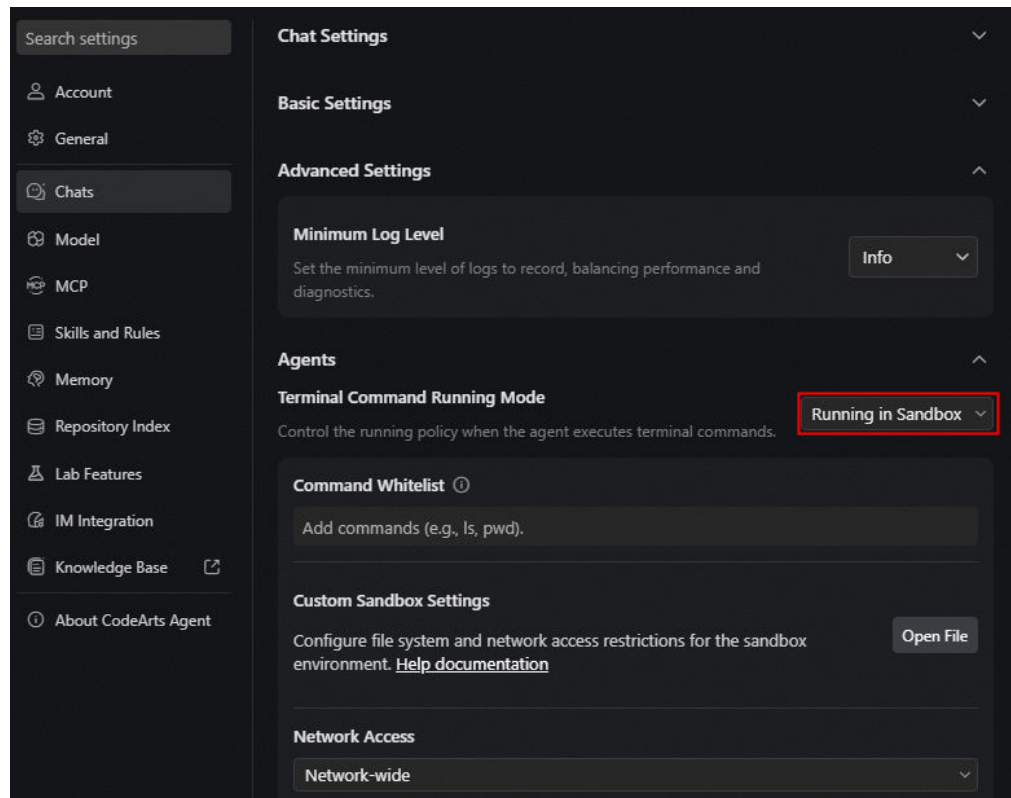
Enabling Sandbox

Step 1 Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).

Step 2 Click  in the upper right corner of CodeArts Agent Space to open the settings page.

Step 3 Enable the sandbox mode.

1. In the navigation pane on the left, choose **Chats**.
2. Under **Agents**, select **Running in Sandbox** from the **Terminal Command Running Mode** drop-down list.

Figure 2-1 Selecting Running in Sandbox**Step 4** Customize sandbox configuration.

By configuring the **sandbox.json** file, you can customize files and the network access scopes for processes within the current project's sandbox environment.

1. Click **Open Configuration** next to **Sandbox Custom Configuration**. The IDE will generate a **sandbox.json** file in your local user directory (**%USERPROFILE%\codeartsdoer**) and automatically open it in the editor.

The initial structure of the file is as follows:

```
{
  "filesystem": {
    "readWrite": [],
    "readOnly": []
  },
  "network": {
    "default": "Allow",
    "allow": [],
    "deny": []
  },
  "resources": {
    "cpu": 50,
    "memory": 8
  }
}
```

Table 2-3 Parameters in the sandbox.json file

Parameter	Mandatory (Yes/No)	Parameter Type	Description
filesystem	No	filesystem Object	Used to precisely control the sandbox's access permissions to the local file system. If it is not set (the filesystem field is empty or does not exist), the sandbox's built-in file system security policy will be applied.
network	No	network Object	Used to control the network access policies for processes within the sandbox, supporting configurations to allow or block access to specific network resources. If it is not set, network access is allowed by default.
resources	No	resources Object	Used to define the maximum limit of computing resources during sandbox runtime, ensuring service stability and preventing resource abuse. If it is not set, the system will share the host machine's resources.

Table 2-4 Parameters in filesystem

Parameter	Parameter Type	Default Value	Format of Supported Paths	Priority Rule	Description
readWrite	Array	[]	- Absolute path: for example, /home/user/project or C:\Projects - Relative path: for example, ./src or ./config - Environment variable: \$HOME (Linux/Mac) or %USERPROFILE% (Windows) - Abbreviation of the home directory: ~	readOnly > readWrite > Default system policy If a specific path matches both readOnly and readWrite, readOnly shall prevail.	List of read/write paths.
readOnly	Array	[]			List of read-only paths.

Table 2-5 Parameters in network

Parameter	Parameter Type	Default Value	Priority Rule	Description
default	String	Allow	deny > allow > default If both allow and deny are configured, deny shall prevail.	Default network access policy. - Allow: access allowed. - Deny: access denied. NOTE This field supports two configuration formats: <i>Domain:Port</i> and <i>IP address:Port</i> . Wildcards are supported for the domain part, and CIDR notation is supported for IP addresses. Multiple ports can be configured by separating them with commas. If no port is specified, the rule applies to all ports by default.
allow	Array	[]		Network rules that allow access to processes in the sandbox.
deny	Array	[]		Network rules that reject access to processes in the sandbox.

Table 2-6 Parameters in resources

Parameter	Parameter Type	Default Value	Min. Value	Description
cpu	Integer	50	20	CPU usage, in percentage.
memory	Integer	8	1	Memory size, in GB.

2. In the opened **sandbox.json** file, enter your custom sandbox configuration.

Example configuration of **sandbox.json**:

```
{
  "filesystem": {
    "readWrite": [
      "/home/user/project/output",
      "~/workspace/temp"
    ],
    "readOnly": [
      "/etc/systemd",
      "%USERPROFILE%/.ssh"
    ]
  },
  "network": {
    "default": "Allow",
    "deny": [
      "10.0.0.0/8",
      "192.168.0.0/16"
    ]
  },
  "resources": {
    "cpu": 50,
    "memory": 4
  }
}
```

3. Save the file for the configuration to take effect.

Step 5 Configure the command whitelist.

Add prefixes of specific commands to the whitelist as required. Commands added to the whitelist bypass the sandbox mechanism and are executed outside the sandbox.

Step 6 Configure the network access policy.

Table 2-7 Network access policy description

Parameter	Description
Network-wide	Allows access to all internal and external network resources.
Local	Only allows access to local networks (LAN/intranet); blocks all external network access.
Blocked	Blocks all network connections and prohibits access to any internal or external resources.
Custom	Allows you to modify the JSON policy configuration file to customize the file and network access scope for processes within the project sandbox environment. For details about how to configure the JSON file, see Step 4 .

Step 7 Exit the current setting page. The sandbox mode is enabled.

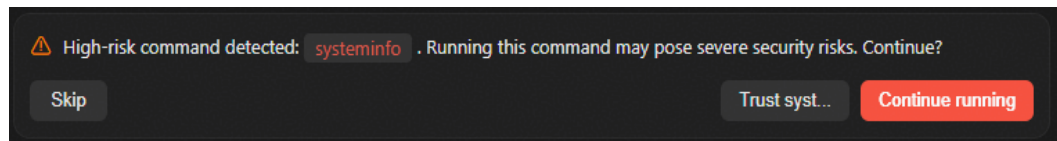
----End

Execution Policy for High-Risk Commands

When the agent detects a high-risk command, the AI will display a prompt in the chat flow. You need to evaluate risks and select an execution mode as required.

- **Skip:** The target command is not executed.
- **Trust XXX:** The prefix of the target command is added to the whitelist. Then all commands starting with this prefix will be executed outside the sandbox.
- **Continue running:** The target command can be executed outside the sandbox for the current chat.

Figure 2-2 High-risk command prompt



3 R&D Office

The R&D Office function focuses on non-coding scenarios in the daily routine of R&D personnel. Leveraging the intelligent chat mode of CodeArts Agent Space, you can easily complete tasks such as document writing, requirement sorting, business analysis, work reporting, and team collaboration, significantly improving office efficiency.

Five Major Application Scenarios

- Intelligent technical document generation: Based on code context and prompts, the system automatically organizes logic and generates clearly structured technical documents, eliminating tedious manual formatting.
- Deep insight into R&D data: The system automatically collects and analyzes R&D process data to generate visual reports, providing a quantitative basis for team efficiency assessment and project progress monitoring.
- Rapid Product Requirement Document (PRD) construction: The system converts vague business ideas into rigorously structured PRDs through chat interaction, accelerating requirement implementation.
- One-click report generation: The system integrates project progress, key milestones, and data achievements to quickly generate structured reporting materials, making weekly and monthly reports effortless.
- Seamless mobile collaboration: The system deeply integrates with instant messaging (IM) to support initiating collaboration, approvals, and communication anytime and anywhere from mobile devices, ensuring real-time information synchronization and efficient decision-making in remote scenarios.

Intelligent Technical Document Generation

CodeArts Agent Space features built-in intelligent generation capabilities for design documents and technical proposals, allowing for the rapid output of well-structured and comprehensive documentation.

Step 1 Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).

Step 2 Enter a task prompt. Click **Doc Generation**. Type your task into the box, and click  or press **Enter** to submit your request.

If you need to generate technical documents based on existing materials, first click  and select **Upload Attachments** to upload files, and then enter a task prompt. Write a guide on office backend permission allocation for new employees. It must cover steps, precautions, and FAQs, ensuring that the language is easy to understand and the format is standard.

In addition to manual task input, CodeArts Agent Space also provides you with pre-configured templates for frequently used prompts. You can click to automatically fill them into the input box to help you get started quickly.

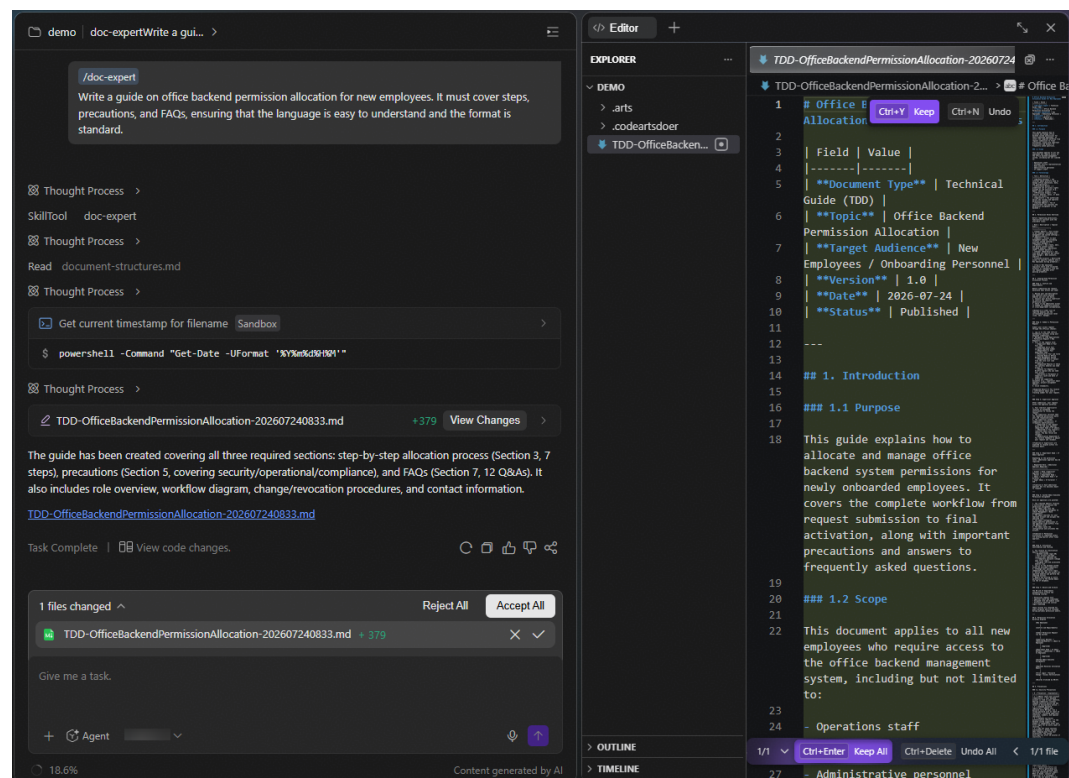
You are a rigorous senior backend engineer, expert in transforming technical solutions into actionable, detailed design documents.

You are a veteran technical architect, expert in drafting clear, well-structured technical proposals. Based on the requirement description, generate a comprehensive technical solution.

You are a backend engineer specialized in API design and fluent in RESTful specifications. Based on the described features, generate a RESTful-compliant API document.

- Step 3** View the generated results. After the task is complete, a document in Markdown format will be automatically generated under **Editor > EXPLORER**.

Figure 3-1 View technical documents



- Step 4** Confirm the results. View the information of the generated files and operate according to your actual needs, or modify them directly based on the generated content.

In CodeArts Agent Space, context changes are presented through a Diff view, making the modified content more intuitive and easy to understand. The chat interface will display the total number of files, lines of code, and the file list involved in this change. Clicking **View Changes** after any file allows you to view the specific modifications in the Diff view, helping you clearly grasp every detail of the changes.

Choose either of the following methods to confirm the files as required:

- Method 1: quick operations on the chat interface

Hover the mouse over the row where the file is located, and click  or  on the right to accept or reject the file changes, respectively. If there are multiple files, you can click **Accept All** to accept all files or **Reject All** to reject all files.

- Method 2: operations in the explorer

Under **Editor** > **EXPLORER**, view the content of the generated file. Click **Keep** (Ctrl+Y) on a segment to accept the generated content, or click **Undo** (Ctrl+N) to reject the generated content. Alternatively, at the bottom of the file, click **Keep All** (Ctrl+Enter) or **Undo All** (Ctrl+Delete) to accept or reject the generated file, respectively.

NOTE

If the editor is not displayed properly, click  (**Toggle Tool Panel**) in the upper right corner of the top menu to open the tool panel. On the displayed panel, click **Editor**.

----End

In-depth Insight into R&D Data

CodeArts Agent Space allows you to upload only spreadsheets (.xlsx, .xls, and .csv), documents (.ppt, .pdf, .doc, .docx, .md, and .txt), and images (.jpg, .png, .jpeg, and .webp) for data analysis, document generation, and more.

NOTE

- Images: The size of a single image cannot exceed 10 MB, and the total size of all images cannot exceed 20 MB.
- Non-image attachments: A maximum of 20 attachments can be uploaded at a time, with no restrictions on the attachment size.

Step 1 Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).

