Tencent Cloud Agent Development Platform

Operation Guide

Product Documentation

Contents

Operation Guide
- Application Management
- Application Center
  - Application Configuration
  - Knowledge Base Management
    - File
    - Q&A
    - Label Management
    - Set Knowledge Base Search Scope
    - File Comparison Task
  - Release Management
- System Management
  - Call Statistics
- Application API Documentation
  - Dialogue API Overview
  - Dialog API Documentation (WebSocket)
  - Dialog API Documentation (HTTP SSE)
  - Offline Document Upload
  - Tencent Cloud Agent Development Platform Operation COS Guide

Operation Guide
Application Management

Last updated：2025-05-29 14:24:28

You can create applications in the Application Management space. It supports viewing, calling, and deleting existing applications.

Create Application

Tencent Cloud Agent Development Platform provides zero-code creation and configuration capabilities for preset application paradigms, enabling efficient online configuration of corporate proprietary applications and reducing the development threshold for large model applications.
1. Enter Tencent Cloud Agent Development Platform, click on the left menu Application Management.
Note:
Tencent Cloud Agent Development Platform requires activation before usage.

2. Click Create Application. Enter the Create Application interface. Fill in the Application Name, upload the Application Icon, and then click Create.
app icon: Support JPG and PNG formats. The image size should be less than 500 KB.
app name: Supports customizing the application name. The maximum character count is 20. Does not support same name.

3. After successfully creating the application, click View on the right side of the application to enter the application configuration page. You can further edit the application's related configuration items, perform a dialogue test, and then publish.

Application Editing

The application management list supports managing the application, searching for the application and the last modified by.
Application operations support viewing, calling, and deletion.

The field descriptions in the application list are as follows:

Field	Field Description

Thinking Model	When not online, display the thinking model of the testing environment. After going live, only display the thinking model of the release environment.
Generative Model	When not online, display the generative model of the testing environment. After going live, only display the generative model of the release environment.
Status	Release environment service status

Application Center
Application Configuration

Last updated：2025-05-29 17:01:05

Create Application. After entering the application configuration page, the **configuration information** on the left can be used to configure parameters. Different parameters will affect the application's effect, and real-time debugging can be performed in the **debugging area** on the right.

Basic settings

After creating the application, click the app icon to set the app icon and application name in the **Edit Application** pop-up window.

The app icon and name will be displayed in the user end interface window after completion and publishing.

Model Configuration

Support selecting thinking models and generative models in model configuration. Thinking models are used for intent recognition and task planning. Generative models are used for reading comprehension and generating reply summaries. Support selecting platform presets from Tencent Cloud Intelligence development platform.

Large Model Service: New users of Tencent Cloud Intelligence development platform will automatically receive a certain amount of free quota, and can freely apply debugging by selecting different types of models; based on test results, you can further purchase and use.

Context Rewriting: After enabling the switch, it can combine context content to identify reference objects or omit words, rewrite the current question, and generate a coherent answer.
Context Memory Rounds: Set the number of context dialogue history rounds provided as a prompt to the large model. The higher the number of rounds, the higher the relevance of multi-round dialogue, but the token consumption will also increase.
Advanced Settings:
Temperature: Increasing the temperature makes the model's output more diverse and random, suitable for scenarios with high creative requirements, such as poetry creation. Conversely, decreasing the temperature makes the output more adherent to instructions, suitable for scenarios with high deterministic requirements, such as code generation.
Top P: Top P is a cumulative probability. When generating output, the model selects words starting from those with the highest probability until the cumulative probability of the selected words reaches the Top P value. It can restrict the model to select only these high-probability words, thereby controlling the diversity of output content. The larger the value, the stronger the diversity of generated content.

Role Instructions

After a user asks a question, the application will provide answers based on the task role defined in "**Role Instructions**". You can refer to the given input to limit the language and tone of the model's responses. Currently, the Tencent Cloud Intelligence development platform supports Chinese and English QA output.
Template: A pre-defined role instruction format template. It is advisable to fill in according to the template for better effect when following the instructions.

Welcome Message

After filling in the welcome message, the opening remark displayed on the user-end interface for communication with users will be the welcome message.

Knowledge Base

Knowledge base configuration supports setting the switch for documents/Q&A knowledge base separately.
Document: The large model will answer questions based on the document library you build. You can choose to directly upload files or upload web pages. The large model will parse and learn the documents you upload. Relevant content of the document can be viewed in File.
Q&A: The large model will answer questions based on the Q&A database you build. You can choose to directly upload files for batch importing Q&A, manually enter Q&A content, or automatically generate Q&A from files in the document library. Relevant content of the Q&A can be viewed in Q&A.
Advanced Options:
Retrieval strategy: Hybrid retrieval - Executes keyword retrieval and vector retrieval simultaneously. Recommended for scenarios requiring string and semantic association, offering superior overall performance. Semantic retrieval - Recommended when there is minimal overlap between the query and text segments, and semantic matching is required.
Number of document recalls: Retrieve the N document fragments with the highest matching degree returned by the search and provide them as input to the large model for reading comprehension.
Document retrieval matching degree: Returns found text fragments to the large model as reference for replies based on the set matching degree. Lower values mean more fragments are recalled, but may affect accuracy. Content below the matching degree will not be recalled.
Reply to Q&A database answers: If the similarity between the current question and the questions in the Q&A database reaches a certain threshold, selecting direct reply will directly use the entered answer for response; selecting polished reply will send the question to the large model for summarized response.
Number of QA recalls: Retrieve the N QAs with the highest matching degree returned by the search and provide them as input to the large model for reading comprehension.
QA retrieval matching degree: Returns found Q&A content to the large model as reference for replies based on the set matching degree. Lower values mean more fragments are recalled, but may affect accuracy. Content below the matching degree will not be recalled.

Output Configuration

Support selecting answers to be output in streaming or non-streaming forms. Streaming means word-by-word output, while non-streaming means the answer is output all at once after full generation.
For unknown questions outside the knowledge source, reply settings can be performed. Reply using the configured unknown question reply messages or use the large model to intelligently reply to all questions.

Advanced Configuration

Set synonyms: Proprietary terms in business scenarios can be imported. For synonyms in queries, they will be uniformly replaced with unified names from the knowledge base before retrieval to improve retrieval accuracy.

Testing Pane

Reference Source

Application configuration page by default displays reference sources, supports viewing answer reference fragments after clicking.

Simultaneously support tracing the source of reference sources and viewing the original document on the dialogue test page. Support navigating to the corresponding fragment's page in a new tab (supports pdf\\doc\\ppt\\pptx); csv\\xlsx\\xls supports jumping to specific sheets, and images support opening the original image.
If you need to display the reference source in the official environment (API integration), you need to turn on the **External Reference Link** switch in the document settings. For detailed operations, check the File.

Debugging Information

Click to expand the running process below the dialogue pane to show the operating status of the current session.

Regenerating

Click **Regenerate**, and the model will resend this round's question to generate answer again.

Correcting

Click **Correct** to support entering the current QA into the Q&A database. Answers can be corrected manually to fix the current issue's reply. Next time a similar type of question appears, the corrected answer from the Q&A database will be prioritized for matching.

Knowledge Base Management
File

Last updated：2025-05-29 14:32:48

File-based knowledge refers to knowledge in the form of documents, such as files in PDF, docx, txt formats and web pages. By leveraging the capabilities of Tencent Cloud Agent Development Platform, after importing knowledge into the system, the set application will form a business knowledge base based on relevant knowledge, which can directly answer user questions according to the content of the knowledge base.

Note:

The knowledge base has limited capacity. You need to purchase a knowledge base capacity expansion package for use. After the knowledge base capacity expansion package expires, files/Q&As exceeding the character capacity will change to the [Over-quota Invalidation] status. Manual restoration is required after expansion. The invalidated knowledge cannot be retrieved on the dialogue end and will be automatically deleted one month after invalidation. It is advisable to renew or purchase additional packages before expiration.

You can enter **Application - Knowledge Management - File** to manage file-based knowledge. It supports importing files, downloading files, deleting files, and files classification.

Importing Files

Document import supports webpage content import and local file import. Tencent Cloud Agent Development Platform supports importing files from webpage content and local files. Tencent Cloud Agent Development Platform will learn from the imported files and provide responses based on the files.

File Import Settings

File Label: Used to tag files. You can configure the matching relationship between API parameters and tags in Knowledge Base Search Scope Settings. Pass API parameters through the custom_variables field of the application dialogue endpoint API. When users ask questions with different API parameter values, retrieve file content that matches the tag values. For details, see Knowledge Base Search Scope Settings.

