Agent 沙箱服务

SDK 使用指南

腾讯云

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

SDK 使用指南
配置环境变量

最近更新时间：2025-09-28 18:50:42

为了方便您的使用，我们兼容了 E2B SDK，您可以复用您的 E2B 工作流，仅需少许操作就可以切换到 AgentSandbox 沙箱服务。

> **说明：**
> 为了保持与 E2B SDK 的兼容，沙箱实例内部包含了基于 E2B Infra 二次开发的部分组件。需注意，您需要使用 E2B SDK 2.0及以上版本。

您可以通过以下三种常见方式设置环境变量，推荐您使用 dotenv 文件来配置环境变量。

dotenv 文件

您可以在您的项目文件夹创建 `.env` 文件，在其中配置环境变量。

.env

```
E2B_DOMAIN=ap-guangzhou.ags.tencentcs.com
E2B_API_KEY=ark_xxxxxxxx
```

然后在您的代码中使用 dotenv 库来引入，通过该种方式引入的环境变量在当前代码范围内生效。

Python

```
from dotenv import load_dotenv
load_dotenv()
```

终端环境变量

您可以通过在执行代码的终端中设置环境变量，通过该种方式引入的环境变量在通过该终端启动的程序内生效。

Bash

```
export E2B_DOMAIN=ap-guangzhou.ags.tencentcs.com
export E2B_API_KEY=ark_xxxxxxxx
```

您也可以在终端配置文件 .bashrc 中添加以上两行来全局应用，该种方式在所有 bash 终端中生效。

内嵌环境变量

您可以在代码中通过导入系统库来改变程序的环境变量，通过该种方式引入的环境变量在当前代码范围内生效。

Python

```
import os
os.environ["E2B_DOMAIN"]="ap-guangzhou.ags.tencentcs.com"
os.environ["E2B_API_KEY"]="ark_xxxx"
```

沙箱实例

最近更新时间：2025-09-28 18:50:42

前提条件

- 在执行下文所有代码前，请先按照 配置环境变量 部分完成环境变量设置。
- 若您使用 e2b_code_interpreter sdk 且不指定 template，必须已在账号下创建名为 code-interpreter-v1 的代码解释器沙箱工具。

创建沙箱实例

Python

```python
from e2b import Sandbox
from e2b_code_interpreter import Sandbox as CodeSandbox
# 创建一个沙箱工具名为"code-interpreter-v1"的沙箱实例
sandbox = Sandbox.create(template="code-interpreter-v1")
# 您也可以从e2b_code_interpreter中导入Sandbox,默认模板为"code-interpreter-v1"
sandbox = CodeSandbox.create()
# 您可以在创建时指定超时时间
# 当沙箱的运行时间超过超时时间时，沙箱将会被自动删除
sandbox = Sandbox.create(
    template="code-interpreter-v1",
    timeout=300 # 单位为秒
)
```

说明：
create 的 metadata,envs,secure,allow_internet_access 参数暂不可用，请期待后续功能更新。

获取沙箱实例列表

您可以使用 Sandbox.list() 获取到分页器，并使用分页器来以页为单位获取具体的沙箱实例信息。

Python

```python
from e2b_code_interpreter import Sandbox
# 调用list方法获取分页器
paginator = Sandbox.list()
```

```
# 若分页器还有下一页，则取出下一页
while paginator.has_next:
    page = paginator.next_items()
    # 打印当前取出页的沙箱实例信息
    for sandbox_info in page:
        print(sandbox_info)
# 您可以指定一页中有多少实例
paginator_limit_5 = Sandbox.list(limit=5)
# 您可以获取当前分页器的偏移值
print(paginator_limit_5.next_token) # None
if paginator_limit_5.has_next:
    paginator_limit_5.next_items()
print(paginator_limit_5.next_token) # offset_5
```