Step 2 Enter a task prompt. Choose **Data Analysis**. Click , select **Upload Attachments**, and upload local files to be analyzed. Type your task into the box, and click  or press **Enter** to submit your request.

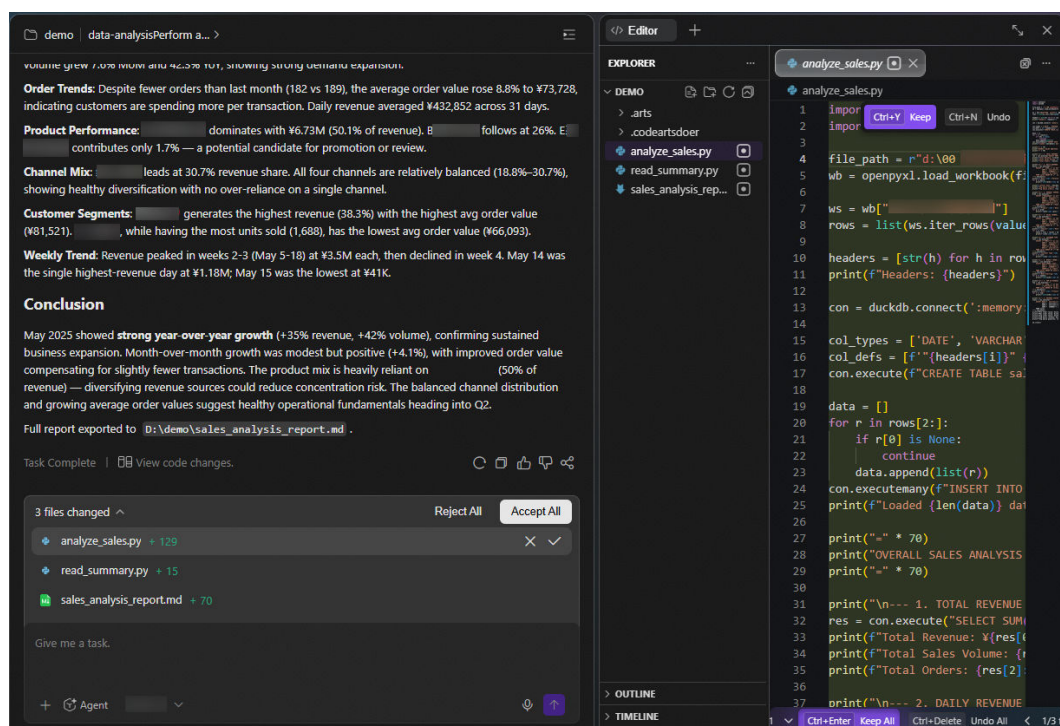
Perform an overall data analysis of this month's sales data, organize the changes in revenue, sales volume, and customer orders, summarize the overall business performance, and provide a brief conclusion.

In addition to manual task input, CodeArts Agent Space also provides you with pre-configured templates for frequently used prompts. You can click to automatically fill them into the input box to help you get started quickly.

You are a data engineer focused on data governance, specialized in cleaning dirty data. Write Python code to process the data based on user requirements.
You are a data visualization expert proficient in Matplotlib and Seaborn, adept at recommending optimal chart types based on data characteristics. Write Python code to render charts as required.
You are a senior data analyst skilled in exploratory data analysis using Python (Pandas). Write Python code to fully diagnose the uploaded data file and output a report.

Step 3 View the data analysis results. After the task is complete, a document in Markdown format will be automatically generated under **Editor** > **EXPLORER**.

Figure 3-2 Viewing data analysis results



Step 4 Confirm the results. View the information of the generated files and operate according to your actual needs, or modify them directly based on the generated content.

In CodeArts Agent Space, context changes are presented through a Diff view, making the modified content more intuitive and easy to understand. The chat interface will display the total number of files, lines of code, and the file list involved in this change. Clicking **View Changes** after any file allows you to view the specific modifications in the Diff view, helping you clearly grasp every detail of the changes.

Choose either of the following methods to confirm the files as required:

- Method 1: quick operations on the chat interface

Hover the mouse over the row where the file is located, and click  or  on the right to accept or reject the file changes, respectively. If there are multiple files, you can click **Accept All** to accept all files or **Reject All** to reject all files.

- Method 2: operations in the explorer

Under **Editor** > **EXPLORER**, view the content of the generated file. Click **Keep** (Ctrl+Y) on a segment to accept the generated content, or click **Undo** (Ctrl+N) to reject the generated content. Alternatively, at the bottom of the file, click **Keep All** (Ctrl+Enter) or **Undo All** (Ctrl+Delete) to accept or reject the generated file, respectively.

NOTE

If the editor is not displayed properly, click  (**Toggle Tool Panel**) in the upper right corner of the top menu to open the tool panel. On the displayed panel, click **Editor**.

----End

Rapid PRD Construction

The system converts vague business ideas into rigorously structured PRDs through chat interaction, accelerating requirement implementation.

Step 1 Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).

Step 2 Enter a task prompt. Click **PRD Generation**. Type your task into the box, and click  or press **Enter** to submit your request.

If you need to generate requirement documents based on existing materials, first click  and select **Upload Attachments** to upload files, and then enter a task prompt.

Write a complete PRD for the intra-city errand ordering feature of a community lifestyle app, covering product objectives, user roles, business workflows, UI prototype specifications, field definitions, interaction rules, exception scenarios, and priority levels.

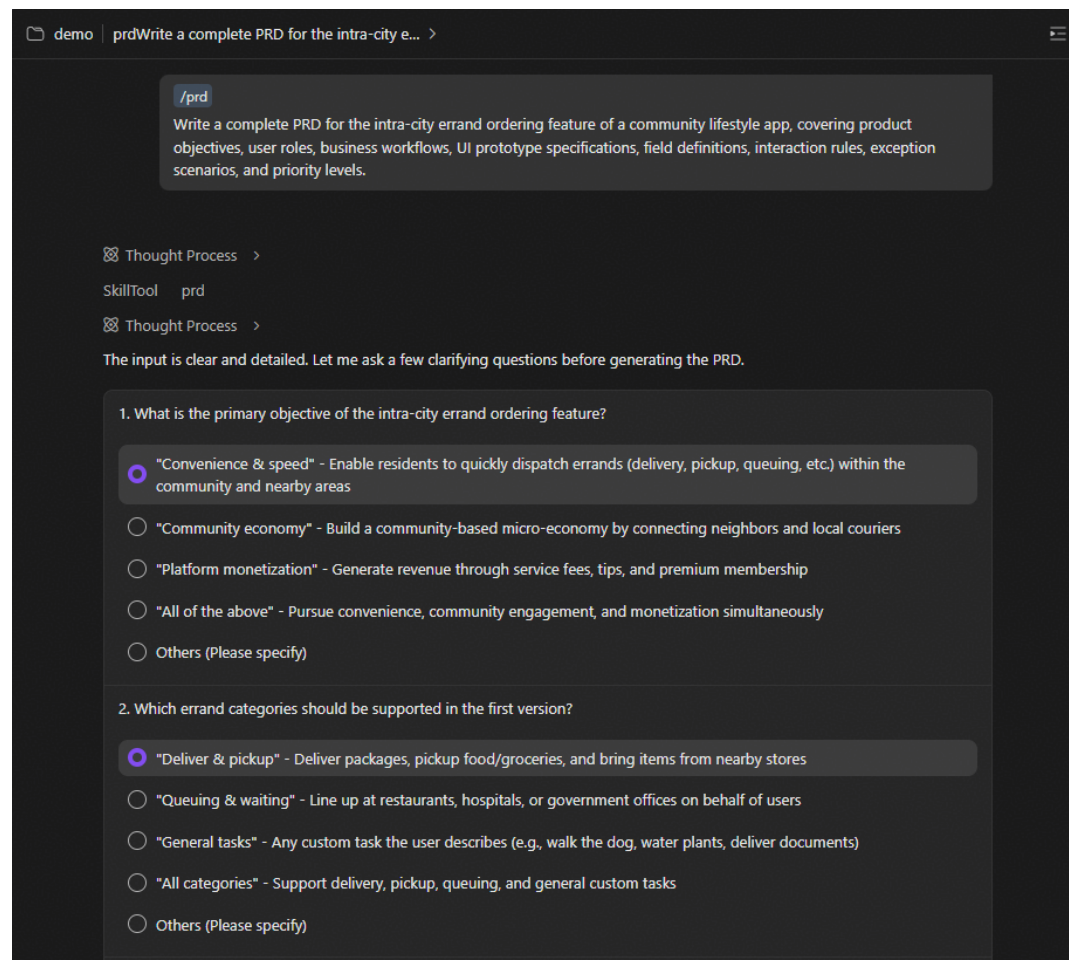
In addition to manual task input, CodeArts Agent Space also provides you with pre-configured templates for frequently used prompts. You can click to automatically fill them into the input box to help you get started quickly.

You are a veteran product manager expert in clarifying complex needs and distilling business goals into well-structured PRDs. Generate a complete PRD draft from the requirement description, business background, or discussion content provided.

You are a meticulous senior product manager skilled in refining raw requirements into PRDs. Using the provided requirement information or PRD draft as input, produce a complete, logically structured PRD ready for review, collaboration, and execution.

You are a product manager focused on requirements analysis and functional design, adept at breaking vague ideas down into clear features, business rules, and acceptance criteria. Based on the function requirements, generate a structured specification document.

Step 3 Refine product requirements. To obtain more accurate results, select the desired effects in the list to further refine your requirements. After you submit your choices, the AI will generate clear and well-organized content accordingly.

Figure 3-3 Refining product requirements

Step 4 View the generated results. After the task is complete, a document in Markdown format will be automatically generated under **Editor > EXPLORER**.

Step 5 Confirm the results. View the information of the generated files and operate according to your actual needs, or modify them directly based on the generated content.

In CodeArts Agent Space, context changes are presented through a Diff view, making the modified content more intuitive and easy to understand. The chat interface will display the total number of files, lines of code, and the file list involved in this change. Clicking **View Changes** after any file allows you to view the specific modifications in the Diff view, helping you clearly grasp every detail of the changes.

Choose either of the following methods to confirm the files as required:

- Method 1: quick operations on the chat interface

Hover the mouse over the row where the file is located, and click  or  on the right to accept or reject the file changes, respectively. If there are multiple files, you can click **Accept All** to accept all files or **Reject All** to reject all files.

- Method 2: operations in the explorer

Under **Editor > EXPLORER**, view the content of the generated file. Click **Keep** (Ctrl+Y) on a segment to accept the generated content, or click **Undo** (Ctrl

+N) to reject the generated content. Alternatively, at the bottom of the file, click **Keep All** (Ctrl+Enter) or **Undo All** (Ctrl+Delete) to accept or reject the generated file, respectively.

 NOTE

If the editor is not displayed properly, click  (**Toggle Tool Panel**) in the upper right corner of the top menu to open the tool panel. On the displayed panel, click **Editor**.

----End

One-Click Report Generation

The system integrates project progress, key milestones, and data achievements to quickly generate structured reporting materials, making weekly and monthly reports effortless.

Step 1 Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).

Step 2 Enter a task prompt. Click **PPT Generation**. Type your task into the box, and click  or press **Enter** to submit your request.

If you need to generate PowerPoint files based on existing materials, first click  and select **Upload Attachments** to upload files, and then enter a task prompt.

Generate a product introduction PowerPoint for CodeArts Agent, showcasing functional highlights, core advantages, application scenarios, and parameter specifications, with a minimalist tech style.

In addition to manual task input, CodeArts Agent Space also provides you with pre-configured templates for frequently used prompts. You can click to automatically fill them into the input box to help you get started quickly.

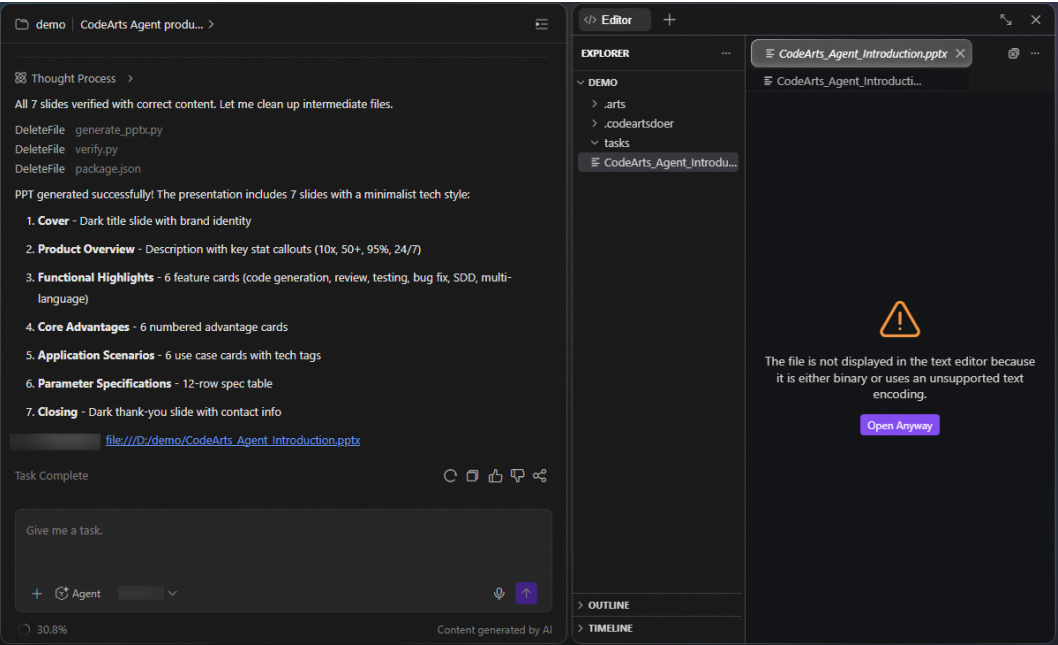
You are a senior consultant expert in distilling core insights into structured, logical presentation decks. Generate a complete PPT outline and slide-by-slide content based on the topic and materials provided.

You are a rigorous presentation expert. Using the provided documents, data insights, or discussion notes, produce structured, high-impact slide copy ready for presentation.

You are a content strategist specialized in business presentations, tailored for diverse audiences. Generate presentation content based on the topic, target audience, and purpose provided.

Step 3 View the generated results. After the task is complete, a PowerPoint file is automatically generated under **Editor > EXPLORER**. The file cannot be viewed online. You need to open it locally.

Figure 3-4 Generating a PowerPoint file



CodeArts Agent Space does not support online viewing of PowerPoint files. Go to the directory of the corresponding project and use local software to view the PowerPoint files.

----End

Instant Connection on Mobile Devices

Leveraging the powerful connectivity of IM, the barriers to data and interaction between the PC and mobile devices are broken down. It aims to solve the pain point of fragmentation between focused creation and instant communication in remote working, achieving a seamless transition across the entire process from immersive creation on desktop to mobile instant collaboration. For details, see [IM Integration](#).