Expiration Time: Set the effective time for file knowledge. You can set it to permanent validity or customize the expiration time. If a custom expiration time is set, the file knowledge will automatically expire after the expiration time.

: Once enabled, it will display the source at the end of the answer and support online viewing; you can choose to display the file link referenced by the large model or customize the reference link (such as the homepage).

File Classification: Establishing file classifications in the knowledge base helps conveniently manage knowledge Q&A in different categories. It supports up to 10 levels of classification. Supports renaming, deleting, and searching for

classifications. When you hover over any classification, ... will be displayed on the right. Click **...** to display a dropdown. Click **Add** subclass, enter the name and press Enter to create a new subclass under the current classification.

Import Steps for Webpage Content

Note:

Website link limit:

The web pages to be crawled require no login authorization verification, that is, they are accessible without verifying the current user's identity and granting the user system access.

Not currently supported to crawl website content of asynchronous loading type.

Please make sure to use this web page parsing tool within the allowed range of laws and regulations, comply with the target platform management specifications, and guarantee the legitimate rights and interests of the right holder. You shall bear independent liability for this. Tencent Cloud Agent Development Platform, as a tool provider, does not assume any responsibility for your parsing or downloading actions.

1. After entering the Knowledge Base QA application details page, select **Knowledge Management > File** to enter the file management interface.

2. Click **Import**. A file import drop-down box will be displayed. Select **Import from Webpage** and switch to the webpage import window.

3. Enter the website address, click **Retrieve Webpage Content**. Tencent Cloud Agent Development Platform will automatically crawl images, text, etc. from the webpage and display the preview. After configuring the file, click **Save as Document** to complete the webpage import.

Local Files Import Steps

1. After entering the Knowledge Base QA application details page, select **Knowledge Management > File** to enter the file management interface.

2. Click **Import**. A file import drop-down box will be displayed. Select **Import from Local Files**. A file import window will pop up.

3. After completing the relevant information input in the file import window, click the **Import Files** button to complete the file import.

Note:

Conditions for importing local files:

Support pdf, doc, docx, ppt, pptx formats. Size limit: 200 MB.

Support xlsx, xls, md, txt, csv formats. Size limit: 20 MB.

Support importing images with text, including png, jpg, jpeg formats. Size limit: 50 MB. Aspect ratio no more than 1:7.

Table files (in xlsx, xls, or csv format) support up to 10,000 rows and 100 columns of data. It is advisable to store only one table in a sheet. The occurrence of completely empty rows in the table will affect the Q&A effect.

Support batch import of files.

File Operations

View Document: After entering the Knowledge Base QA application details page, select the **Knowledge Management > File** Tab and click the file name to view the file content.
File status: Refers to the processing status of the file by the system and the custom active status after the file is uploaded.
Status Description
Parsing in progress: Performing file parsing work. No setting adjustment is supported for files in this status.
Parsing failure: File parsing fails. A pop-up will prompt and details are viewable.
Under review: Performing file review work. No setting adjustment is supported for files in this status.
Review failure: The file review fails. The reason may be that the content of the file does not conform to the specified standard or requirement.
Learning: Performing file learning work. No setting adjustment is supported for file in this status.
Learn failure: The knowledge base QA application fails to learn the file content and is unable to conduct Q&A in dialogue tests and the official environment based on the file.
- Awaiting release: The file has been deployed and can be tested in the dialogue test. It is to be published to the formal environment to take effect. For file in this status, QA can be generated.
Releasing: The file is being published from the testing environment to the formal environment. No setting adjustment is supported for file in this status.
Released: The file has been published to the formal environment, and the application can answer user questions based on the file.
Expired: The file has expired and is invalid. QA cannot be conducted based on the file in the dialogue test and the official environment.
Manual Appeal: When the file review fails and is submitted for manual review, the status during the manual review process is Manual Appeal.
Manual Appeal Failed: The manual review is not approved, the file status is Manual Appeal Failed, and it is required to modify the file offline and then import it again.
Excess Capacity Invalidation: When the knowledge base capacity expires and the used knowledge base capacity exceeds the available knowledge base capacity, the files exceeding the capacity constraint are processed as in an excess capacity invalidation state.
Recovery from Excess Capacity Invalidation: The process of restoring knowledge in an excess capacity invalidation state to its status before invalidation. Documents in excess capacity invalidation need to be manually recovered.
Document Search: Support searching for files by file name/tag name.
Document Download: Download imported files from Tencent Cloud Agent Development Platform to your local system.
Document Deletion: Delete files in the knowledge base.
Note:

Deleted files will not result in the deletion of the corresponding Q&A in the Q&A database.

Q&A

Last updated：2025-05-29 16:50:13

QA knowledge exists in pairs in the form of "question-answer".
By leveraging the capabilities of Tencent Cloud Agent Development Platform, after importing knowledge into the system, the configured application will form a business knowledge base based on relevant knowledge, which can directly answer user questions according to the content of the knowledge base.

Note:

The knowledge base has limited capacity. A knowledge base capacity expansion package is required for use. After the knowledge base capacity expansion package expires, files/Q&As exceeding the character capacity will change to [Over-quota Invalidation] status. Manual restoration is required after expansion. The invalidated knowledge cannot be retrieved from the dialogue end and will be automatically deleted one month after invalidation. It is advisable to renew or purchase additional capacity before expiration.
The knowledge base QA application supports creating Q&As, moving Q&As, exporting Q&As, verifying Q&As, and categorizing Q&As. After entering Q&As, when a customer inputs a question, the system will retrieve and match the customer's question with those in the Q&A database. If the similarity reaches a certain threshold, it will output the corresponding answer.

Creating Q&As

Tencent Cloud Agent Development Platform supports multiple methods for creating Q&As, including batch importing Q&As, manually entering Q&As, and generating Q&As from documents. The platform prioritizes adopting Q&A content for replies, enhancing the accuracy of key question responses.

Batch Importing Q&A Steps

1. After entering the Knowledge Base QA application details page, select the **Knowledge Management > QA** Tab to enter the QA management page.
2. Click **Create**, display the Create QA drop-down box, select **Batch Import Q&A**, and a Batch Import Q&A window will pop up.
3. Download the template file, complete the QA based on the instructions, and then batch import the completed QA into Tencent Cloud Agent Development Platform via file upload.

Note:

Requirements for batch importing Q&A files:

1. File size is within 5 MB.
2. Note that the length of each question and answer cannot exceed 2000 characters.
3. Single import: The maximum number of entries must not exceed 10,000.
4. Category title cannot exceed 10 characters.

Manually Enter QA Steps

1. After entering the Knowledge Base QA application details page, select the **Knowledge Management > QA** Tab to enter the QA management page.
2. Click **Create**, display the Create QA drop-down box, select **Manual Entry QA**, and a QA Entry window will pop up.
3. After completing the Q&A content input in the Enter QA window, for other settings, please see Q&A. Click **Enter** to complete the Q&A entry.

Steps to Generate Q&A From Documents

Note:
Generate QA from files is only supported for files that have been imported into the system. Before using this feature, please import files in **Knowledge Management > File**. For related steps, see File.
The time required to generate QA varies depending on the document size. The more characters there are, the longer it takes to generate QA. After generation, a pinned message will be displayed as a reminder.
1. After entering the Knowledge Base QA application details page, select the **Knowledge Management > QA** Tab to enter the QA management page.
2. Click **Create** to display the Create QA drop-down list, select **Generate QA from Document**, and a document selection page will pop up. After selecting a document, click **Generate**, and the backend will generate QA based on the selected document. You can click **QA Generation Task** to check the task status.

3. After QA generation is completed, a notification will be sent via pinned messages and Message Center. The generated QAs can be viewed in **Pending Verification QAs**.

Note:
Supported file formats for document-to-QA: pdf, doc, docx, ppt, pptx, md, txt, jpg, png, jpeg; Tables are not supported.

4. Click **Pending Verification QAs** to manually verify the generated QA results. Click **Verify** to modify the generated questions and answers.

5. After verification, click **Adopt** to incorporate the QA into the Q&A Knowledge Base; unadopted issues will not take effect and can be viewed and modified later on the Unadopted page.

Note:

Conflicting QAs: Q&A pairs automatically generated by large models may have conflicts, meaning the same question has two different answers. To ensure answer accuracy, please merge conflicting QAs. The system will notify you of conflicts via pinned messages. You can click **Process** to retain the answers of conflicting QAs.

Performing Q&A Operations

Question and Answer Categories: Creating question and answer categories in the knowledge base helps with convenient management of different categories of Q&A; supports up to 10 levels of categorization. Supports renaming, deleting, and searching categories. When hovering over any category, ... will be displayed on the right. Click **...** to display a dropdown, click **Add** to create a subcategory, enter the name and press Enter to create a new subcategory under the current category.