获取沙箱实例信息

您可以通过使用 get_info 方法来获取沙箱信息。

Python

```
from e2b_code_interpreter import Sandbox
# 创建沙箱实例
sandbox = Sandbox.create()
# 获取沙箱实例信息
info = sandbox.get_info()
print(info)
# 输出结果示例
# SandboxInfo(
# sandbox_id='1234567890qwertyuiopasdfghjklzxf',
# sandbox_domain=None,
# template_id='sdt-12345678',
# name='code-interpreter-v1',
# metadata={},
# started_at=datetime.datetime(1970, 1, 1, 0, 0, 0,
tzinfo=tzoffset(None, 28800)),
# end_at=datetime.datetime(1970, 1, 1, 0, 0, 0, tzinfo=tzoffset(None,
28800)),
# state=<SandboxState.RUNNING: 'running'>,
# cpu_count=1,     # 此处的cpu无实际意义，请使用云API查询沙箱的真实CPU设定
# memory_mb=2048, # 此处的memory无实际意义，请使用云API查询沙箱的真实内存设定
# envd_version='0.2.10',
# _envd_access_token='sit_12345678901234567890123456789012_1234567890'
```

```
# )
```

删除沙箱实例

Python

```python
from e2b_code_interpreter import Sandbox
# 创建沙箱实例
sandbox = Sandbox.create()
# 删除沙箱实例
sandbox.kill()
```

文件系统

最近更新时间：2025-09-28 18:50:42

前提条件

- 在执行下文所有代码前，请先按照 配置环境变量 部分完成环境变量设置。
- 若您使用 e2b_code_interpreter sdk 且不指定 template，必须已在账号下创建名为 code-interpreter-v1 的代码解释器沙箱工具。

文件读取与下载

从沙箱实例中读取文件内容，或下载文件到本地。

Python

```python
from e2b_code_interpreter import Sandbox
# 创建沙箱实例
sandbox = Sandbox.create()
# 从沙箱实例中读取文件
file_content = sandbox.files.read("your-file-path")
# 从沙箱实例中下载文件到本地
with open("local-path","wb") as file:
    # 以二进制形式读取文件
    file.write(sandbox.files.read("your-file-path","bytes"))
```

文件写入与上传

您可以使用 files 类的 write 和 write_files 方法向沙箱实例中写入文件。

Python

```python
from e2b_code_interpreter import Sandbox
# 创建沙箱实例
sandbox = Sandbox.create()
# 向沙箱实例中写入单个文件
sandbox.files.write("your-file-path","file-content")
# 向沙箱实例中上传文件
with open("local-path","rb") as file:
    # 以二进制形式上传文件
    sandbox.files.write("your-file-path",file)
```

```
# 向沙箱实例中写入多个文件
sandbox.files.write_files(
    [
        {"path": "your-path-1", "data": "file-content"},
        {"path": "your-path-2", "data": "file-content"},
    ]
)
```

检查文件是否存在

您可以使用 files 类的 exists 方法来检测沙箱实例中是否存在某个文件。

Python

```
response = sandbox.files.exists("file_name")
# 存在 True
# 不存在 False
```

重命名文件

您可以使用 files 类的 rename 方法来重命名沙箱实例中的某个文件。

Python

```
response = sandbox.files.rename("file_name_before","file_name_after")
```

创建文件夹

您可以使用 files 类的 make_dir 方法来在沙箱实例中创建文件夹。

Python

```
response = sandbox.files.make_dir("dir_name")
```

监控文件变化

您可以使用 files 类的 watch_dir 方法来创建一个文件监控器 watch_handle 类。
通过 watch_handle 类的 get_new_events 方法可以获得这段时间的文件变更事件。

Python

```python
# 创建一个文件监控器
watch_handle = sandbox.files.watch_dir(".")
# 创建一个tempfile.txt并写入内容
sandbox.files.write("tempfile.txt","temp file")
# 删除tempfile.txt
sandbox.files.remove("tempfile.txt")
# 打印这段时间的所有事件
for event in watch_handle.get_new_events():
    print(event)
# 事件示例
# FilesystemEvent(name='tempfile.txt', type=<FilesystemEventType.CREATE:
'create'>)
# FilesystemEvent(name='tempfile.txt', type=<FilesystemEventType.CHMOD:
'chmod'>)
# FilesystemEvent(name='tempfile.txt', type=<FilesystemEventType.WRITE:
'write'>)
# FilesystemEvent(name='tempfile.txt', type=<FilesystemEventType.REMOVE:
'remove'>)
# 停止文件监控器
watch_handle.stop()
```