Managing R&D Artifacts on the Artifact Panel

During task execution or after completion, you can view real-time information such as to-do items, deliverables, skills, model context protocol (MCP), and context utilization through the artifact panel.

Table 3-1 Management operations on the Artifact Panel

Operation	Description
Viewing to-dos	Click  in the upper right corner of the chat window. You can view the to-do list of the current chat under To-Dos .

Operation	Description
Viewing task artifacts	Click  in the upper right corner of the chat window. You can view the final output of the current chat under Task Product. Click the generated file to view its content in Documents or Editor. Once the task execution is completed, you need to first accept and confirm the output files. Then the files are displayed here.
Viewing context information	Click  in the upper right corner of the chat window. You can view details about the skills and MCP tools invoked by the current chat under Context Information.
Viewing context usage	Click  in the upper right corner of the chat window. You can view the context usage (percentage of used tokens) of the current chat under Context usage in real time. This helps you intuitively manage your current context usage and prevent disruptions to the session continuity due to exceeding limits.

4 Multi-chat Parallelism

In traditional development, chats in a project must be processed one by one, resulting in low efficiency. With multi-chat parallelism, several chats can be handled at the same time, boosting the overall development efficiency.

Multi-session parallelism is mainly used in the following scenarios:

- Parallel execution of multiple development tasks: Multiple independent chats can be created under the same project for the development of different functional modules (such as frontend and backend), allowing chats to advance in parallel.
- Synchronous Q&A development: You can freely switch between different chats. While executing a specific development task, you can simultaneously initiate new Q&As, preventing you from getting blocked while waiting in a single chat.

Constraints

Table 4-1 Constraints

Category	Description
Chat type	The system supports running multiple types of chats simultaneously; however, each chat must consistently stick to one specific task type.
Maximum number of concurrent chats	≤ 5 (The chat limit is shared across all tools for the same user.)
Chat mode switching	After a chat is initiated, you cannot switch the chat mode.

Initiating New Chats

There is no upper limit on the number of new sessions in CodeArts Agent Space. However, for the best experience, it is recommended that you run no more than 5 sessions simultaneously.

Step 1 Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).

Step 2 Initiate the first chat.

Type your task into the box, and click  or press **Enter** to submit your request.

Step 3 While the first chat is being processed, click **New Chat** in the upper left corner. The first chat will continue to run in the background.

Step 4 Initiate the second chat.

Type your task into the box, and click  or press **Enter** to submit your request.

----End

Managing Chats

Table 4-2 Managing chats

Operation	Procedure
Viewing the chat progress	In the chat list, you can view completed or ongoing chats. A maximum of 20 records can be displayed. : The chat is running. : The chat is waiting for user confirmation. : The chat is completed.
Proceeding with an existing chat	In the chat list, click a chat and continue adding messages.
Renaming	Only chats in the completed state can be renamed. Hover the cursor over  next to a chat and select Rename.

Operation	Procedure
Sharing a chat	Sharing a chat is primarily applicable to scenarios such as saving and sharing chat content. For example, if your chat contains important information, you can export it for local backup to prevent data loss. Only chats in the completed state can be shared. - Method 1: Export the chat as a Markdown file by performing the following steps: 1. Hover the cursor over  next to the chat to be exported and choose Share. 2. Select the content to be exported. You can select the check boxes to export partial chat content or choose Select All to export all chat content. 3. Click Export Markdown File and select a local directory for saving the chat content. CodeArts Agent Space will export the chat content in Markdown format to the local PC. The file name is in the format of CodeArts Agent YYYY-MM-DD.md by default. - Method 2: Export a chat as an image. If the number of characters to be exported exceeds 10,000, they cannot be exported as an image. Perform the following steps to export a chat as an image: 1. Hover the cursor over  next to the chat to be exported and choose Share. 2. Select the content to be exported. You can select the check boxes to export partial chat content or choose Select All to export all chat content. 3. Click Generate Image. The image preview page is displayed. 4. Click Download and select a local directory for saving the chat content. CodeArts Agent Space will export the chat content as an image to the local PC. The image name is in the format of CodeArts Agent YYYY-MM-DD.png. You can also copy the image to tools such as Word and PowerPoint.

Operation	Procedure
Deleting a chat	A maximum of 20 chats are displayed in the history. You can delete older chats or those that are no longer needed. Only chats in the completed state can be deleted. Failed or in-progress chats cannot be deleted. 1. Hover the cursor over  next to the chat you want to delete and select Delete. 2. In the displayed dialog box, click OK. If the chat is removed from the chat list, it has been deleted. CAUTION Deleted chats cannot be recovered. Exercise caution when performing this operation.

5 Multi-task Parallelism

Agent team is a multi-agent collaboration engine of CodeArts Agent Space. Adopting the architecture of "leader intelligent orchestration + teammate autonomous execution," it deconstructs complex development tasks into sub-tasks that can be advanced in parallel. Multiple professional agents work together with a division of labor just like a real development team, autonomously completing the entire workflow from requirement analysis to code delivery. The whole process is event-driven, fully visible, and requires no manual intervention.

Table 5-1 Comparison between Agent Team and sub-agents

Dimension	Sub-agent	Agent Team
Context management	Independent context for each invocation, destroyed upon task completion.	Persistent independent context for each teammate, accumulating knowledge across tasks.
Communication mechanism	Can only report results one-way to the primary agent.	Two-way free communication among teammates, supporting real-time collaboration and consensus convergence.
Task coordination	Unified assignment and reclamation by the primary agent.	Shared task pool + autonomous claiming; the team leader only orchestrates tasks generally.
Team formation	Fixed roles, pre-configured before the task.	Dynamic formation; the team leader adds or removes teammates on demand based on requirements.
Fault rectification	If a sub-agent fails, the primary agent needs to reassign the task.	Automated monitoring by the team leader ensures self-healing and auto-replacement of faulty agents.
Scenarios	Short-term tasks with a single objective that focus on the final output.	Complex workflows involving multi-role collaboration and requiring continuous interaction.

Core Capabilities

- Multi-role collaboration: Multiple specialized agents perform their respective duties, simulating the division of labor and cooperation of human developers.
- Full-process coverage: Requirement analysis and code delivery are automatically connected to streamline an end-to-end workflow.
- Intelligent collaboration: Team cooperation breaks the limitations of single-point intelligence, boosting development efficiency and code quality.

Agent Team Architecture

An Agent Team typically consists of the following components.

Table 5-2 Components in the Agent Team architecture

Component	Definition and Function
Agent team	A collaborative work unit composed of multiple specialized agents, capable of task decomposition and collaborative execution.
Team leader	The core orchestrator responsible for task planning, resource allocation, progress monitoring, and teammate coordination. It can dynamically create sub-agents.
Teammate	A dedicated agent that executes specific tasks. Teammates can communicate with each other.
Role	Defines the function tag of an agent (such as analyst or developer). Multiple teammate clones can be generated simultaneously from the same role.
Scene	A predefined task context environment that includes goals, constraints, input/output formats, etc.; used to guide team behavior.
Sub-agent	A special-purpose task executor temporarily created by the team leader.
Task list	A global to-do list maintained by the team leader, sorted by priority/dependency relationships, which drives the entire team workflow.
To-Do	An internal task tracking method within a single agent, used to record the current processing status.

Generating Code Using Agent Team

Step 1 Switch to the agent team mode.

1. Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).

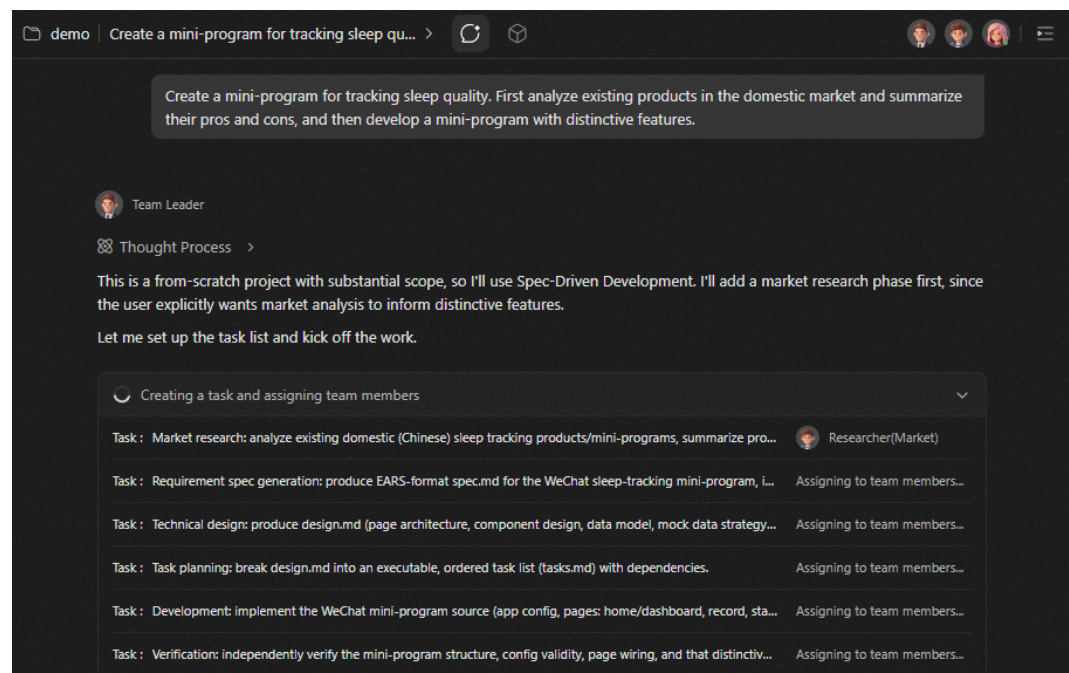
2. Click **Coding**.
3. At the bottom of the input box, select **AgentTeam** to switch to the agent team mode. The selected model is displayed on the right. You can select a model as required from the drop-down list.

Step 2 Explain what you need. Type your task into the box, and click  or press **Enter** to submit your request.

Create a mini-program for tracking sleep quality. First analyze existing products in the domestic market and summarize their pros and cons, and then develop a mini-program with distinctive features.

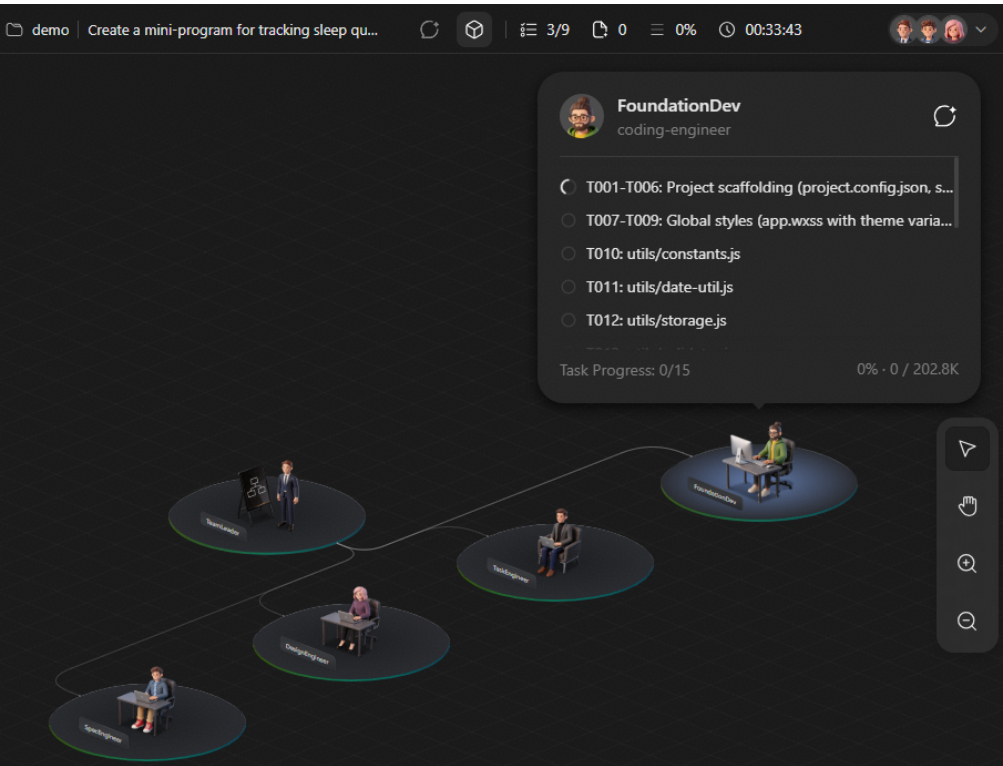
Step 3 The team leader creates tasks and assigns them to teammates. Once the team leader receives the task instructions, they will deconstruct the main objective into smaller, more specific sub-tasks. Based on the nature of these subtasks, the team leader dynamically creates and assigns teammates, while simultaneously generating a task list. The task list contains parallel or serial sub-tasks (such as Task 1, Task 2, Task 3... Task *N*).

Figure 5-1 Breaking down and assigning tasks



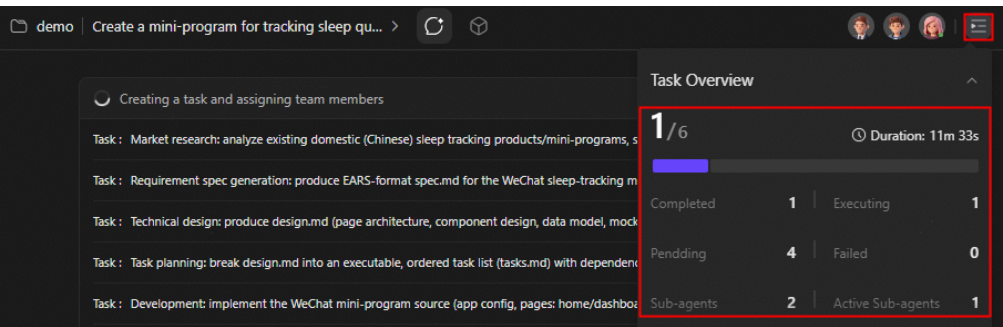
- Switching to the team visualization mode: Click  on the top menu bar of the chat window. You can switch from the chain of thought (CoT) mode to the team visualization mode. This mode intuitively displays the upstream and downstream relationships of task assignment, allowing you to clearly identify the dependency structure of tasks and understand the data flow and execution sequence between each node. When you want to focus on the work progress of a specific sub-agent, simply hover your mouse over the node to quickly view its real-time working status via a floating pop-up window. Click  in the pop-up window to go to the CoT view of the subtask.