QA Settings

: Click **Edit** for a QA or click a question to edit the QA.

The following are the descriptions of the setting items:

Similar Questions: When multiple questions share the same answer, you can add multiple similar questions one by one in the question editing box. When any similar question is hit, the current answer will be provided. Each similar question supports up to 500 characters, and a maximum of 100 similar questions can be entered for the same question. Similar questions can be generated with one click using a large model.

Note:

Generating similar questions with one click will consume the user's token resources.

Q&A Tag: Used to tag Q&A. You can configure the matching relationship between API parameters and tags in Set Knowledge Base Search Scope. Pass API parameters through the custom_variables field of the Application API. When users ask questions with different API parameter values, retrieve document content that matches the tag values. For details, see Set Knowledge Base Search Scope.
Q&A Source: Default setting, does not support modification, auto-matches based on the Q&A import method.
Expiration Time: The effective time setting for Q&A knowledge. It can be set to permanent validity or a custom expiration time. If a custom expiration time is set, the Q&A knowledge will automatically expire after the expiration time. For Q&A generated from documents, the expiration time remains consistent with the source document and cannot be changed.
Associated Document: Allows customization of the source document for Q&A pairs. Supports selection from the document library. If the associated document has reference link display enabled, the document source will be shown at the end when the Q&A is called, and the document can be viewed online.

Note:
For Q&A generated from documents, the Q&A tags and expiration time are consistent with the document and cannot be modified.
Move Q&A categorization: After selecting a Q&A, click **Move to Category** to move the selected Q&A categorization to another category.
Batch export Q&A: After selecting Q&A, click **Batch Export** to generate Excel files for the selected Q&A, which can be downloaded to local.
Filter Q&A: You can filter Q&A by source, associated document, or status.
Search Q&A: You can search in the question field by entering keywords.
Delete Q&A: After selecting Q&A, click **Delete** to remove the selected Q&A from the knowledge base.

Status of Q&A: Refers to the state after uploading and the custom active state published in the system.
Status Description:
Under review: Performing Q&A review work. Editing is not supported for Q&A in this status.
Review failed: The Q&A review has failed. The reason may be that the question, answer, similar question, or problem description of the Q&A does not conform to the specified standard or requirement.
Learning: Performing Q&A learning work. No setting adjustment is supported for questions in this status.
Learn failure: The Q&A content learning has failed. Unable to conduct Q&A in dialogue tests and the official environment based on the Q&A.

Awaiting release: The Q&A has been deployed and can be tested in the dialogue test. It is to be published to the formal environment to take effect.
Releasing: The Q&A is being published from the testing environment to the formal environment. Editing is not supported for Q&A in this status.
Released: The Q&A has been published to the formal environment, and the application can answer user questions based on the Q&A.
Expired: The Q&A has expired and is invalid. QA cannot be conducted based on the Q&A in the dialogue test and the formal environment.
Manual Appeal: When the QA review fails and is submitted for manual review, the status during the manual review process is Manual Appeal.
Manual Appeal Failed: The manual review is not approved, the Q&A status is Manual Appeal Failed, and modifications are needed to the Q&A before saving again.
Excess Capacity Invalidation: When the knowledge base capacity expires and the used knowledge base capacity exceeds the available knowledge base capacity, the Q&A exceeding the capacity constraint is processed as in an excess capacity invalidation state.
Excess Capacity Recovery: The process of restoring knowledge in an excess capacity invalidation state to its state before invalidation. Q&A in excess capacity invalidation requires manual recovery.

Label Management

Last updated：2025-05-29 14:38:18

Label Management is a feature for maintaining label names and label values, which can be used to label files/QA.

Explanation of Related Terms

Term	Explanation
Label name	Customer business fields, for example, region, product type.
Label value - Standard term	Business field value. For example, East China, South China.
Label value - Synonym	Words with the same meaning and similar to the standard term of the current label value. For example, synonyms for East China can be set as Shandong, Jiangsu, Anhui, Zhejiang, Fujian, Jiangxi and Shanghai.

Label Management Portal

In Knowledge Management > File Settings/QA Entry/QA Editing > File Label/QA Label, at the bottom of the list where you can open to select a label name, it supports clicking to enter Label Management.

Add Label

Label management supports two methods: creating and importing.

Create a knowledge label.

click **Create** to open the Create Label pane and configure the label name and label value.

click **Save**. The new label is created successfully.

Import knowledge label.

click **Import** to pop up the Import label pane.

Download template file, fill in the label content as instructed, and after completion, upload the file. Click **Document import** to batch import label into label management.

Edit/Delete Labels

Click Edit to enter the label editing page, where you can modify the label name and label value. If the modified label is used in a file/QA, it will take effect in the release environment after being released.

Support deleting label one by one or in batches. When a label is used in a file/QA, its deletion is not supported.

Set Knowledge Base Search Scope

Last updated：2025-05-29 14:38:31

The knowledge base retrieval scope settings is used to configure the mapping relationship between API parameters and label names, to implement scenarios where users with different identities retrieve knowledge from different scopes when asking questions.

1. API parameter: Refers to the parameters passed in through the custom_variables field of the application dialogue endpoint API. These parameters can be maintained in API parameter management. For details, see Dialogue Endpoint API Documentation (WebSocket) and Dialogue Endpoint API Documentation (HTTP SSE).
2. Label name: Refers to the label name maintained in label management, used for tagging files and QAs.

Feature Description

Set the Knowledge Base Search Scope

Configure the mapping relationship between API parameters and label names. After configuration, the parameter content passed through the custom_variables field of the application dialogue endpoint API will match the knowledge of the corresponding label value. For details, see Overall Overview of Dialogue API.

How to Choose "AND" and "OR"

Case background:

1. Pass in {"UserID":"Internal","Department":"R&D"} in custom_variables.
2. Configure the mapping relationship in the knowledge search scope settings.

Fill in the API parameter with the label name "UserID" mapped to "User".

Fill in the API parameter with the label name "Department" mapped to "Department".

The label values of user identity include internal employee and external user. The label values of department include Product Department, R & D Department, and Test Department.

Set the selected API parameter to "AND".

When importing multiple parameters that map to multiple labels, it will retrieve **knowledge that contains both multiple labels** as well as untagged knowledge.

Take the above case as an example. The final result: Retrieve knowledge with "User" being "Internal" and "Department" being "R&D", as well as untagged knowledge.

Set the selected API parameter to "OR".

When importing multiple parameters that map to multiple labels, it will retrieve **knowledge containing any label** as well as untagged knowledge.
Take the above case as an example. The final result: Retrieve knowledge with "User" being "Internal", or knowledge with "Department" being "R&D", or untagged knowledge.

Configure the Mapping Relationship between API Parameters and Tag Names

API parameter name
Parameters passed through the custom_variables field of the application dialogue endpoint API can be managed in API parameter management. They can be referenced in knowledge base retrieval scope settings and workflow nodes. For details, see Dialogue Endpoint API Documentation (WebSocket) and Dialogue Endpoint API Documentation (HTTP SSE).

Note:
The API parameter name must be consistent with the parameters passed in custom_variables, and the format requires it to begin with an English letter and support English letters and underscores "_".

Label name
Maintain label names and label values in label management. Support tagging for documents/QAs. Support selecting labels in label management in Knowledge Base Retrieval Scope Settings, configuring the mapping relationship between API parameters and label names. After a user asks a question with parameters passed through custom_variables, retrieve knowledge of the corresponding label and answer.

Note:
1. The parameters passed through custom_variables to retrieve the knowledge of the corresponding label must configure the mapping relationship between API parameters and label names in the knowledge base retrieval scope settings.
2. Before configuring the mapping relationship, it is advisable to first maintain labels and tag documents/QAs in label management.

Usage Scenarios and Operation Steps

Scenario Overview

When in business scenarios, it is expected that users with different identities are restricted to retrieving a certain range of knowledge for answers. For example, employees in department A can only consult knowledge under department A.

If they ask about knowledge from other departments, a refusal response needs to be given.
Below is an example of the following scenario:
Scenario case:
User identity (distinguish between external users and internal employees). Internal employees need to be distinguished by different departments (Product Department, R & D Department, Test Department). External users do not need to be distinguished by other identities. To achieve knowledge isolation between different identities and departments, when a user asks questions, only knowledge corresponding to the identity/department can be retrieved for response.
UserID
Department

Operation Step Description

Step 1: Create a Label in Label Management

Enter the label management feature from the document/QA label. Create two label names: "User" and "Department". The standard words for user identity include **external user** and **internal employee**; the departments include **Product Department**, **R&D Department**, and **Testing Department**.