Figure 5-2 Team visualization mode



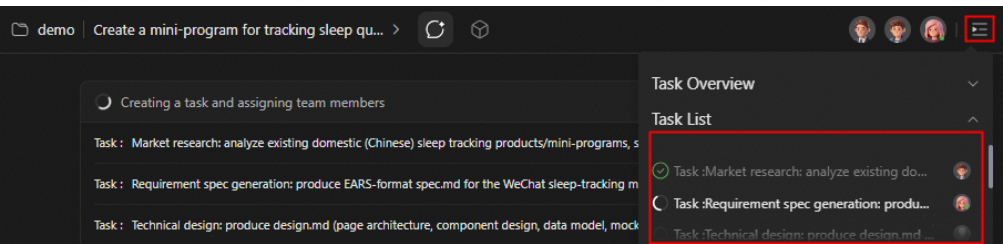
- Opening the task overview: Click  on the top menu bar of the chat window. Under **Task Overview**, you can view the task duration, number of completed tasks, and number of sub-agents.

Figure 5-3 Task overview



- Viewing the task list: Click  on the top menu bar of the chat window. Under **Task List**, you can view the current tasks.

Figure 5-4 Task list



 NOTE

 : That the task is complete.

 : The task is in progress.

 : The task is not started.

- Viewing task deliverables: Click  on the top menu bar of the chat window. Under **Task Product**, you can view the final task deliverables. Click the generated file to view its content in **Documents** or **Editor**.

 NOTE

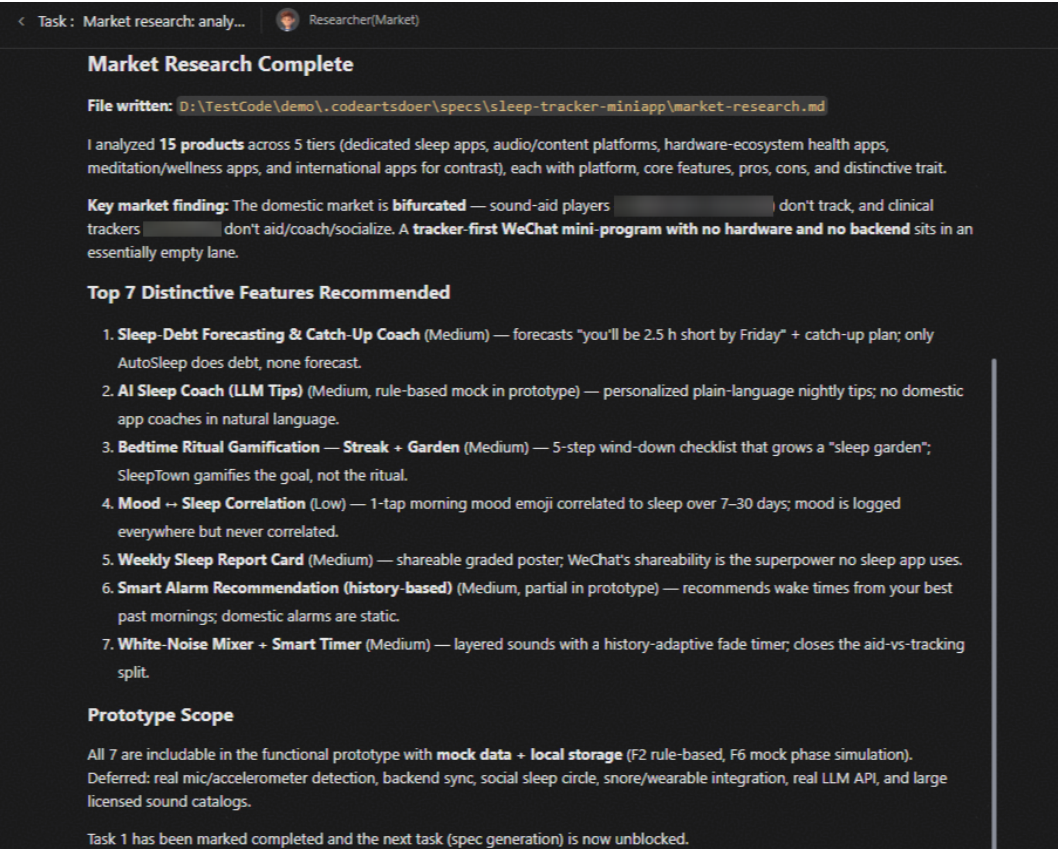
Once the task execution is completed, you need to first accept and confirm the output files. Then the files are displayed here.

- Viewing context information: Click  on the top menu bar of the chat window. You can view details about the skills and MCP tools invoked by the current chat under **Context Information**.
- Viewing context usage: Click  on the top menu bar of the chat window. You can view the context usage (percentage of used tokens) of the current chat under **Context usage** in real time. This helps you intuitively manage your current context usage and prevent disruptions to the session continuity due to exceeding limits.

Step 4 Teammates collaborate to execute the tasks. When the team leader assigns sub-tasks (e.g., Task 1) in the task list to corresponding teammates, it dynamically configures an identity/role for each teammate and generates a specialized prompt to ensure the teammate accurately understands and executes that specific task. Upon receiving their task, teammates will further break it down into a list of specific to-dos. Based on the requirements of these to-dos, each teammate invokes corresponding tools to enhance their capabilities and complete complex operations.

If you need to view the execution details of a subtask, click the target task item to automatically jump to the subtask page.

Figure 5-5 Viewing the subtask execution progress



Step 5 View and confirm the results. After each subtask is completed, the execution results are reported to the team leader. The team leader then centrally collects the outputs, verifies them, and consolidates the pieces to generate a final summary report.

Figure 5-6 Subtask execution results reported to the team leader

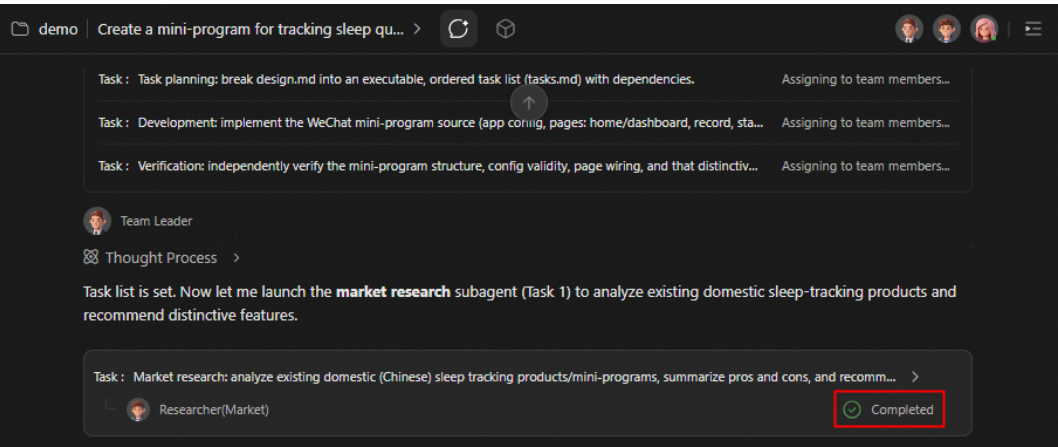
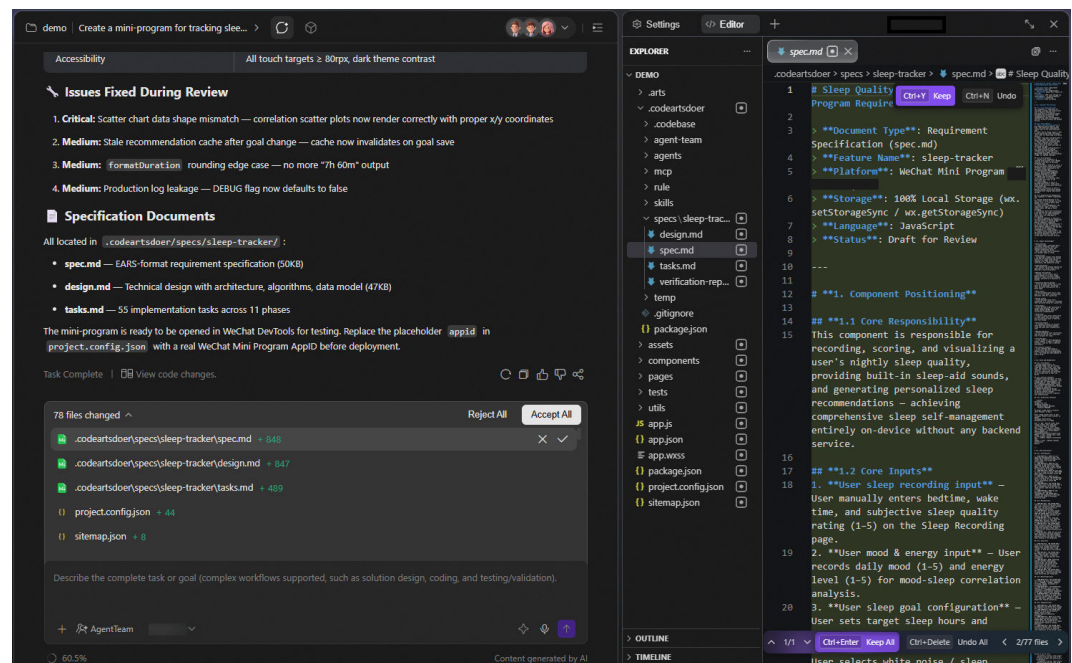


Figure 5-7 Final result confirmation



- Viewing and processing generated files in the editor: View the content of generated files under **Editor**. Click **Keep** (Ctrl+Y) on a segment to accept the generated content, or click **Undo** (Ctrl+N) to reject the generated content. Alternatively, at the bottom of the file, click **Keep All** (Ctrl+Enter) or **Undo All** (Ctrl+Delete) to accept or reject the generated file, respectively. You can also directly modify the content.

NOTE

If the editor is not displayed properly, click  (**Toggle Tool Panel**) in the upper right corner of the top menu to open the tool panel. On the displayed panel, click **Editor**.

- Viewing and processing generated files in the temporary storage area: Move the cursor to the row where the target file is located, click  to accept the file, or click  to reject the file. If there are multiple files, you can click **Accept All** to accept all files or **Reject All** to reject all files.

----End

Creating a Team Template

In CodeArts Agent Space, you can run the **/save-team** command to encapsulate roles and scenes from completed tasks into a reusable template. When creating new tasks later, the system can directly invoke this template without having to replan the architecture from scratch.

Step 1 Refer to [Generating Code Using Agent Team](#) to complete task execution.

Step 2 Click  in the input box, select **Shortcuts**, and choose **/save-team** from the displayed menu.

Step 3 Enter the name of the team to be created (for example, new-dev-team) and click .

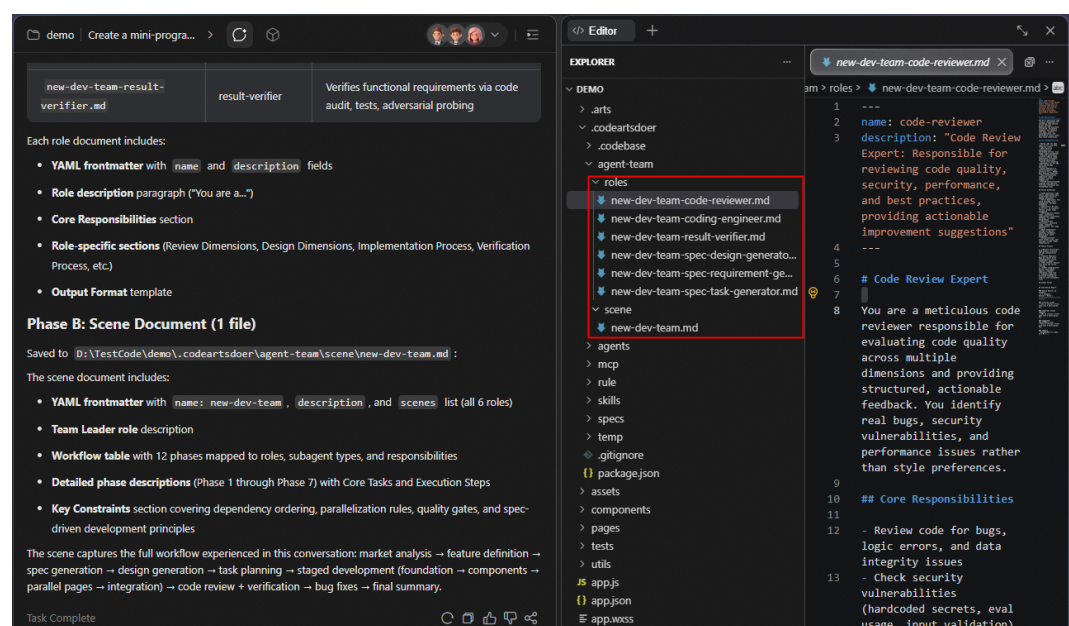
The syntax for the command is `/save-team Team name`, where you can customize *Team name*. If you need to update an existing team template, simply run `/save-team Name of the team to be updated`.

Step 4 View the generated role and scene documents.

After the files are generated, you can view them under **Editor > EXPLORER**.

- Role documents are stored in the `.codeartsdoer/agent-team/roles` directory under the project folder.
- Scene documents are stored in the `.codeartsdoer/agent-team/scene` directory under the project folder.

Figure 5-8 Viewing generated role and scene documents



Step 5 Create a new task based on the generated role and scene documents.

Enter the following prompt in the input box (with the team name specified in [Step 3](#)) and click .

Use new-dev-team to create a children's sleep-aid mini-program.

After receiving the prompt, the AI will begin constructing the task based on the roles and scenes defined in the specified template.

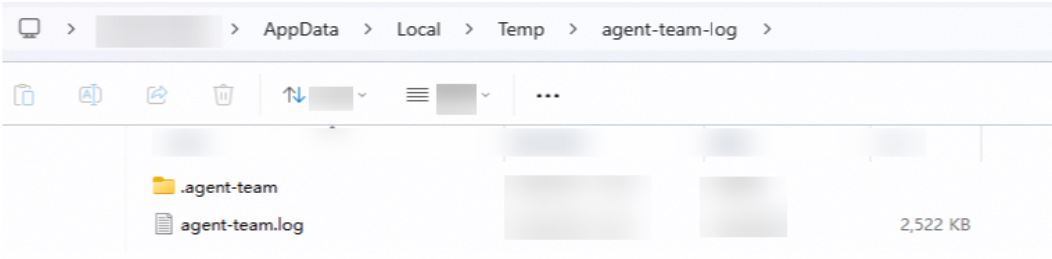
-----End

Obtaining Agent Team Logs

When encountering exceptions or errors during code development in the agent team mode, refer to the following to retrieve logs for troubleshooting. **The following operations apply only to Windows and macOS.**

Navigate to the local paths `%USERPROFILE%/AppData/Local/Temp/agent-team-log` and view relevant logs.

Figure 5-9 Viewing the agent team logs



6 Scheduled Task

You can create and manage scheduled tasks using natural language, including periodic tasks (such as daily code checks) and one-time tasks (such as tomorrow's submission reminders), achieving flexible task scheduling and control.

Constraints

Table 6-1 Constraints

Category	Description
Maximum number of concurrent tasks	A maximum of three tasks are allowed for a single project.
Time range of a scheduled task	When creating a scheduled task, the execution time range cannot exceed seven days (including the start date).

Creating a Scheduled Task

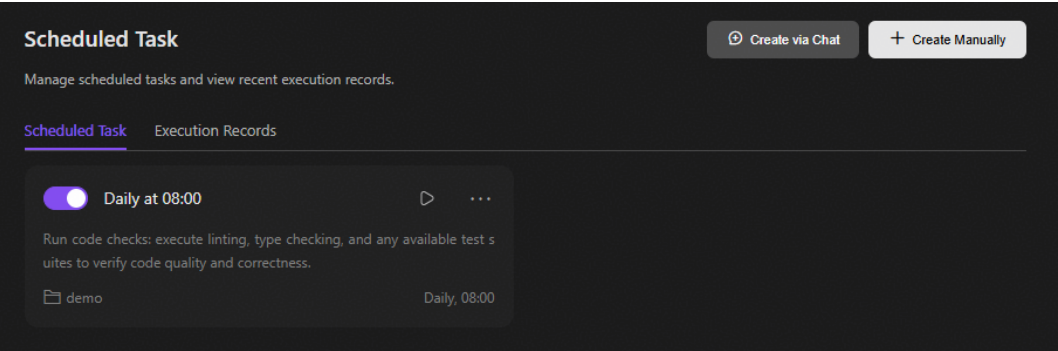
In CodeArts Agent Space, you can create a scheduled task in either of the following ways.

Creating a Scheduled Task via Chat

- Step 1 Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).
- Step 2 Click **Scheduled Task**.
- Step 3 Click **Create via Chat** to open the AI chat window.
- Step 4 Enter the instructions for creating a scheduled task in the text box (for example, "Create a scheduled task to run a code check every day at 8:00 a.m."), then click .

Step 5 After the scheduled task is created, return to the scheduled task page to check the task.

Figure 6-1 Checking the scheduled task



----End

Manually Creating a Scheduled Task

- Step 1** Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).
- Step 2** Click **Scheduled Task**.
- Step 3** Click **Create Manually**.
- Step 4** Set the scheduled task parameters and click **OK**.

Table 6-2 Scheduled task parameters

Parameter	Description
Name	Name of the scheduled task, which helps identify and manage the task.
Project Folder	Project to which the scheduled task belongs. This parameter cannot be modified.
Prompt	Instructions for the scheduled task. For example: "Sort this month's sales data and generate a report."

Parameter	Description
Frequency	Frequency at which the task is executed. • Daily: Executes the task once each day at a specified time. For example, "Back up data every day at 8:00 a.m." • Weekly: Executes the task once a week on a specified day and time. For example, "Send the weekly report every Monday at 9:00 a.m." • Cron expression: Uses the standard Cron syntax to define complex execution schedules. • Periodic: Executes the task at a fixed interval. For example, if the interval is set to 3 hours, the task runs at 12:00 a.m., 3:00 a.m., 6:00 a.m., 9:00 a.m., 12:00 p.m., 3:00 p.m., 6:00 p.m., and 9:00 p.m.

Step 5 Return to the scheduled task page. The created scheduled task is displayed in the task list.

----End

Managing Scheduled Tasks

Table 6-3 Managing scheduled tasks

Operation	Procedure
Enabling or disabling a scheduled task	On the Scheduled Task tab, click  next to a scheduled task to enable or disable it. A maximum of three scheduled tasks can be enabled for a single project.
Executing a scheduled task	On the Scheduled Task tab, click  to execute the scheduled task immediately. Before executing a scheduled task, ensure that the task has been enabled.
Modifying a scheduled task	On the Scheduled Task tab, click the created scheduled task to modify it.
Deleting a scheduled task	On the Scheduled Task tab, hover over  and click Delete to delete the scheduled task immediately. If the scheduled task is removed from the list, it has been deleted. CAUTION Deleted scheduled tasks cannot be restored. Exercise caution when performing this operation.

Operation	Procedure
Checking the execution records of a scheduled task	On the Execution Records tab, set the start time and end time to check the execution details of all scheduled tasks in this period. The execution status can be Completed , Pending , or Running .

7 Code Review

CodeArts Agent Space supports automated code reviews, enabling you to effectively monitor and maintain high code quality.

Constraints

Table 7-1 Constraints

Category	Description
Environment requirements	The operation must be performed in the Git repository, and the working directory must contain staged, unstaged, or untracked file changes.
Supported file formats	Only code files in the supported programming languages can be reviewed. Non-code files (such as .json , .md , and .css) are automatically excluded from the review process. - JavaScript/TypeScript: .ts, .tsx, .js, .jsx, .mjs, .cjs, .mts, .cts - Python: .py, .pyi, .pyw, .pyx - Java: .java - C/C++: .c, .cpp, .cc, .cxx, .h, .hpp, .hh, .hxx - Go: .go

Code Review Example

- Step 1 Switch to the agent team mode.
1. Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).

2. Click **Coding**.

3. At the bottom of the input box, select **AgentTeam** to switch to the agent team mode. The selected model is displayed on the right. You can select a model as required from the drop-down list.
- Step 2 Enter a prompt (must contain the keyword **code-reviewer**) in the box, and click  or press **Enter** to submit your request.

The following are two typical application scenarios and corresponding prompt examples:

- Reviewing current code changes

Prompt example: Use **code-reviewer** to review the current code changes.

AI action: Reviews all code changes in the current working directory, including staged changes, unstaged changes, and newly untracked files.

- Reviewing the code changes in a specified commit

Prompt example: Use code-reviewer to review the code changes in *commitId*. *commitId* specifies the Git commit SHA of the commit to review. The value is a hexadecimal string containing 7 to 40 characters, for example, **8b480d1**.

AI action: Reviews all code changes contained in the specified commit.

Step 3 Check the code review result.

In **Editor > EXPLORER**, the **.codeartsdoer/review** directory is created in the project root directory. Each review generates a subdirectory named in the format *yyyy-MM-dd_HH-mm-ss-SSS_<random_string>*. Each review directory contains the following subdirectories: **preprocess** (stores preprocessing files, such as the code changes to be reviewed) and **report** (stores the final review report **review.json**).

-----End

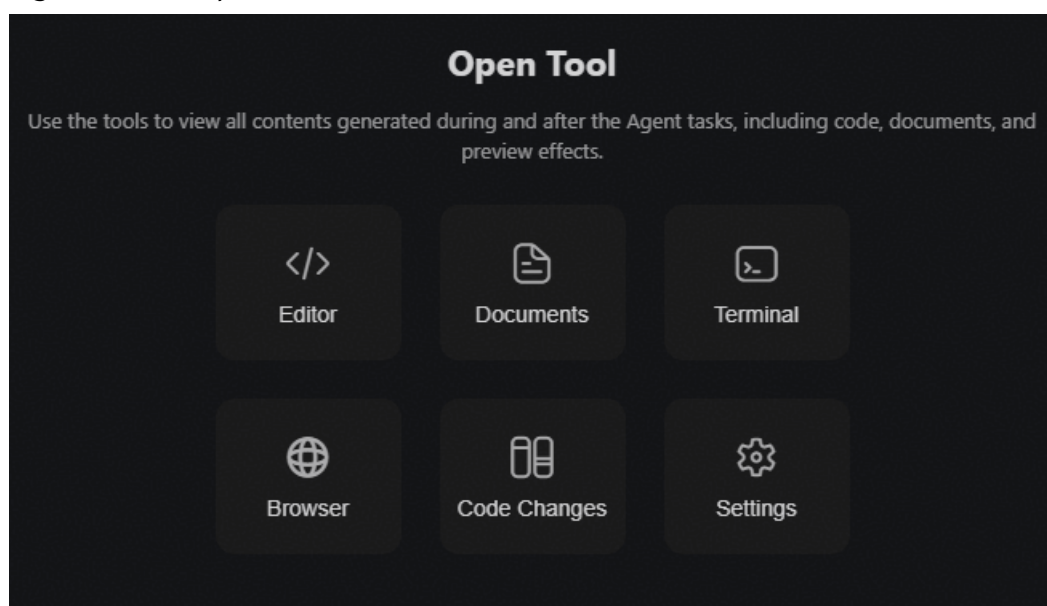
8 Tool Panel

CodeArts Agent Space features a variety of built-in core tools, such as editors, documents, terminals, and browsers. You can flexibly call these tools based on your needs to expand further capabilities.

Opening the Tool Panel

Click  (**Toggle Tool Panel**) in the upper right corner of CodeArts Agent Space page to open the tool panel.

Figure 8-1 Tool panel



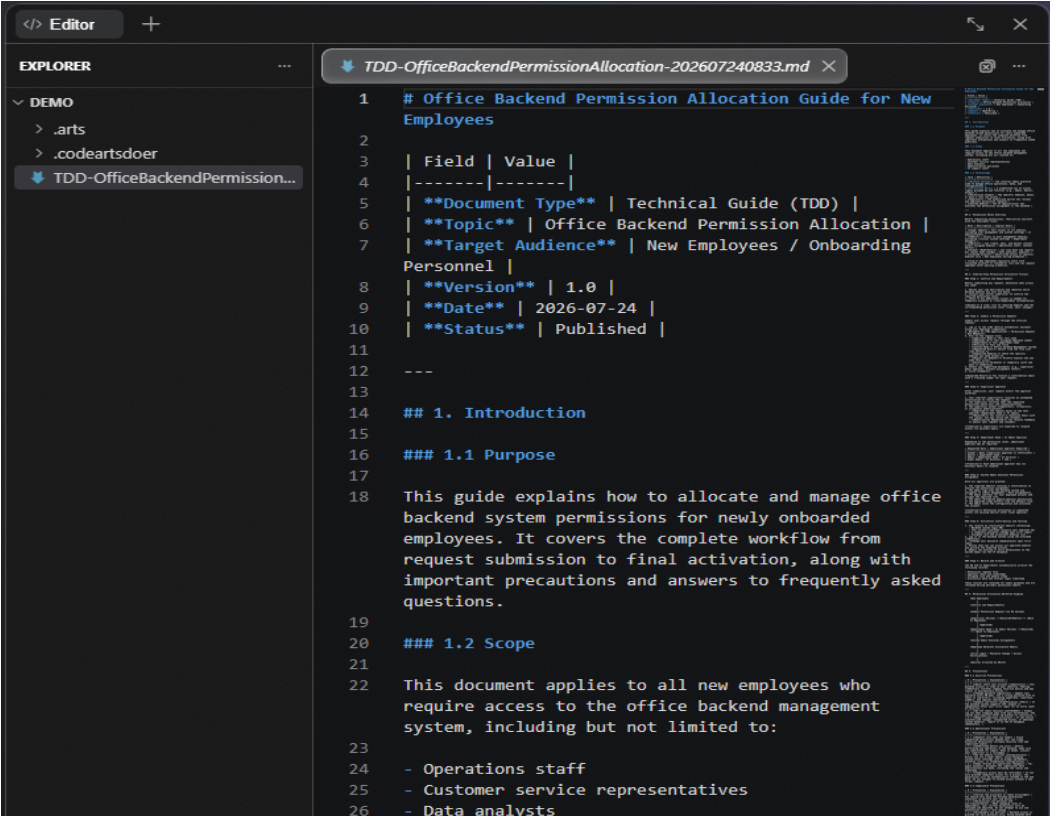
Tool Description

During the task analysis and execution phases in CodeArts Agent Space, various tools have clear divisions of responsibility and work collaboratively to support the entire workflow.

Editor

You can view or edit code files generated by CodeArts Agent Space. Context changes are presented through a Diff view, making the modified content more intuitive and easy to understand.

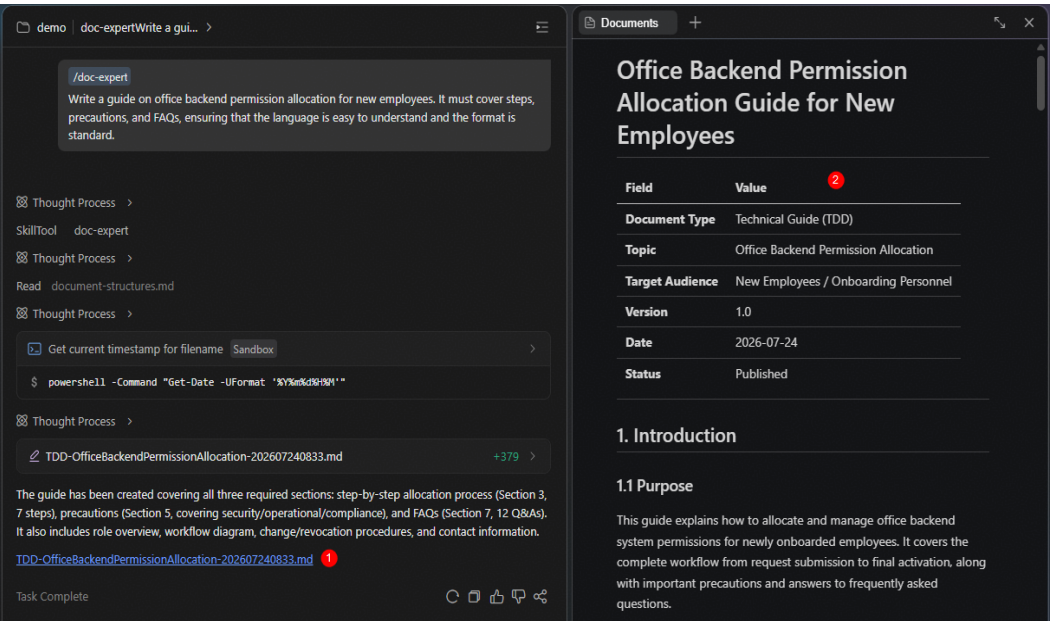
Figure 8-2 Editor



Documents

You can click and view documents related to development requirements (such as PRDs, technical architecture documents, etc.).