Step 2: Tagging Documents/QAs

1. Documents visible to external users. Select "Label Name = User" and "Label Value = External" in the file label; no need to set the department label.

2. **Documents visible to internal employees**
"Label Name = User", "Tag Value = Internal".
"Label Name = Department", "Tag Value = R&D". Select the label value based on the actual visible scope of employees.

Step 3: Set the Knowledge Base Search Scope

In **Knowledge Management > Knowledge LIbrary Settings > Knowledge Base Retrieval Scope Settings**, set the API parameter to "AND", with API parameter name "UserID", mapped label name "User"; API parameter name

"Department", mapped label name "Department".

Step 4: Apply Dialogue API Input Parameters

In the application dialogue terminal's custom_variables field, input parameters and parameter values.

External User

```
"custom_variables":{
  "UserID": "External"
}
```

Internal User

```
"custom_variables":{
  "UserID": "Internal"
  "Department": "R&D"
}
```

Complete the above configuration and parameter passing steps, and in the dialogue, it is achievable to restrict users with different identities from asking questions and retrieving a certain range of knowledge for answers.

File Comparison Task

Last updated：2025-05-29 14:38:49

The file comparison task is used to compare the differences between two files, such as file names, file contents, and Q&A pairs generated from file. It assists customers in handling duplicate or similar content in the knowledge base and guarantees the effect of knowledge Q&A.

Feature Description

How to Trigger a File Comparison Task

The ways to trigger a file comparison task include **automatic trigger comparison** and **manual trigger comparison**.

1. Automatic trigger comparison: A file comparison task automatically generated by the system when the following conditions are met.

When the file names are the same: When the name and file type of the file uploaded by the local customer are the same as those in the file list, but the file content is different, and the import status of the newly imported file (i.e., the new file) is the pending release status, the file comparison will be automatically triggered.

Same website address source: When a customer imports a file with the same website address, and the import status of the new file is the pending release status, the file comparison will be automatically triggered.

Note:

Note: To view the file comparison task entry: After triggering file comparison, you can click the "View Comparison" notification in the banner display above the file list or the "File Comparison Task" button in the top right corner of the file list to enter the file comparison task list.

2. Manual comparison: Support customers to add file comparison tasks in the file comparison task list.

Select 2 files from the file list (only supports selecting files in the pending release and published status). Once saved, 1 file comparison task will be generated.

How to Handle File Comparison Tasks

1. The status of a file comparison task includes pending, in processing, invalid, and completed.

Status	Status Description
Pending processing	For comparison tasks that support operations, after selecting file operations and QA operations and clicking Start Processing, it will enter the processing status (if no Q&A pairs have been generated for the file, there is no need to select QA operations).

Processing	A file comparison task belongs to an asynchronous task. It is in the processing status during execution. The processing status is a temporary state.
Expired	After file deletion, the comparison task will become invalid.
Completed	After the comparison task is processed and completed, the status is completed, and you can view the comparison task result.

2. Start processing the file comparison task
For pending status file comparison tasks, after selecting "File Operation" and "QA Operation", it supports one-by-one or batch processing tasks.
New files: For recently imported files, the list will display the import time and the situation of Q&A pair generation.
Old files: Files imported earlier, the list will display the import time and the situation of Q&A pair generation.
Contrast reasons: Include identical names, identical website addresses, and manual addition, distinguished by the trigger source.
View comparison details between two files. The different parts are distinguished by color.

File operations: Support customization to choose to delete old and new files, rename old and new files, etc.
QA operations: If the file has generated Q&A pairs, support generating Q&A based on file difference fragments, and associate the Q&A pairs of deleted files with reserved files.

Use Cases

Scenario 1: File update scenario
Upload a file with the same name will automatically trigger a comparison task. The operator can delete the old file and its Q&A pairs through the comparison task.
Scenario 2: Generate Q&A pairs from file difference fragments
Upload two files with similar content. The operator can manually create a comparison task and select the QA operation of "Generate Q&A from the difference fragments of the new file".

Release Management

Last updated：2025-05-29 14:39:06

Publishing and going live is the process of releasing an application from the testing environment to the formal environment. After the configuration and knowledge base of the current application have passed the effectiveness verification through testing, the relevant content can be released from the testing environment to the formal environment. Once published, you can use the Experience Link on the user end and conduct Q&A with the knowledge base through API management and the QA application based on the released knowledge and configuration.

Awaiting Release

1. Click the **Release Management > To Be Released** Tab. The interface will display all the content awaiting release for the application, including documents, Q&A, rejected questions, and configurations.

2. Click Publish. A pop-up confirmation window will appear, displaying the actual quantity of the content awaiting release.

3. Click Publish in the release window.
The backend will enter the release process.
After publishing is completed, you will be notified via the message center.
You can also view the release details through the Release History Tab.

Release History

Click the **Release History** Tab to view historical release details and released status.

Call Information

After successful release, you will obtain a share link for experience and support the creation of API calling keys. You can share this link with users or related testing operators, enabling them to immediately experience your large model application demo on the web experience page; or experience it on mobile phones via **share QR code**; at the same time, you can also directly call through the application API calling interface in the form of API.

System Management
Call Statistics

Last updated：2025-05-29 14:39:25

Call statistics provide the total number of resource consumption and call details, including usage statistics, concurrency statistics, and knowledge base capacity statistics.

Usage Statistics

Usage statistics provides reports and details on token consumption of models in application services.

Statistical Report

Reports distinguish between models/applications and can filter models and specific applications.
Model/Application: Viewable API call count for billing, number of tokens consumed, with support for time-based export of statistical details.

Single Call Detail

Support viewing consumption details of each call based on invocation type.

Concurrency Statistics

1. Support viewing the available concurrency of models, the peak concurrent number of successful calls, and the number of calls exceeding the available concurrency. The available concurrency is greater than or equal to the peak concurrency number of successful calls.
Available concurrency number: The maximum usable concurrency number of the current model.
Peak concurrent number of successful calls: The maximum number of concurrent successful calls to the current model within the currently specified time range of the filter, which does not exceed the available concurrency number.
Number of calls exceeding available concurrency: When the number of calls exceeds the available concurrency, queuing or call failures may occur, and the number of such calls will be recorded.
2. Support viewing details of calls exceeding available concurrency, view the time of each call and query. Based on the details of calls exceeding available concurrency, assess whether it is necessary to purchase additional concurrency.

Knowledge Library Capacity Statistics

Provide the total number of available characters and the total number of expired characters in the knowledge base. You can view the usage and proportion of each application's knowledge base in the Knowledge Library Capacity Statistics.

Total number of available characters in the knowledge base: The total number of characters in the knowledge base, including trial resources and purchased characters.

Total number of expired characters: The total number of characters in the knowledge base that are processed as in an excess capacity invalidation state. The number of expired characters requires purchasing a capacity package as soon as possible.

Application API Documentation
Dialogue API Overview

Last updated：2025-05-30 15:35:13

Overview

After creating an application on Tencent Cloud Agent Development Platform, completing the application configuration, conducting a dialogue test, and publishing to obtain the AppKey, you can use the dialogue API to interact with Tencent Cloud Agent Development Platform.
The platform provides two commonly used access methods: WebSocket and HTTP SSE.

1. WebSocket Access Method Overview

WebSocket is connection-oriented and provides a full-duplex channel. A Token must be obtained from the server before establishing a connection. This token is required to create a WebSocket connection with the server. The Token is only valid at the time of connection establishment and becomes invalid after the connection is successfully established.
Access Process Diagram:

2. HTTP SSE Access Method Overview

HTTP SSE is a one-way communication channel. After the client initiates an HTTP request, the server continuously pushes streaming data to the client. At this point, bidirectional interaction is not supported.

Note:
If you want to learn about relevant APIs on the configuration end, please visit: API document.

Dialog API Documentation (WebSocket)

Last updated：2025-05-29 16:21:28

WebSocket is connection-oriented and provides a full-duplex channel. A Token must be obtained from the server before establishing a connection. This token is required to create a WebSocket connection with the server. The Token is only valid at the time of connection establishment and becomes invalid after the connection is successfully established.
Access Process Diagram:

1. Retrieve the Token for Establishing a Websocket Connection

1.1 Calling Method

Call the GetWsToken API using the Tencent Cloud SDK to retrieve the Token. For details, refer to the GetWsToken document.

Note:

The obtained Token **is for one session connection only**, and **will expire**. Please establish a persistent connection promptly after obtaining the Token. If you need to establish another connection, you must reobtain the Token.
Tag-related values are no longer passed using this interface. Please refer to custom_variables in 3.1 Data Structure to pass in knowledge base search scope related parameters.