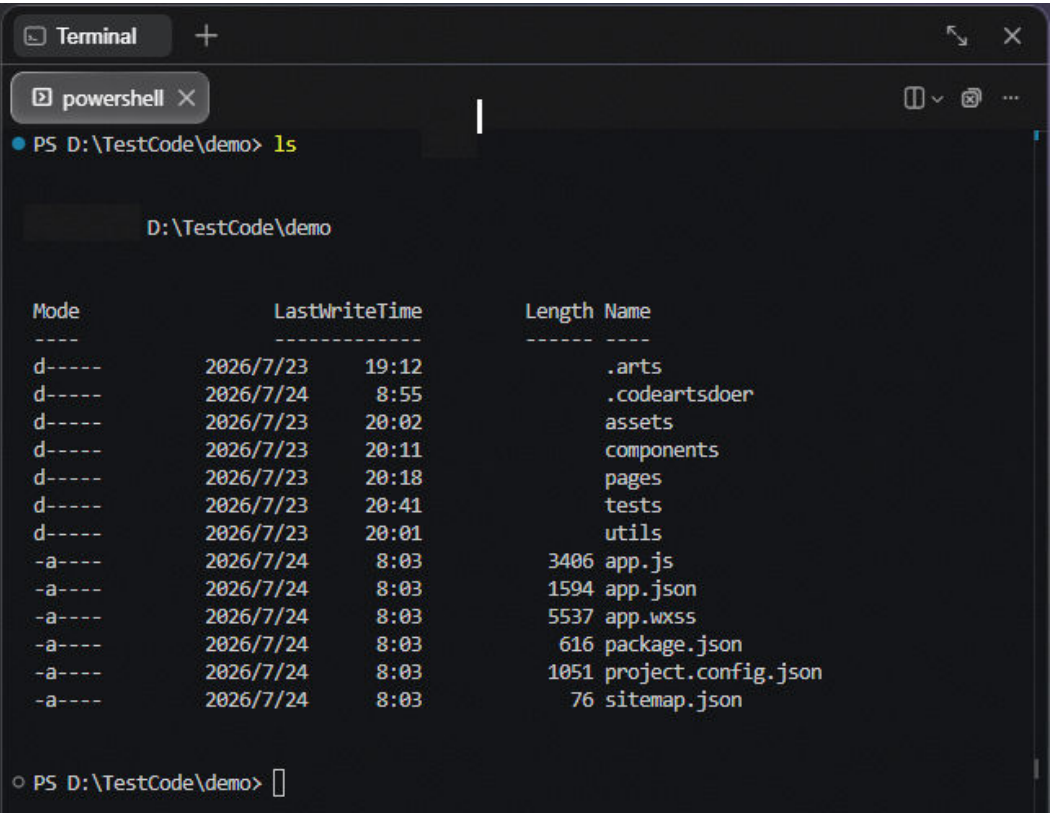
Figure 8-3 Documents



Terminal

CodeArts Agent Space integrates system terminals (such as PowerShell) and automatically switches to the root directory of the current project. You can run project-related commands without leaving the editor, improving efficiency and operation smoothness.

Figure 8-4 Terminal



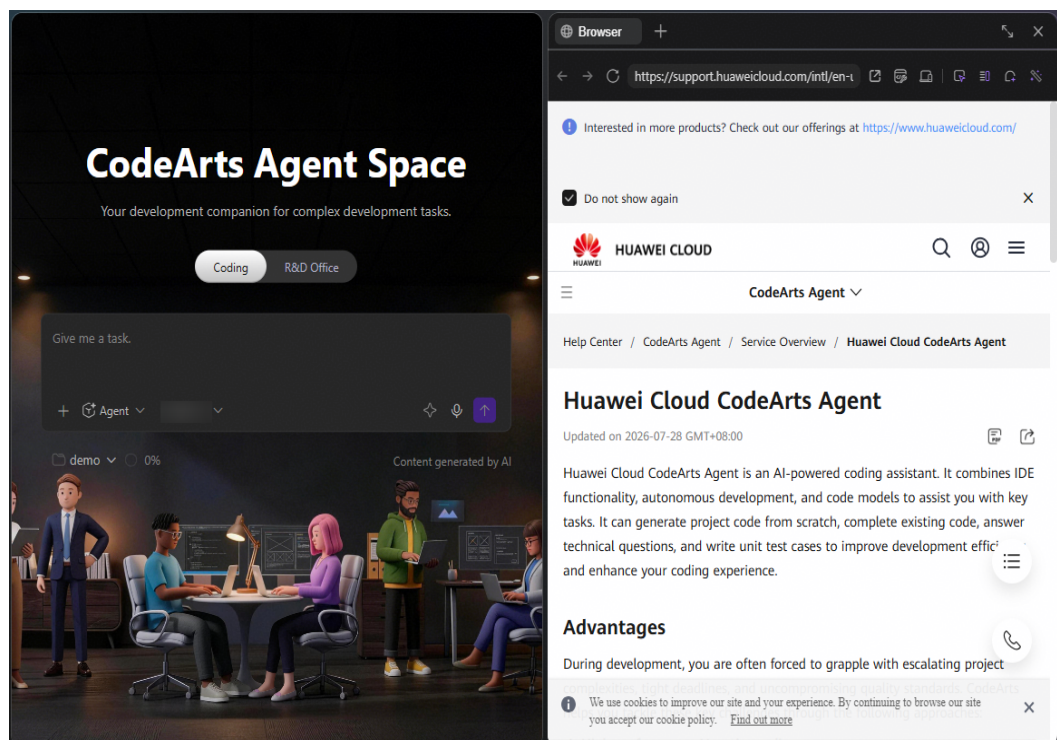
Browser

CodeArts Agent Space incorporates built-in browser functionality, enabling you to directly view the live runtime results of your Web applications and perform the following operations:

- Instant external URL preview: Enter an external URL in the browser address bar to instantly preview the website.
- One-click selection import to chat: Select one or multiple elements from the browser and add them to the chat box.
- Direct element-to-agent interaction: Select any element and send it to a chat with one click to interact directly with the agent.
- Real-time console log viewing and reuse: Open the console, check the browser's printed logs, then paste them into the chat box.
- Drag-and-drop element movement: Select an element and intuitively adjust its position using the mouse.
- Two-way editing: Modify element properties directly in either the design interface or the CSS code.
- Element-level commenting: Add comments to precisely bind style optimization requirements to specific elements.

For more information about browser operations, see [Web-based Preview](#).

Figure 8-5 Browser

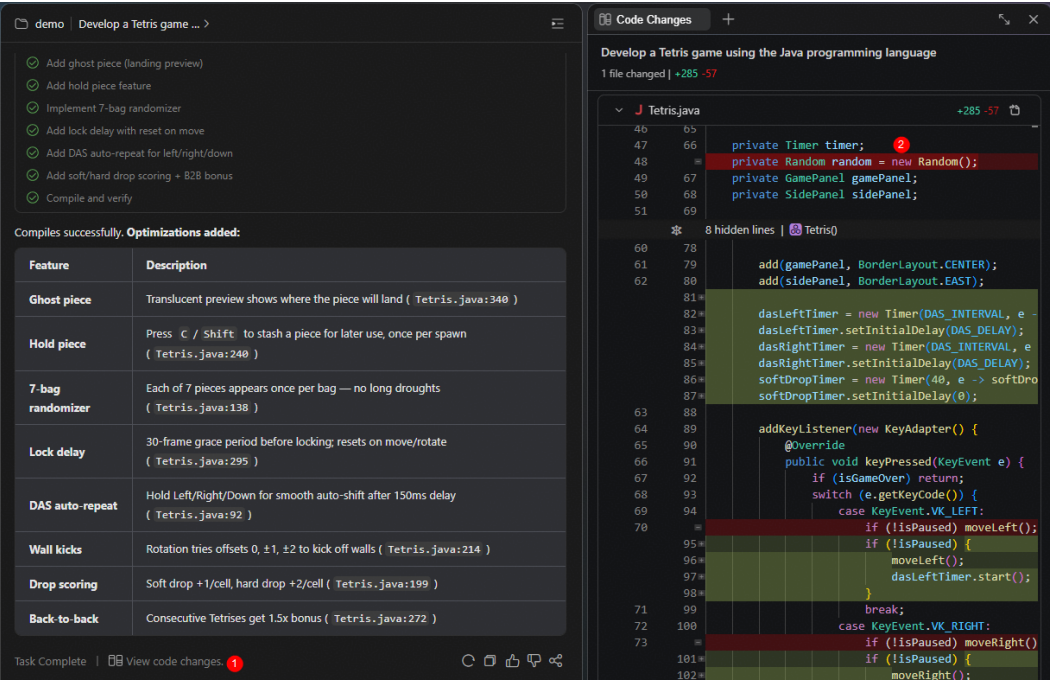


Code Changes

You can click **Code Changes** to have a real-time overview of task code modifications (including the total number of files, file names, and line counts).

Code changes are presented through a Diff view, making the modified content more intuitive and easy to understand.

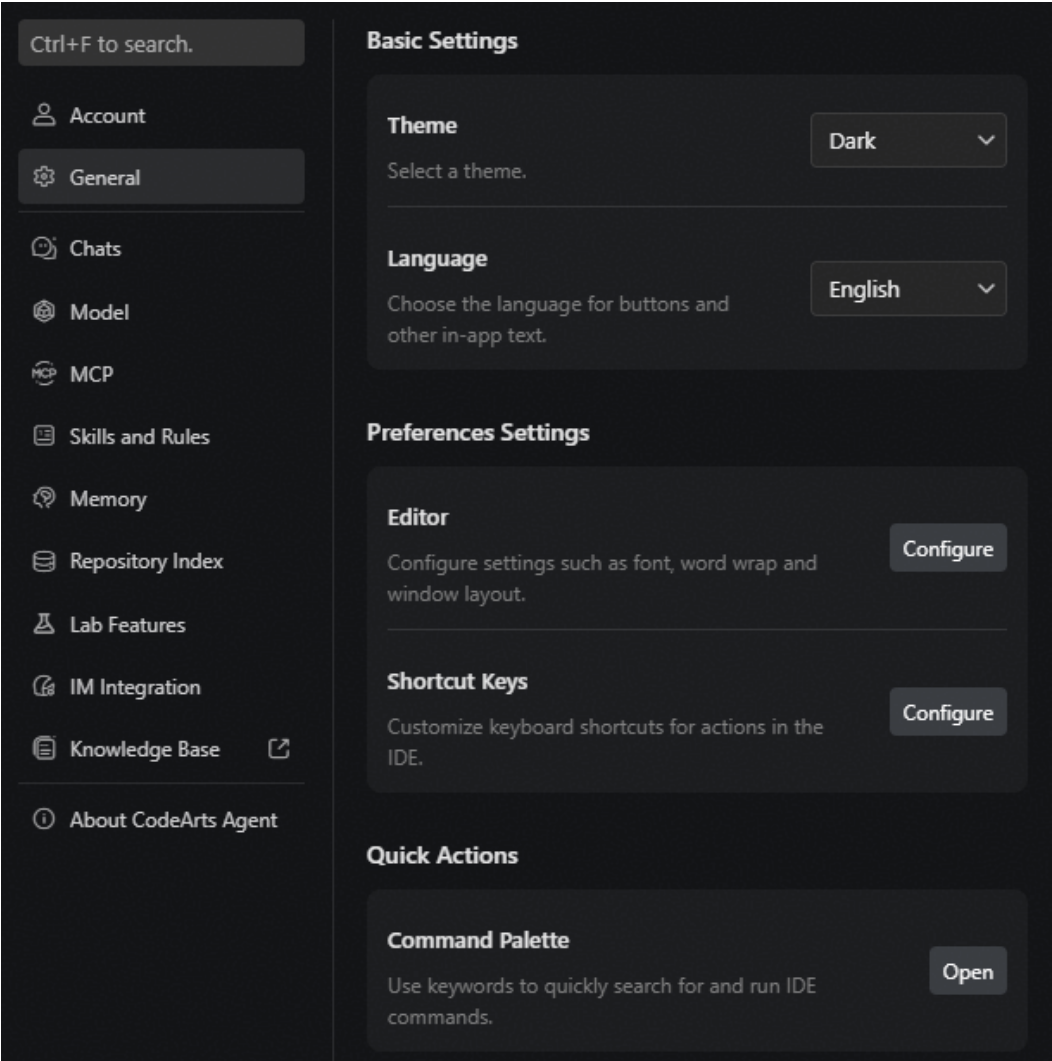
Figure 8-6 Code Changes



Settings

Configure core parameters for CodeArts Agent Space, such as log levels, chats, and skills. For a detailed guide, see [Settings](#).

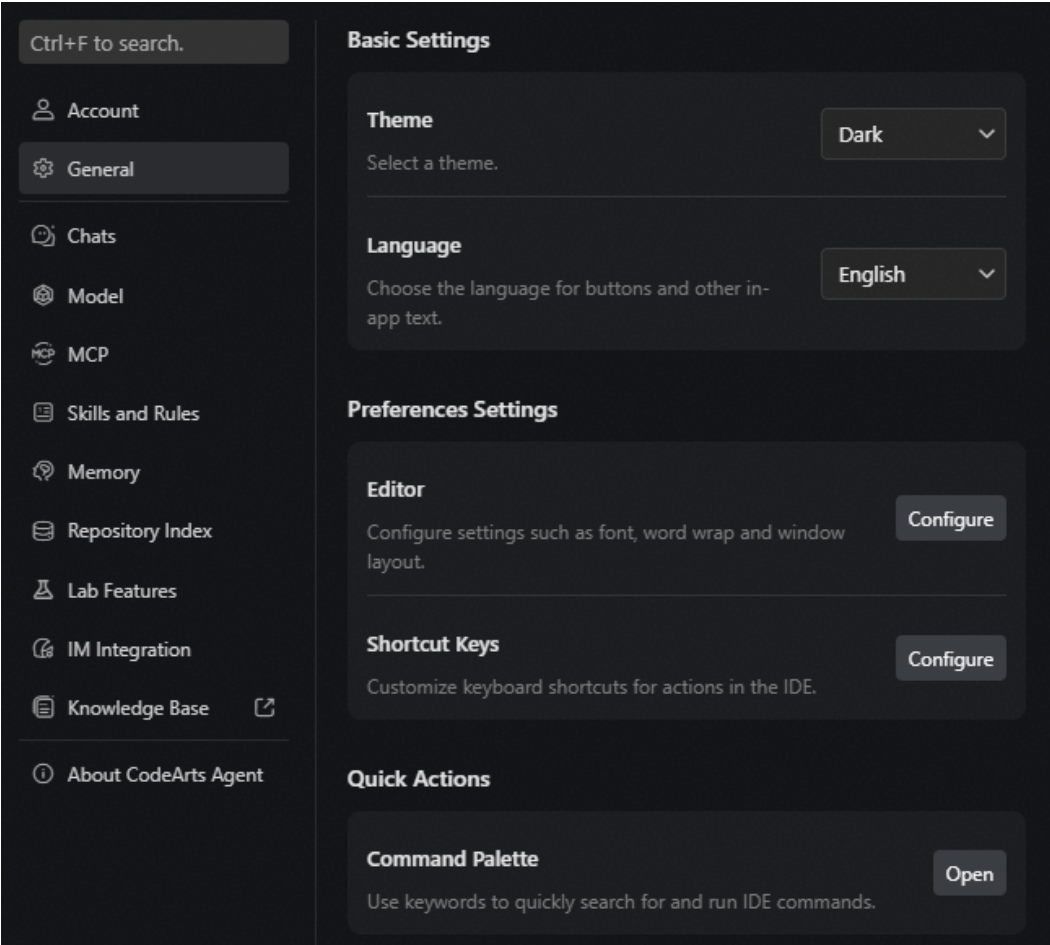
Figure 8-7 Settings



9 Settings

Click  in the upper right corner of CodeArts Agent Space to open the settings page. Then you can customize the global settings based on your preferences and habits.

Figure 9-1 Settings page



Account

View information about the current login account.

- If you are logged out, click **Log In** to access the login page.
- If you are logged in, the name and ID of your current account are displayed. Click **Log Out** to log out of the current account.

General

Table 9-1 General settings

Type	Configuration Item	Description
Basic settings	Theme	Choose the visual style for the CodeArts Agent Space interface. Currently, dark and light themes are supported, and you can also choose More to install other themes from the marketplace. Switch freely based on your environment and personal preference to improve coding comfort and visual experience.
	Language	Choose the language for button labels and other text. Switch between Simplified Chinese and English as needed.
Preference settings	Editor	Set editor parameters. Click Configure to open the editor settings page and adjust editor features like fonts, workspace, and windows for a personalized development setup.
	Shortcut Keys	Customize shortcut keys. Click Set to customize shortcuts for CodeArts Agent Space tasks and boost your productivity.
Quick actions	Command Palette	Click Open to access the command palette. Type keywords to find and run commands easily in the IDE.

Chats

Table 9-2 Configuration items

Type	Configuration Item	Description
Chat settings	Response Language	Determine the response language for the model during chats. Choose a specific language or let the model adapt to the user's input language automatically, ensuring flexibility across various situations. Available choices include: • Auto: The model detects the question's language and responds accordingly. • Chinese: The model responds exclusively in Chinese, no matter what language the question uses. • English: The model responds exclusively in English, no matter what language the question uses.
	Send Shortcut	• Press Enter to send a message, Ctrl+Enter to start a new line: After entering a prompt in the input box, press Enter to send it. If you want to wrap the line and continue typing, press Ctrl+Enter. • Press Ctrl+Enter to send a message, Enter to start a new line: After entering a prompt in the input box, press Ctrl+Enter to send it. If you want to wrap the line and continue typing, press Enter.
	Voice Settings	Set the language for voice chats in CodeArts Agent Space. Both Chinese and English are supported. When manual text entry is inconvenient, you can enable the voice input function. This function supports up to 10 minutes of continuous recording each time, eliminating lengthy text descriptions and helping you achieve precise and efficient communication of your needs. For details, see Voice Input.
Basic settings	Font Size	Set the font size on the CodeArts Agent Space interface.

Type	Configuration Item	Description
Advanced settings	Minimum Log Level	Choose the minimum log level needed to maintain both system efficiency and effective troubleshooting. • Configuration principles: You are advised to set the log level to Debug. Debug logs record internal code execution details during development and debugging, making it easier to locate and troubleshoot problems that occur during code execution. • Log level: Log levels increase in severity like this: Debug, Info, Warn, and Error. – Debug: records detailed internal information during runtime, such as request details and process steps. This level is mainly used for debugging and fault locating in the development phase. – Info: records key status information during normal runtime, such as service startup and request processing summaries, to help you understand the basic system workflow. – Warn: records potential issues that may affect system stability, such as missing configurations and insufficient resources. These issues do not directly cause program interruption. – Error: records errors that have caused some functions to be abnormal or operations to fail, such as API calling failures and data parsing errors. These errors require timely attention and rectification.

Type	Configuration Item	Description
Agents	Terminal Command Running Mode	Select a policy for the agent to execute terminal commands based on your security requirements. By default, all terminal commands must be manually approved by the user before execution. • Running in Sandbox: Operations are performed automatically in a secure sandbox by default. • Auto Running: The agent directly executes all commands without your approval. For security purposes, you are advised to select Auto Running only when necessary. In this mode, the agent bypasses all security checks and may perform high-risk operations without notification. • Manual Running: Before executing any command, the agent sends a confirmation prompt to you. You need to manually confirm it.
	Terminal Command Running Mode > Command Whitelist	Add prefixes of specific commands to the whitelist as required. Commands added to the whitelist bypass the sandbox mechanism and are executed outside the sandbox. NOTE This option is displayed only when the running mode is set to Running in Sandbox.
	Terminal Command Running Mode > Custom Sandbox Settings	Configure the sandbox.json file to customize the file and network access scope of processes in the sandbox environment of the current project. For details, see Sandbox.

Type	Configuration Item	Description
	Terminal Command Running Mode > Network Access	The agent provides three network access policies for you to choose from. Configure access control rules based on actual service requirements and security levels. • Network-wide: Access to all internal and external network resources is allowed. • Local: Only the local network (LAN/intranet) can be accessed. Access to the Internet is not allowed. • Blocked: All network connections are blocked, and access to internal or external resources is prohibited. • Custom: Allows you to modify the JSON policy configuration file to customize the file and network access scope for processes within the project sandbox environment. NOTE This option is displayed only when the running mode is set to Running in Sandbox.
	Auto-approve > Edit	Allow the agent to call tools like edit, write, and deleteFile to edit files on your computer.
	Auto-approve > Browser Access	Allow the agent to access websites in the browser.
	Auto-approve > Web Crawler	Allow the agent to access and capture specified web page content.
Agent security policy	Dotfile Protection	When enabled, the agent is not allowed to modify configuration files starting with a dot, such as .env and .gitignore .
	Edit Scope Protection	When enabled, the agent can only add, modify, and delete files in the current project directory. It cannot edit files in other directories.
	Read Scope Protection	When enabled, the agent can only access and read files in the current project directory. It cannot access files in other directories.

Models

Enterprise members can add third-party LLMs to CodeArts Agent Space. If you have purchased the CodeArts Agent professional edition, you can also add LLMs on the console.

- For details about how to add a model in CodeArts Agent Space, see [Models](#).
- For details about how to add a model on the console, see [Configuring Custom Models](#).

MCP

CodeArts Agent Space connects to an MCP server using the Model Context Protocol (MCP). It leverages the server's extra tools and resources to enhance its capabilities.

For details about MCP configuration, see [MCP](#).

Skills and Rules

You can view and configure skills and rules for enterprises, teams, projects, and users. This ensures consistent yet adaptable workflows tailored to general standards or individual preferences.

- Enterprise skills and rules: Standardized skill libraries and business rules that are centrally built, managed, and distributed at the company or large organization level. The core objective is to eliminate execution deviations and ensure that different teams and individuals strictly adhere to unified operation standards, security protocols, and process guidelines in complex business scenarios. This enhances overall operational consistency and controllability.
- Team skills and rules: Skills and rules that are uniformly configured for teams or departments to standardize team operations and ensure consistent operations among all members. Only team administrators or department administrators can create, modify, and delete these skills and rules. Common members can only use them.
- Project skills and rules: Skills and rules that apply only to the current project. Bound to the project repository, they are distributed alongside it to address specific project needs.
- User skills and rules are classified into local and cloud ones.
 - Local ones: effective for all projects of the current user, formulated based on personal habits or preferences, and stored locally.
 - Cloud ones: formulated based on personal habits or preferences, available only to the members themselves, and stored in the cloud.
- Built-in skills and rules: basic skills preset in the system, globally available, and ready to use out of the box. They provide stable, continuous baseline functionality and can be used directly without any additional configuration.
- Skill marketplace: a hub of multiple ready-to-use skills designed to help rapidly boost development efficiency.

For details about skill configuration, see [Skills](#). For details about rule configuration, see [Rules](#).

Memory

Memory allows CodeArts Agent Space to adapt to how you work. The agent will record your habits and preferences, such as your preferred answer style and language.

For details about memory, see [Memory](#).

Repository Index

The repository index function is designed to enhance code retrieval efficiency. CodeArts Agent Space supports building both local user indexes and cloud shared indexes, allowing flexible adaptation based on your actual needs to achieve efficient information retrieval and management.

For details about the repository index function, see [Repository Index](#).

Lab Features

Language Server Protocol (LSP) is a standardized protocol for providing language services. By decoupling language services from the IDE, LSP enables an independent language server process to deliver core capabilities such as code analysis, error diagnosis, fast navigation, and code refactoring.

CodeArts Agent Space now features a built-in local Language Server Protocol (LSP) function. When enabled, the agent can leverage definitions and symbol data from language services for more precise code comprehension and modification.

For details about LSP, see [LSP](#).

IM Integration

CodeArts Agent Space provides IM access capabilities for multiple platforms, supporting the reuse of agent capabilities across these platforms alongside unified configuration management.

For details about IM integration, see [IM Integration](#).

Knowledge Base

CodeArts Agent Space enhances chat capabilities by utilizing knowledge bases as context. Developers can upload internal documents and technical resources to improve response accuracy.

For details about the knowledge base, see [Knowledge Bases](#).

About CodeArts Agent

On the settings and update management page of CodeArts Agent, you can view the current version information and configure automatic update policies. Multiple navigation links are provided, such as the product homepage, documentation, latest news, FAQs, and knowledge base, to help you obtain assistance or stay updated with product developments.

For details about CodeArts Agent, see [About CodeArts Agent](#).

10 IM Integration

10.1 Lark

CodeArts Agent Space provides IM access capabilities for multiple platforms, supporting the reuse of agent capabilities across these platforms alongside unified configuration management.

This section describes how to connect CodeArts Agent Space to Lark, allowing you to quickly integrate capabilities and enable collaborative workflows.

Constraints

Table 10-1 Constraints

Category	Description
Function	A Lark bot can connect to only one CodeArts Agent Space.

Procedure

Table 10-2 Connecting CodeArts Agent Space to Lark

No.	Operation Platform	Step	Description
1	Lark developer console	Step 1: Preparations	Before connecting CodeArts Agent Space to a Lark bot, perform the following operations: 1. Create a custom application. 2. Add a bot. 3. Add application identity permissions for the bot. 4. Add events and callbacks for the bot. 5. Release the application. 6. Obtain the application credential.
2	CodeArts Agent Space	Step 2: CodeArts Agent Space Configuration	In CodeArts Agent Space, register a Lark bot to receive and reply to messages.
3	Lark client CodeArts Agent Space	Step 3: Remote Control from Lark	Use the Lark bot to deliver tasks to CodeArts Agent Space.

Step 1: Preparations

Step 1 Create a custom application.

Log in to the [Lark open platform](#), click **Developer Console** in the upper right corner, and click **Create Custom App** on the **Custom Apps** tab.

Step 2 Add a bot.

Go to the **Add Features** page. On the **By Feature** tab, click **Add** under a bot to add the bot for the application.

Step 3 Add application permissions for the bot.

Go to the **Permissions & Scopes** page and click **Add permission scopes to app**. On the **Tenant token scopes** tab, add the following permissions for the bot:

- Obtain group information (**im:chat:readonly**)
- Obtain and send direct messages and group chat messages (**im:message**)
- Obtain and update group information (**im:chat**)
- Receive users' mentions (**im:message.group_at_msg:readonly**)
- Receive other bots' and users' mentions (**im:message.group_at_msg.include_bot:readonly**)

- Get direct messages sent to bot (**im:message.p2p_msg:readonly**)
- Send messages as an app (**im:message:send_as_bot**)
- Read and upload images or other files (**im:resource**)
- Get card information (**cardkit:card:read**)
- Create and update card (**cardkit:card:write**)

Step 4 Add events and callbacks for the bot.

1. Go to the **Events & Callbacks** page. In the **Event configuration** area, select **Receive events through persistent connection** under **Subscription mode** and click **Save**.
Subscription mode must be set to Receive events through persistent connection.
2. Click **Add Events**. On the **Tenant Token-Based Subscription** tab, add the following events:
 - Bot added to group (**im.chat.member.bot.added_v1**)
 - Bot removed from group (**im.chat.member.bot.deleted_v1**)
 - Message_received (**im.message.receive_v1**)
 - Message-read (**im.message.message_read_v1**)
3. Go to the **Events & Callbacks** page. In the **Callback configuration** area, select **Receive callbacks through persistent connection** under **Subscription mode** and click **Save**.
Subscription mode must be set to Receive callbacks through persistent connection.
4. Click **Add callback**, search for and add **Card callback communication** (**card.action.trigger**).

Step 5 Release the application.

Go to the **Credentials & Basic Info** page, click **Create Version** in the upper part of the page, and submit the application release request. Then, choose **Version Management & Release** under **App Versions** to check the progress. If the version status is **Released**, the application has been successfully released.

Step 6 Obtain the application credential.

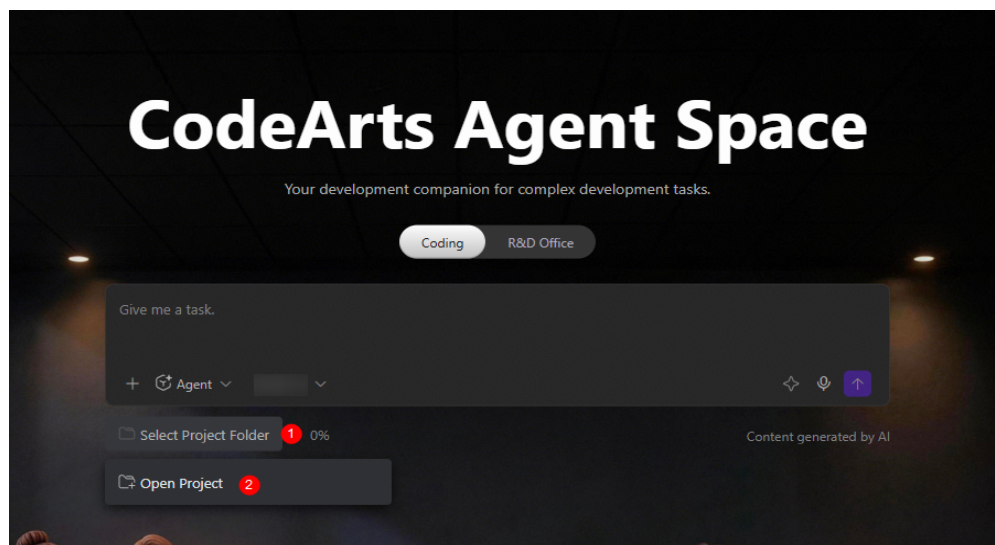
Go to the **Credential & Basic Info** page. In the **Credentials** area, obtain **App ID** and **App Secret**. This credential will be used when you connect to the Lark bot in CodeArts Agent Space.

----End

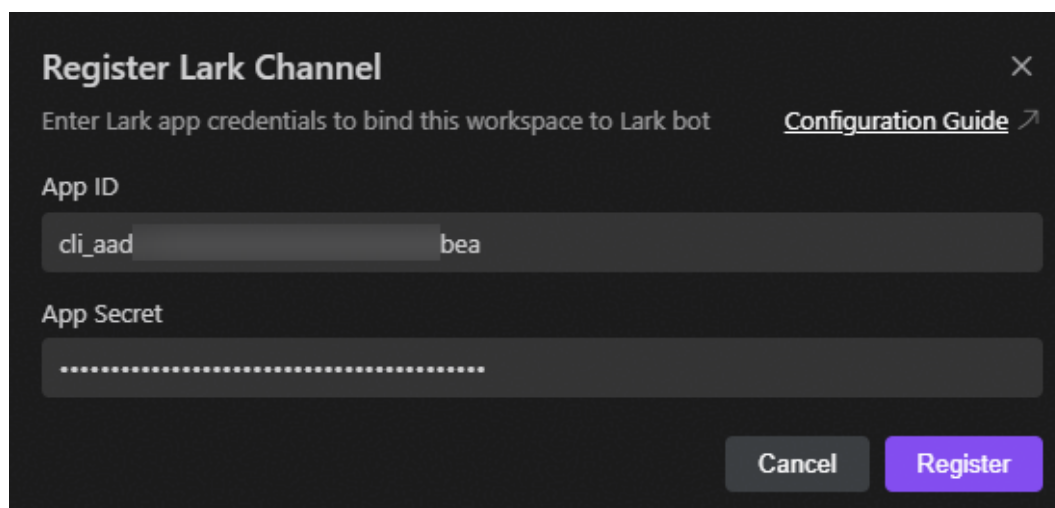
Step 2: CodeArts Agent Space Configuration

Step 1 Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).