1.2 How to Obtain an AppKey

In the **Application Management** interface, find your running application (must be published first), click **Call**, and the "Call Information" window will pop up.

In the **Call Information** window, you can see the AppKey. Click **Copy** to replicate it.

2. Using Token to Create a Websocket Connection

Request Address: `wss://wss.lke.tencentcloud.com/v1/qbot/chat/conn/?EIO=4&transport=websocket`

Request Protocol: Socket.IO v4 (Reference Document)

Note:

You can refer to the Dialogue Endpoint Demo Code at the bottom of this page.

2.1 Establishing Connections

After the Websocket connection is established successfully, the server responds as follows:

```
0{"sid":"xxx","upgrades":[],"pingInterval":25000,"pingTimeout":5000}
```

2.2 Transmitting Token Authentication

Pass authentication via Socket.IO's messaging.

After establishing the connection and receiving the server response, send Token authentication. The Token format is as follows:

```
40{"token":"xx-xx-xx-xx-xx"}
```

2.3 Heartbeat Packet Processing

The heartbeat packet sent by the server is as follows ("2" is the content of the heartbeat packet):

```
2
```

At this point, the client needs to respond (the "3" is the content that requires a response).

```
3
```

Note:

Socket.io V4 has heartbeat packets, which must be handled; otherwise, the connection will be disconnected by the server.

If you implement the Client yourself, note that you need to handle the heartbeat packet. The Demo provided in this document has automatically handled the heartbeat packet.

3. WebSocket Supported Events

The format of Socket.IO events is as follows. Pay attention to its structure when implementing. It is advisable to use the standard Client provided by Socket.IO as much as possible, or refer to the frontend and backend Demo provided below in this document.

```
42["type",{"payload":{event body}}]
```

3.1 Sending Events

Event name: send

Event direction: **Frontend > Backend**

Note:

Before sending a send event, you need to publish an application.

When a user sends a message (send event), the server will return the message as-is (reply event, where is_from_self = true) so that the message is confirmed to be received by the server and the corresponding message ID and timestamp are updated.

If you need to carry knowledge tag information, include it when obtaining the token.

Data structure:

Name	Type	Required or Not	Description
request_id	string(255)	Yes	Request ID, used to identify a request (for message concatenation, it is advisable to use a different request_id for each request)
session_id	string(64)	Yes	session ID, used to identify a session (provided by an external system. It is advisable that different user ends pass in **different** session_ids; otherwise, message records of different users within the same application may get mixed up). **Parameter Length: 2 to 64 characters** **Verification Rule: ^[a-zA-Z0-9_-]{2,64}$, a uuid can generally be used to generate this value** **uuid Example: 1b9c0b03-dc83-47ac-8394-b366e3ea67ef**
content	string(6000)	Yes	Message content. If sending pictures, transmit markdown-formatted image links herein, for example , where the image link must be publicly readable.
custom_variables	map[string]string	No	Customize the value of **API parameter**. Multiple key:value pairs can be configured, where the key is the parameter name of the custom parameter, and the value is the runtime value of the corresponding custom parameter. When using the value in the knowledge base scope settings, if multiple parameter values are passed, separate them with an English vertical bar (|), for example: "user1|user2". **Note:**

			If you need to specify a knowledge base for retrieval in a question, you can input relevant tag values through this field. For details, refer to Set Knowledge Base Search Scope.
system_role	string(2000)	No	Role instruction (prompt content). If empty, use the application's default configuration; when filled, take the current value.

3.2 Replying to Events

Event name: reply

Event direction: **Backend > Frontend**

Note:

If the received message has is_evil == true, it indicates that the message hits sensitive content and the sending fails.

Exceeding the concurrency limit causes a queue timeout, resulting in a "concurrency limit exceeded" error.

Data structure

Name	Type	Description
request_id	string(255)	Request ID, used to identify a request (for message concatenation, it is advisable to use a different request_id for each request)
content	string	Reply message content
file_infos	Object array	File information
record_id	string(64)	Message unique ID
related_record_id	string(64)	Associated message unique ID
session_id	string(64)	session ID, used to identify a session (provided by an external system. It is advisable that different user ends pass in **different** session_ids; otherwise, message records of different users within the same application may get mixed up).
is_from_self	bool	Whether the message is sent from the client
can_rating	bool	Whether this message record can be evaluated
timestamp	int64	message timestamp (in seconds)
is_final	bool	Whether the message has been completely output In streaming mode, messages are returned multiple times, and each return overwrites the previous answer.

		When is_final == true, the stop generation button is hidden, and the like/dislike buttons are displayed.
is_evil	bool	Whether it hits sensitive content **Note: After message uplink, sensitive content detection will be performed first, and a [reply] event will be returned to inform the sensitive content detection result. Normal business logic processing will only proceed after the sensitive issue detection passes.**
is_llm_generated	bool	Whether it is model-generated content
reply_method	uint8	Reply method 1: Reply from a large model 2: Reply to an unknown question 3: Reply to a rejected question 4: Sensitive reply 5: Preferentially reply with accepted Q&A pairs 6: Welcome message reply 7: Reply when concurrency limit is exceeded
knowledge	Object array	Knowledge hit
custom_params	string array	User-customized business parameters, used for passing through business parameters in QA Description: This field may be empty. If it is empty, this field will not be returned.

Data structure for knowledge hits

Name	Type	Description
id	string	Knowledge hit ID
type	uint32	Type of knowledge hit: 1: QA 2: Document fragment

StatisticInfo LLM statistical information:

Name	Type	Description
ModelName	string	Model name

FirstTokenCost	uint32	First token duration
TotalCost	uint32	Total reasoning time
InputTokens	uint32	Number of input tokens
OutputTokens	uint32	Number of output tokens
TotalTokens	uint32	Total number of input and output tokens

3.3 Token Statistics Event

Event name: token_stat

Event direction: **Backend > Frontend**

Data structure

Name	Type	Description
session_id	string(64)	Session id
request_id	string(255)	Request id for sending corresponding event
record_id	string(64)	Message record id for sending corresponding event
status_summary	string	Current conversation status: processing, success, failed
status_summary_title	string	Current conversation status description
elapsed	int	Call duration of this round, unit ms
token_count	int	Token consumption in this round's request (when containing multiple processes, calculations will be aggregated)
procedures	Object array	List of invocation processes

Data structure for the list of invocation procedures

Name	Type	Description
name	string	English name, in one-to-one correspondence with the following title field. knowledge, large_language_model

title	string	Invocation process description, corresponding to the name field, with the following Chinese meanings: Call the knowledge base The large model replies
status	string	Invocation process status: processing, success, failed
input_count	int	Token consumption of this process input
output_count	int	Token consumption of this process output
count	int	Token consumption of this process: input + output

Example:


```
[
    "token_stat",
    {
        "type": "token_stat",
        "payload": {
            "elapsed": 1616,
            "order_count": 50000000,
            "procedures": [
                {
                    "count": 323,
                    "input_count": 308,
                    "name": "knowledge",
                    "output_count": 15,
                    "status": "success",
                    "title": "Call the knowledge base"
                }
            ],
            "record_id": "Hpe_20240625_185659_215_EsH2uf8L",
            "request_id": "8PUcDU6xyQ-301747294000",
            "session_id": "2d071ef7-ef76-44df-84a4-9210672ed700c8",
            "status_summary": "success",
            "status_summary_title": "Call the knowledge base"
            "token_count": 323,
            "used_count": 553
        },
        "message_id": "89d91395-06bc-4f2e-b240-06f7b4498b0c6e"
    }
]
```

3.4 Evaluating Events

Event name: rating

Event direction: **Bidirectional**

Note:

When the client sends an evaluation event, it will also receive this event so that it can confirm the message is sent successfully.

Data structure

Name	Type	Required or Not	Description
record_id	string(64)	Yes	Message ID (of the reply event being evaluated)
score	uint8	Yes	Score 1: Like 2: Dislike
reasons	string array	No	Selected reasons (user feedback content, can have multiple)

3.5 Stopping Event Generation

event name: stop_generation

Event direction: **Frontend > Backend**

Data structure

Name	Type	Required or Not	Description
record_id	string(64)	Yes	Message ID (of the reply event message that needs to be stopped)

3.6 Reference Source Event

Event name: reference

Event direction: **Backend > Frontend**

Data structure

Name	Type	Description
record_id	string(64)	Message unique ID
references	Object array	Reference source

Data structure of references

Name	Type	Description
id	uint64	The usage method of this ID is divided into two parts. Refer to the following two examples: 1. When the reference source type is 1, 2, or 3, you can call the get source detail list API to view the reference source details. If you need to preview, you can obtain the corresponding parameters through the above API and concatenate them for redirection access: https://lke.tencentcloud.com/preview?id=${docid}&botBizId=${appid}&page=${redirected document page number}&name=${redirected sheet name}&test=1 **Note:** The id field corresponds to the ReferBizIds field in DescribeRefer 2. When the reference source type is 4, this id indicates the serial numbers of multiple reference sources.
type	uint32	Reference source type 1: QA 2: Document fragment 3: Documentation
url	string	Reference link (used when the source type is document fragment only)
name	string	Reference source name
doc_id	uint64	Reference source document ID
doc_biz_id	uint64	Reference source document business ID, callable document detail API to check corresponding document's basic information
doc_name	string	Reference source document name
qa_biz_id	string	Reference source QA business ID

Example

Reference source type: **1: QA; 2: Document fragment; 3: Document**.