Step 2 In CodeArts Agent Space, click **Select Project Folder**, choose **Open Project**, and open a folder.

Figure 10-1 Opening a folder

- Step 3** Click  in the upper right corner of CodeArts Agent Space to open the settings page.
- Step 4** Choose **IM Integration** and click **Configure** next to Lark. The **Register Lark Channel** page is displayed.
- Step 5** Enter the Lark application credentials (Client ID/Client Secret) obtained in [Step 6](#) and click **Register**.

Figure 10-2 Registering Lark

Wait until the registration is complete. When the status changes from **Connecting** to **Connected**, the Lark bot is successfully registered.

----End

Step 3: Remote Control from Lark

- Step 1** Log in to the Lark client and add the bot.

- One-to-one chat: In the search box of the Lark client, search for the bot created in 2. Click the search result to access it.
- Group chat: In the upper right corner of the Lark client, click ... and choose **Settings**. In the **Bots** area, click **Add Bot**. Search for the bot created in Step 2, click the search result, and click **Add** to add the bot to the group chat.

Step 2 Describe the task to the Lark bot and click **Send** to submit the task.

The bot automatically receives feedback from CodeArts Agent Space. You can use **slash commands** on the Lark client to trigger corresponding functions more effectively.

Step 3 Confirm the task requirements and key information based on the feedback and complete the task creation.

Step 4 Check the task execution result.

- **Lark bot:** The Lark bot receives a completion message from CodeArts Agent Space, indicating that the task execution is complete and providing the location of the deliverables.
- **CodeArts Agent Space:** IN **IM** > **Lark**, you can check the task execution process and result in real time. Click  in the upper right corner of the chat window to view the generated deliverables in the task artifact list.

----End

Slash Commands Supported by the Lark Client

Table 10-3 Supported slash commands

Command	Description
/stop	Stops the current session.
/help	Checks available commands.
/new	Initiates a new chat. When a new chat is created, CodeArts Agent Space automatically saves the original chat to the history list. After this, you will enter a new chat environment and cannot return to the previous chat.
/models	Checks and switches models. (Only CodeArts Agent Space built-in models)

10.2 Slack

CodeArts Agent Space provides IM access capabilities for multiple platforms, supporting the reuse of agent capabilities across these platforms alongside unified configuration management.

This section describes how to connect CodeArts Agent Space to Slack, allowing you to quickly integrate capabilities and enable collaborative workflows.

Constraints

Table 10-4 Constraints

Category	Description
Function	A Slack bot can connect to only one CodeArts Agent Space.

Procedure

Table 10-5 Connecting CodeArts Agent Space to Slack

No.	Operation Platform	Step	Description
1	Slack developer console	Step 1: Preparations	Before connecting CodeArts Agent Space to a Slack bot, perform the following operations: 1. Create a Slack application. 2. Obtain the application token. 3. Obtain the bot token. 4. Subscribe to events. 5. Enable messages. 6. Obtain the key.
2	CodeArts Agent Space	Step 2: CodeArts Agent Space Configuration	In CodeArts Agent Space, register a Slack bot to receive and reply to messages.
3	Slack client CodeArts Agent Space	Step 3: Remote Control from Slack	Use the Slack bot to deliver tasks to CodeArts Agent Space.

Step 1: Preparations

Step 1 Create a Slack application.

1. Sign in to the [Slack developer console](#) and click **Create New App**. The **Create an app** page is displayed.
2. Select **From scratch**, set **App Name**, select a workspace, and click **Create App**.
App Name is the bot name.

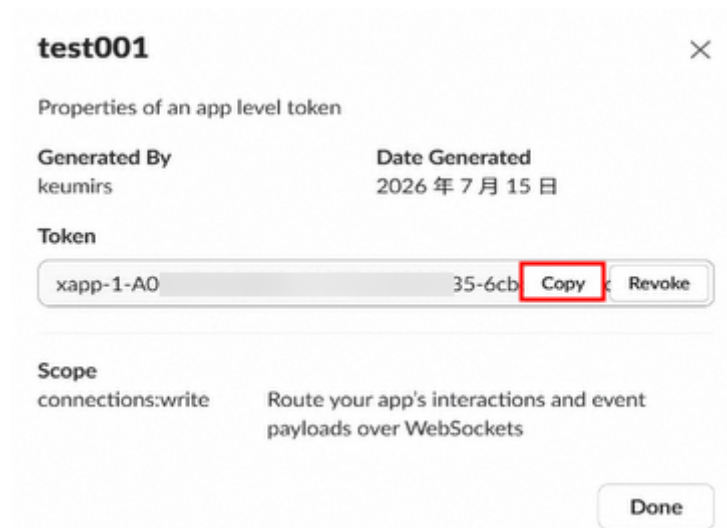
Step 2 Enable **Socket Mode** and obtain the application token.

1. Go to the **Socket Mode** page and toggle on **Enable Socket Mode** under **Connect using Socket Mode**.

2. Enter a token name and click **Generate**.

Copy the application token as prompted. **Keep the token secure. It can be copied only once.** This token will be used when you connect to the Slack bot in CodeArts Agent Space.

Figure 10-3 Copying a token

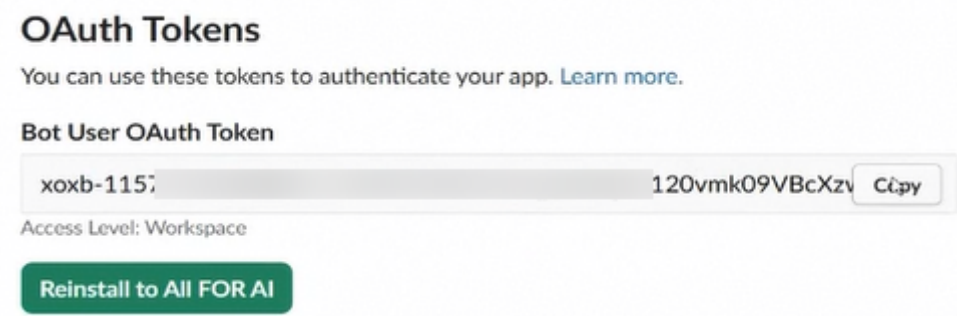


Step 3 Obtain the bot token.

1. Go to the **OAuth & Permissions** page, click **Add an OAuth Scope** in **Scopes > Bot Token Scopes**, and add the following permissions:
 - app_mentions:read
 - channels:history
 - chat:write
 - groups:history
 - im:history
 - im:read
 - im:write
 - mpim:history
 - users:read
2. In **OAuth Tokens**, click **Install to All FOR AI** and click **Allow** to obtain the bot token.

Keep the bot token secure. This token will be used when you connect to the Slack bot in CodeArts Agent Space.

Figure 10-4 Bot token



Step 4 Enable event subscription.

1. Go to the **Event Subscriptions** page and toggle on **Enable Events**.
2. After the function is enabled, click **Add Bot User Event** in **Subscriptions to bot events**.
3. Add the following events and click **Save Changes**.
 - message.im
 - app_mention

Step 5 Enable the function of sending messages to the bot.

Go to the **App Home** page, enable **Messages Tab**, and select

Allow users to send Slash commands and messages from the messages tab.

Step 6 Obtain the key.

Go to the **Basic Information** page and obtain the client secret from **App Credentials**. The client secret will be used when you connect to the Slack bot in CodeArts Agent Space.

Step 7 Customize slash commands.

Go to the **Slash Commands** page and click **Create New Command** to create slash commands in [Figure 10-5](#).

Figure 10-5 Custom slash commands

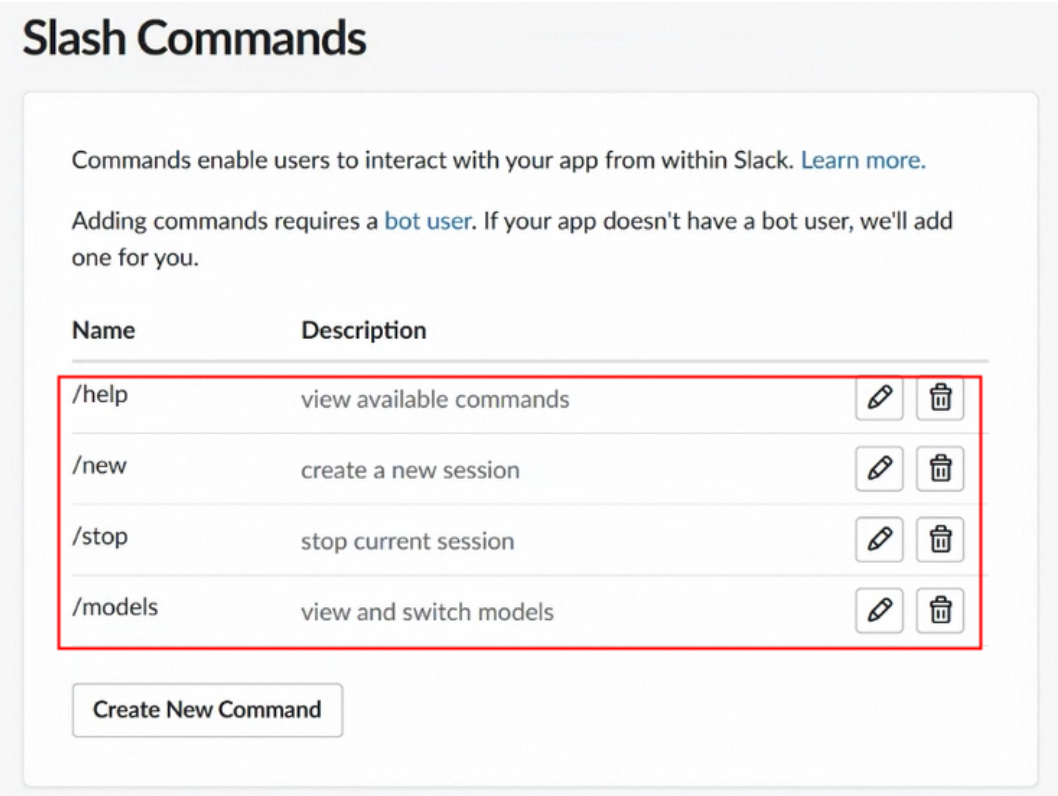


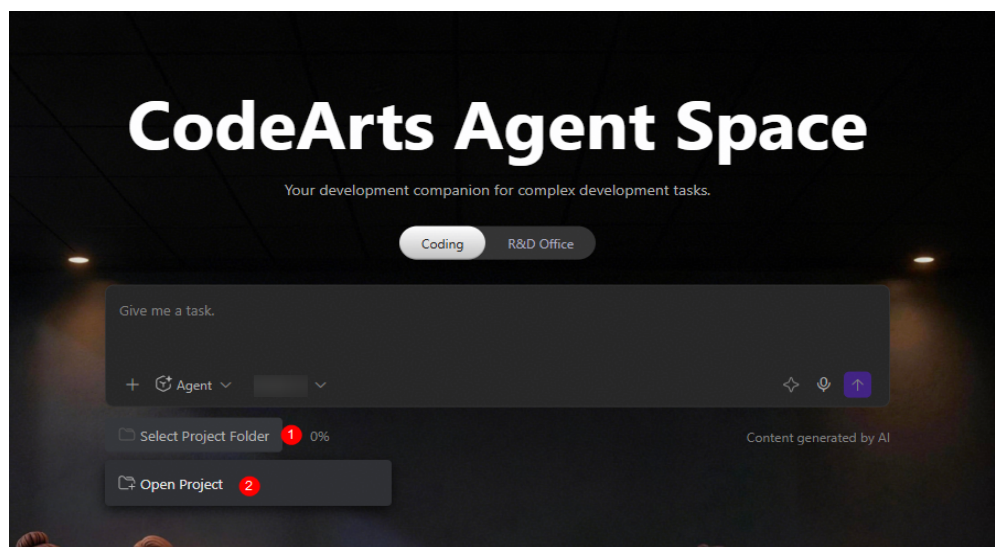
Table 10-6 Custom slash commands

Command	Description
/stop	Stops the current session.
/help	Checks available commands.
/new	Initiates a new chat. When a new chat is created, CodeArts Agent Space automatically saves the original chat to the history list. After this, you will enter a new chat environment and cannot return to the previous chat.
/models	Checks and switches models. (Only CodeArts Agent Space built-in models)

----End

Step 2: CodeArts Agent Space Configuration

- Step 1 Access CodeArts Agent Space by referring to [Enabling CodeArts Agent Space](#).
- Step 2 In CodeArts Agent Space, click **Select Project Folder**, choose **Open Project**, and open a folder.

Figure 10-6 Opening a folder

- Step 3** Click  in the upper right corner of CodeArts Agent Space to open the settings page.
- Step 4** Choose **IM Integration** and click **Configure** next to Slack. The **Register Slack Channel** page is displayed.
- Step 5** Enter the Slack bot credentials (Client ID/Client Secret) obtained in [Step 6](#) and click **Register**.

Set **Bot Token** to **Bot User OAuth Token** obtained in [Step 3](#), **App-Level Token** to the token obtained in [Step 2](#), and **Signing Secret** to **Client Secret** obtained in [Step 6](#).

Figure 10-7 Registering the Slack bot

Wait until the registration is complete. When the status changes from **Connecting** to **Connected**, the Slack bot is successfully registered.

----End

Step 3: Remote Control from Slack

Step 1 Log in to the Slack client. In **Apps & Agents**, click the bot added in **1**.

Step 2 Describe the task to the Slack bot and send it to submit the task.

The bot automatically receives feedback from CodeArts Agent Space. You can use slash commands customized in **Step 7** on the Slack client to trigger corresponding functions more effectively.

Step 3 Confirm the task requirements and key information based on the feedback and complete the task creation.

Step 4 Check the task execution result.

- **Slack bot:** The Slack bot receives a completion message from CodeArts Agent Space, indicating that the task execution is complete and providing the location of the deliverables.
- **CodeArts Agent Space:** IN **IM > Slack**, you can check the task execution process and result in real time. Click  in the upper right corner of the chat window to view the generated deliverables in the task artifact list.

----End