Call Tencent Cloud SDK's DescribeRefer to view reference source details. At this point, the id field corresponds to the ReferBizIds field in DescribeRefer. At this point, the id field corresponds to the ReferBizIds field in DescribeRefer.

reference Event:

```
[
    "reference",
    {
        "type": "reference",
        "payload": {
            "record_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            "references": [
                {
                    "doc_biz_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "doc_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "doc_name": "One Hundred Idioms and Their Historical Figures' S
                    "id": "xxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "name": "One Hundred Idioms and Their Historical Figures' Stori
                    "qa_biz_id": "0",
                    "type": 2,
                    "url": "xxxxxxxxxxxxxxxxxxxxxxxxxxx"
                }
            ],
            "trace_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxx"
        },
        "message_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxx"
    }
]
```

3.7 Error Event

Event name: error

Event direction: **Backend > Frontend**

Data structure

Name	Type	Description
request_id	string(255)	Request ID, used to identify a request (for message concatenation, it is advisable to use a different request_id for each request)
error	Object	Error

error data structure

Name	Type	Description
code	uint32	Error Code

message	string	Error message.

3.8 Thinking About Events

Event name: thought

Event direction: **Backend > Frontend**

Note:

This event currently only returns when using the DeepSeek-R1 model.

Data structure

Name	Type	Description
elapsed	int	Call duration of this round, unit ms
procedures	Object array	List of invocation processes
record_id	string(64)	Message record id for sending corresponding event
request_id	string(255)	Request id for sending corresponding event
session_id	string(64)	Session id
trace_id	string	Link id

Data structure for procedures

Name	Type	Description
debugging	Object array	Debug process information
index	uint32	Process index
name	string	English name, in one-to-one correspondence with the following title field. knowledge large_language_model thought
title	string	Invocation process description, corresponding to the name field, with the following Chinese meanings: Call knowledge base Large model reply Think
status	string	Status: processing - in use success - successful failed - failed

icon	string	Icon, used for show
elapsed	uint32	Current request execution time, unit ms

Data structure for debugging

Name	Type	Description
content	string	Output during debugging

Effect

Example


```
[
    "thought",
    {
        "type": "thought",
        "payload": {
            "elapsed": 0,
            "is_workflow": false,
            "procedures": [
                {
                    "debugging": {
                        "content": " Okay, the user asked about the features of Dee
                    },
                    "elapsed": 9954,
                    "icon": "https://lke-realtime-1251316161.cos.ap-guangzhou.myqcl
                    "index": 0,
                    "name": "thought",
                    "plugin_type": 0,
                    "status": "processing",
                    "switch": "",
                    "title": "think"
                    "workflow_name": ""
                }
            ],
            "record_id": "piQ_20250208_140511_254_45ja7HPq",
            "request_id": "rjbuFuYeDB-8487350785",
            "session_id": "dfd04cd0-ef69-4fb6-a447-a0b44018a93f",
            "trace_id": "9a9613dbaa9edd27958bd1180c524295 / piQ_20250208_140511_254
            "workflow_name": ""
        },
        "message_id": "7070ef0d-4c28-4d5a-abc8-2ea2e7fa027d"
```

```
        }
]
```

4. Error Code

Error Code	Error message.
400	request parameter error, please see the access documentation
460001	Token Verification Failure
460002	Event handler does not exist
460004	Application does not exist
460006	Message does not exist or insufficient permissions
460007	Session creation failed.
460008	Prompt fail to render
460009	Visitor user does not exist.
460010	The session does not exist or has insufficient permissions.
460011	Exceed the concurrency limit
460020	Model request timeout.
460021	Knowledge base unpublished
460022	Visitor creation failed.
460023	Message like/dislike failed.
460024	Invalid tag
460025	Image Analysis failed.
460031	Current application connections exceed the request limit. Try again later.
460032	Current application model has insufficient balance.
460033	Application does not exist or insufficient permissions.
460034	Input content is too long.

460035	Calculation content is too long, has been stopped.
460036	Task process node preview parameter exception
460037	Search resources exhausted, call failed.
460038	The AppID request shows abnormal behavior, call failed.

5. Dialogue Endpoint Demo Code

Demo code describes a complete link establishment and message sending and receiving process.

5.1 Frontend Version

Note:

The get Token API requires the backend to call the Tencent Cloud SDK.

JS version (to be provided subsequently)

5.2 Backend Version

Golang version

Python version

JAVA version

Other programming languages have no Demo available. See document and existing Demo to implement your own.

Dialog API Documentation (HTTP SSE)

Last updated：2025-05-29 16:53:06

HTTP SSE is a one-way channel. After the client initiates an HTTP request, the server continuously pushes streaming data to the client. At this point, bidirectional interaction is not supported.

1. HTTP SSE API Request

Request Address: `https://wss.lke.tencentcloud.com/v1/qbot/chat/sse`

Request Method: POST

Note:

Before triggering the dialogue API, you need to have a published application.

1.1 Parameter Description

Place it in the HTTP Body and send it in the form of JSON. Details are given below:

Name	Type	Required or Not	Description
request_id	string(255)	Yes	Request ID, used to identify a request (for message concatenation, it is advisable to use different request_ids for each request)
content	string(6000)	Yes	Message content. If sending a picture, transmit the markdown-formatted image link here, for example , where the image link must be publicly readable.
session_id	string(64)	Yes	session ID, used to identify a session (provided by an external system. It is advisable for different user ends to input **different** session_ids; otherwise, the message records of different users within the same application may get mixed up). **Parameter Length: 2 to 64 characters** **Verification Rule: ^[a-zA-Z0-9_-]{2,64}$. a uuid can generally be used to generate this value.** **uuid Example: 1b9c0b03-dc83-47ac-8394-b366e3ea67ef**
bot_app_key	string(8)	Yes	Application key (see
visitor_biz_id	string(64)	Yes	Visitor ID (external input, recommended to be unique,

			indicating the user of the current connected session)
visitor_labels	Object array	No	Knowledge Tag (used for search filter in knowledge base); The structure of Knowledge Tag is: [{"name": "subject", "values": ["chinese", "math"]}] Knowledge Tag defines related attributes, attribute identifiers, and tags. The applicable scope of the document selected the "chinese" and "math" tags. **Note: The name field in visitor_labels corresponds to the attribute identifier in the above figure.** **Warning:** This field is about to go offline. Please use the following custom_variables field as a substitution for this field to perform knowledge scope retrieval.
streaming_throttle	int32	No	Streaming response frequency control: Controls the application's response packet frequency. This value indicates how many characters the application accumulates before replying to the caller once. A smaller value results in more frequent responses (smoother experience but higher traffic overhead). If no value is provided or it is set to 0, the system configuration will be used. **Note: This setting item does not accelerate the output time of large models, but only changes the frequency of replying to the caller. Therefore, if the setting is very large, there may be a long time without a reply.**
custom_variables	map[string]string	No	Customize the **API parameter** values. Multiple key:value pairs can be configured, where the key is the name of the custom parameter and the value is the runtime value of the corresponding custom parameter. When using the value in knowledge base scope settings, if multiple parameter values are passed, separate them with an English vertical bar (|), for example: "user1|user2". **Note:** If you need to specify a knowledge base for retrieval in a question, you can pass in relevant tag values

			through this field. For details, refer to Knowledge Base Search Scope Settings.
system_role	string(2000)	No	Role instruction (prompt content), use the application's default setting when empty, take the current value when filled.

data structure of visitor_labels knowledge tag list

Name	Type	Description
name	string	Knowledge tag name
values	string array	knowledge tag value

1.2 How to Obtain an AppKey

In the **Application Management** interface, find your running application (must be published first), click **Invoke**, and the "Invocation Information" window will pop up.

In the **Invocation Information** window, you can see the AppKey. Click **Copy** to replicate it.

Calling the API Using Curl

```
curl -XPOST -vvv --no-buffer --location 'https://wss.lke.tencentcloud.com/v1/qbot/c
--header 'Content-Type: application/json' \\
--data '{
    "content": "message content"
    "bot_app_key": "<your appkey>",
    "visitor_biz_id": "<your visitor id>",
    "session_id": "<your session_id>",
    "visitor_labels": []
}'
```

1.4 Postman Call Example

2. HTTP SSE API Response

2.1 Replying to Events

Event Name: reply

Event direction: **backend > frontend**

Note:

If the received message has is_evil == true, it indicates the message hits sensitive content and sending fails.

Exceeding the concurrency limit causes a queue timeout, resulting in a "Concurrency limit exceeded" error.

data structure

Name	Type	Description
request_id	string(255)	Request ID, used to identify a request (for message concatenation, it is advisable to use different request_ids for each request)
content	string	message content
file_infos	Object array	file information
record_id	string(64)	Message unique ID
related_record_id	string(64)	associated message unique ID
session_id	string(64)	session ID, used to identify a session (provided by an external system. It is advisable for different user ends to input **different** session_ids; otherwise, the message records of different users within the same application may get mixed up).
is_from_self	bool	Whether the message is sent by itself (if it is sent by itself, display on the right side of the chat box; otherwise, display on the left side)
can_rating	bool	Whether this message record can be evaluated
timestamp	int64	Message timestamp (in seconds)
is_final	bool	Whether the message has been fully output (In streaming mode, messages are returned multiple times, and each return overwrites the previous answer.) When is_final == true, the stop generation button is hidden, and the like/dislike buttons are displayed.
is_evil	bool	Whether it hits sensitive content **Note: After message uplink, sensitive content detection will be performed first, and a [reply] event will be returned to inform the detection result. Normal business logic processing will only proceed after sensitive content detection passes.**

is_llm_generated	bool	Whether it is model-generated content
reply_method	uint8	Reply method 1: Large model reply 2: Unknown question reply 3: Rejected question reply 4: Sensitive reply 5: Accepted Q&A pair prioritized reply 6: Welcome message reply 7: Reply when the number of concurrencies exceeds the limit
knowledge	Object array	Knowledge of hits
custom_params	string array	User-customized business parameters, used for passing through business parameters in QA Note: This field may be empty. If it is empty, this field does not return.

Data structure for knowledge hits

Name	Type	Description
id	string	ID of knowledge hits
type	uint32	Type of knowledge hits: 1: QA 2: document fragment

StatisticInfo LLM statistical information:

Name	Type	Description
ModelName	string	Model name
FirstTokenCost	uint32	First token duration
TotalCost	uint32	Total time consumed for reasoning
InputTokens	uint32	Number of input tokens
OutputTokens	uint32	Number of output tokens
TotalTokens	uint32	Total number of input and output tokens

2.2 Token Statistics Event

Event name: token_stat

Event direction: **backend > frontend**

data structure

Name	Type	Description
session_id	string(64)	Session id
request_id	string(255)	Request id for sending corresponding event
record_id	string(64)	Message record id for sending corresponding event
status_summary	string	Current conversation status: processing, success, failed
status_summary_title	string	Current conversation status description
elapsed	int	Call duration of this round, unit ms
token_count	int	Token consumption in this round's request (when containing multiple processes, calculations will be aggregated)
procedures	Object array	List of invocation processes

Data structure for the list of invocation procedures

Name	Type	Description
name	string	English name, in one-to-one correspondence with the following title field. knowledge, large_language_model
title	string	Invocation process description, corresponding to the name field, with the following Chinese meanings: Call knowledge base, large model replies
status	string	Invocation process status: processing, success, failed
input_count	int	Token consumption of this process input
output_count	int	Token consumption of this process output
count	int	Token consumption of this process: input + output

Example:

```
[
    "token_stat",
    {
        "type": "token_stat",
        "payload": {
            "elapsed": 1616,
            "order_count": 50000000,
            "procedures": [
                {
                    "count": 323,
                    "input_count": 308,
                    "name": "knowledge",
                    "output_count": 15,
                    "status": "success",
                    "title": "Call knowledge base"
                }
            ],
            "record_id": "Hpe_20240625_185659_215_EsH2uf8L",
            "request_id": "8PUcDU6xyQ-301747294000",
            "session_id": "2d071ef7-ef76-44df-84a4-9210672ed700c8",
            "status_summary": "success",
            "status_summary_title": "Call knowledge base"
            "token_count": 323,
            "used_count": 553
        },
        "message_id": "89d91395-06bc-4f2e-b240-06f7b4498b0c6e"
    }
]
```

2.3 Reference Source Event

Event name: reference

Event direction: **backend > frontend**

data structure

Name	Type	Description
record_id	string(64)	Message unique ID
references	Object array	Reference source

data structure of references

Name	Type	Description
id	uint64	The method of use for this ID is divided into two parts. You can refer to the following two examples: 1. When the reference source type is 1, 2, or 3, you can call the get source detail list API to view the reference source details. **Note:** The id field corresponds to the ReferBizIds field in DescribeRefer 2. When the reference source type is 4, this id indicates the serial numbers of multiple reference sources.
type	uint32	Reference source type 1: QA 2: document fragment 3: document
url	string	Reference source link (used only when the reference source type is document fragment)
name	string	Reference source name
doc_id	uint64	Reference source document ID
doc_biz_id	uint64	Reference source document business ID, callable document detail API to check corresponding document's basic information
doc_name	string	Reference source document name
qa_biz_id	string	Reference source QA business ID

Example

Reference source type

1: Q&A

2: Document Fragment

3: document

You can call the Tencent Cloud SDK's DescribeRefer to view the reference source details. At this point, the id field corresponds to the ReferBizIds field in DescribeRefer.

reference event:

```
[
```

```
    "reference",
    {
        "type": "reference",
        "payload": {
            "record_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            "references": [
                {
                    "doc_biz_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "doc_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "doc_name": "One Hundred Idioms and Their Historical Figures' S
                    "id": "xxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "name": "One Hundred Idioms and Their Historical Figures' Stori
                    "qa_biz_id": "0",
                    "type": 2,
                    "url": "xxxxxxxxxxxxxxxxxxxxxxxxxxx"
                }
            ],
            "trace_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxx"
        },
        "message_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxx"
    }
]
```


2.4 Error Event

Event name: error

Event direction: **backend > frontend**

data structure

Name	Type	Description
request_id	string(255)	Request ID, used to identify a request (for message concatenation, it is advisable to use different request_ids for each request)
error	Object	Error

Erroneous data structure

Name	Type	Description
code	uint32	Error Codes
message	string	Error message.

Example:

```
event:reply
data: {"type": "reply", "payload": {"can_rating": false, "content": "Who are you",

event:reply
data: {"type": "reply", "payload": {"can_rating": true, "content": "I am a Large Mo

event:error
data: {"type": "error", "error": {"code": 460004, "message": "application does not
```

Note:

When using the API, judge whether the obtained value is 200. If yes, it will be a normal return.

2.5 Thinking About Events

Event name: thought

Event direction: **backend > frontend**

Note:

This event currently only returns when using the DeepSeek-R1 model.

data structure

Name	Type	Description
elapsed	int	Call duration of this round, unit ms
procedures	Object array	List of invocation processes
record_id	string(64)	Message record id for sending corresponding event
request_id	string(255)	Request id for sending corresponding event
session_id	string(64)	Session id
trace_id	string	Link id

Data structure for procedures

Name	Type	Description
debugging	Object array	Debugging process information
index	uint32	Process index
name	string	English name, in one-to-one correspondence with the title field below. knowledge large_language_model thought

title	string	Invocation process description, corresponding to the name field, with the following Chinese meanings: Call knowledge base Large model reply Thinking
status	string	Status: processing - in use success - successful failed - failed
icon	string	Icon, used for show
elapsed	uint32	Current request execution time, unit ms

data structure of debugging

Name	Type	Description
content	string	Output during debugging

Effect

Example


```
[
    "thought",
    {
        "type": "thought",
        "payload": {
            "elapsed": 0,
            "is_workflow": false,
            "procedures": [
                {
                    "debugging": {
                        "content": " Okay, the user asked about the features of Dee
                    },
                    "elapsed": 9954,
                    "icon": "https://lke-realtime-1251316161.cos.ap-guangzhou.myqcl
                    "index": 0,
                    "name": "thought",
                    "plugin_type": 0,
                    "status": "processing",
                    "switch": "",
                    "title": "Think"
                    "workflow_name": ""
```

```
                }
            ],
            "record_id": "piQ_20250208_140511_254_45ja7HPq",
            "request_id": "rjbuFuYeDB-8487350785",
            "session_id": "dfd04cd0-ef69-4fb6-a447-a0b44018a93f",
            "trace_id": "9a9613dbaa9edd27958bd1180c524295 / piQ_20250208_140511_254
            "workflow_name": ""
        },
        "message_id": "7070ef0d-4c28-4d5a-abc8-2ea2e7fa027d"
    }
]
```


3. Error Code

Error Code	Error message.
400	request parameter error, please see the access documentation
460001	Token Verification Failure
460002	Event handler does not exist
460004	Application does not exist
460006	Message does not exist or insufficient permissions
460007	Session creation failed
460008	Prompt fail to render
460009	Visitor user does not exist
460010	The session does not exist or has insufficient permissions.
460011	Exceed the concurrency limit
460020	Model request timeout
460021	Knowledge base unpublished
460022	Visitor creation failed
460023	Message like/dislike failed
460024	Invalid tag

460025	Image Analysis failed.
460031	Current application connections exceed the request limit. Try again later.
460032	Current application model has insufficient balance.
460033	Application does not exist or insufficient permissions.
460034	Input content is too long.
460035	Content is too long to calculate, has been stopped.
460036	Task process node preview parameter exception
460037	Search for resource exhausted, call failed.
460038	The request from this AppID exhibits abnormal behavior, call failed.

4. Dialogue Endpoint Demo Code

Demo code describes a complete link establishment and message sending and receiving process.
JS version (provided subsequently)

4.1 Backend Version

Golang version
Python version
JAVA version
Other programming languages have no Demo available. See the document and existing Demos to implement your own.

Offline Document Upload

Last updated：2025-05-29 14:55:02

Overview

1. The customer has purchased Tencent Cloud's cos, which is used in their own other scenarios and managed by the customer themselves.
2. Customer purchases or try Tencent Cloud Agent Development Platform and needs to perform file upload, download, access, etc. At this time, the cos to which the files belong is the cos internally applied for by the Tencent Cloud Agent Development Platform team. This cos is maintained by the Tencent Cloud Agent Development Platform product team, including file storage location, file access permission, etc. It is not supported to save files to the cos address purchased by the customer themselves when using Tencent Cloud Agent Development Platform. The cos bucket address and other related information will be provided when calling the temporary key API [DescribeStorageCredential].
3. When using the API to call the SaveDoc API, there are three steps: obtain a temporary key, upload the file to the cos of the Tencent Cloud Agent Development Platform, and call SaveDoc to save; it cannot be done in one step.

Definitions

Offline Documentation: Offline documentation mainly refers to the [File] and [QA] under the [Knowledge Management] of Tencent Cloud Agent Development Platform.
Offline Documentation

Real-time Document (Not Supported): Real-time document mainly refers to the upload of documents in the dialogue interface.

Step-by-Step Instructions

Upload files offline in three steps [The first two steps also apply to real-time documents]:
1. Call the DescribeStorageCredential API to get a temporary key.
2. Upload the file to the cos provided by Tencent Cloud Agent Development Platform.
3. Call the SaveDoc API to store the basic information of the file in Tencent Cloud Agent Development Platform. As shown in the figure below:

Get Temporary Key

Reference: Obtain a temporary key for file upload.

1. Add the request parameter FileType, where FileType is a normal file name type suffix, such as xlsx, pdf, docx, png, etc.

1.1 Fill in the FileType field. The obtained key has only upload permission. The return parameter uses UploadPath.

1.2 Do not fill in the FileType field. The obtained key has only download permission.

1.3 **TypeKey distinguishes between offline files and real-time files**.

1.4 BotBizID is a required option.

1.5 Each file upload needs to get a different temporary key, and the temporary key has a valid period.

Note:

Note: Do not mix the parameter values of TypeKey. Use "offline" for document upload and "realtime" for real-time document upload. If it is empty, the default value is "offline".

2. Add the request parameter IsPublic (case-sensitive for public or private scenarios)

Example

Public scenario: /public/12332323/21321321321/image/1.png

Private scenario: /corp/12332323/12332323/doc/1.pdf

Note:

IsPublic: Select a scene when uploading files or images. When uploading offline documentation, IsPublic is set to false. When uploading real-time documentation, if it is an image, IsPublic is set to true; if it is other types of files, IsPublic is set to false.

API response parameters below are used in subsequent steps.

Parameter Name	Description
Bucket	bucket location <offline documentation and real-time documentation differ here>
TmpSecretId	Temporary secretID
TmpSecretKey	Temporary secretKey
Token	token for uploading
UploadPath	In subsequent steps, use and cooperate to upload to cos

API response sample:

```
{
    "code": 0,
```

```
    "data": {
        "Response": {
            "Bucket": "lke-intl-1251316161",
            "CorpUin": "0",
            "Credentials": {
                "TmpSecretId": "*******************************",
                "TmpSecretKey": "*******************************",
                "Token": "*******************************"
            },
            "ExpiredTime": 1722234043,
            "FilePath": "",
            "ImagePath": "",
            "Region": "ap-jakarta",
            "RequestId": "771ca2ee-03a7-487f-b77e-ccbb240f3cb8",
            "StartTime": 1722233443,
            "Type": "cos",
            "UploadPath": "/corp/17468272416xxxxxxxxxx6.txt"
        }
    },
    "message": "OK",
    "reqId": "749c46ae-8070-457f-84be-5d0daa5cce05"
}
```


2. Call the cos Storage Interface Provided by Tencent Cloud to Store Files in the cos of Tencent Cloud Agent Development Platform

Reference: API doc > PUT Object

Note:

1. Require the use of TmpSecretId, TmpSecretKey, Token, and UploadPath from Step 1.

2. Upload files to cos [putObject]. You cannot directly copy code from the API Explorer. **The following two have differences in the code when uploading files via code. See the following code example (demo) for details:**

<API-Explorer uses a fixed key.>

<Note: The code upload uses a temporary key.>

3. Calling SaveDoc to Save Metadata

Reference: API doc > Save Document

Important field descriptions. For other fields, refer to the API document.

Field	Description
IsRefer	When it is true, during the user-side dialogue process, if knowledge information is matched, relevant links will be provided. The effect is as follows.
CosUrl	The uploadPath obtained in Step 1

CosHash	The "X-Cos-Hash-Crc64ecma" in the response header of Step 2
Etag	The "Etag" in the response header of Step 2
Opt	Document operation types: 1: Batch import (batch import of Q&A pairs); 2: Document import (normal import of a single document) **When this value is 1 and the imported file is an excel file, the excel header will be verified to meet expectations**

Code Demo

offline_upload_and_save_doc_20240729.zip

Tencent Cloud Agent Development Platform Operation COS Guide

Last updated：2025-05-29 14:55:14

Core Conformance Gaps

The main differences between the built-in COS service in Tencent Cloud Agent Development Platform (TCADP) and the COS purchased by customers are as follows:

1. Storage ownership

TCADP's data is stored in the platform's COS, with no need to separately purchase storage services, and costs are billed per character.

Does not support uploading files to customer-purchased COS

2. Permission control

The platform implements fine-grained permission management through temporary keys to avoid unauthorized operations impacting other users' data.

Note:

Using Tencent Cloud Agent Development Platform product, no need to purchase cos, does not support uploading files to your purchased cos.

Operation Process Comparison

Operation Steps	Tencent Cloud Agent Development Platform **Built-In COS**	**Customer Self-Purchased COS**
1. Key Acquisition	Call the API document > DescribeStorageCredential API to obtain a temporary key	Use the fixed key (SecretId/Key) obtained when purchased
2. Client Initialization	Create a low-privilege CosClient using a temporary key	Create a CosClient using a fixed key
3. Data Operation	Call COS API (requires temporary key permission match)	Direct call COS API

Note:

Incorrect parameter combination of temporary key can cause operation failure (such as 403 insufficient permissions).

Parameter Configuration Guide for Temporary Keys

Parameter combination examples for common scenarios, please fill in according to actual needs (gray fields are required):

Scenario	Parameter Combination	Remark
Knowledge Base Upload Documents (Offline)	{"BotBizId":"182693390xxxx", "FileType":"pdf", **"IsPublic":false**}	Refer to Offline Document Upload
File Download	{"BotBizId":"182693390xxxx", **"IsPublic":false**}	No need to specify a file type

Common Issues

Why is a temporary key needed?

The platform isolates user data through dynamic permissions to ensure security.

What to do if parameters are filled incorrectly?

Check whether the scenario matches (e.g., uploading images requires IsPublic:true), or verify API input parameters using a packet capture tool on the corresponding operation page.

Can parameters be used across scenarios?

Mixed use is forbidden. For example: TypeKey:"realtime" cannot be used for offline document upload.