终端命令

最近更新时间：2025-09-28 18:50:42

前提条件

- 在执行下文所有代码前，请先按照 配置环境变量 部分完成环境变量设置。
- 若您使用 e2b_code_interpreter sdk 且不指定 template，必须已在账号下创建名为 code-interpreter-v1 的代码解释器沙箱工具。

执行命令

您可以使用 commands 类的 run 方法在沙箱实例中运行终端命令。

Python

```python
from e2b_code_interpreter import Sandbox
# 创建沙箱实例
sandbox = Sandbox.create()
# 执行终端命令
response = sandbox.commands.run("ls")
# 可以通过user参数指定执行命令的身份，支持user和root
response = sandbox.commands.run("ls",user="root")
```

流式返回

Python

```python
# 支持流式返回
command = """
echo foo
sleep 1
echo bar
"""
# 您可以根据您的需要传递on_stdout和on_stderr回调
sandbox.commands.run(
    command,
    on_stdout=lambda data: print(data),
    on_stderr=lambda data: print(data)
)
```

后台执行

Python

```
# 支持后台执行命令
echo_code = """
for i in {1..3}; do
    echo -n $i
    sleep 1
done
echo done
"""
# 后台执行指令
echo_handler = sandbox.commands.run(echo_code,background=True)
# 等待后台执行的命令完成
response = echo_handler.wait(on_stdout=lambda data:print(data))
```

发送 stdin 终端输入

Python

```
import asyncio
async def send_stdin_test():
    # 等待用户输入
    read_code = """
    echo test_stdin
    read test
    echo $test
    """
    # 执行命令
    read_handler = sandbox.commands.run(read_code,background=True)
    # 创建一个任务来等待命令完成并显示输出
    wait_task = asyncio.create_task(
        asyncio.to_thread(
            read_handler.wait,
            on_stdout=lambda data: print(data, end=""),
        )
    )
    # 发送stdin
    sandbox.commands.send_stdin(read_handler.pid,"input\n")
    # 等待命令完成
```

```
    response = await wait_task
asyncio.run(send_stdin_test())
```

列出正在运行的后台命令

Python

```
command_list = sandbox.commands.list()
```

结束后台运行的命令

Python

```
command_list = sandbox.commands.list()
for command in command_list:
    sandbox.commands.kill(command.pid)
```

代码运行

最近更新时间：2025-09-28 18:50:42

前提条件

- 在执行下文所有代码前，请先按照 配置环境变量 部分完成环境变量设置。
- 若您使用 e2b_code_interpreter sdk 且不指定 template，必须已在账号下创建名为 code-interpreter-v1 的代码解释器沙箱工具。

运行代码

您可以使用 run_code 方法来在沙箱内置的 jupyter server 中执行代码。

Python

```python
from e2b_code_interpreter import Sandbox
# 创建沙箱实例
sandbox = Sandbox.create()
# 默认执行python代码
response = sandbox.run_code("print(\"hello\")")
```

支持多种语言

您可以指定代码的语言，支持 python, javascript, typescript, java, r, bash。

Python

```python
response = sandbox.run_code("console.log(\"hello\")","javascript")
```

创建代码执行上下文

默认情况下，每个语言会运行在其默认的上下文中，同一上下文共享变量。
您可以通过 create_code_context 来创建上下文。

Python

```python
ctx = sandbox.create_code_context(language="python")
```

您可以通过 context 参数来指定运行时上下文，需要注意，仅能设置 language 和 context 中的一个参数，两者互斥。

Python

```
response = sandbox.run_code("print(\"hello\")",context=ctx)
```

代码执行流式返回

代码执行可以流式返回，您可以通过指定回调函数来接收流式返回的数据。

Python

```
python_stream_code="""
import time
for i in range(10):
    print(i,end='')
    time.sleep(1)
"""
sandbox.run_code(
    python_stream_code,
    on_stdout=lambda data: print(data),
    on_stderr=lambda data: print(data),
    on_result=lambda data: print(data),
    on_error=lambda data: print(data)
)
```

指定代码运行环境变量

您可以为代码运行指定环境变量。

Python

```
response = sandbox.run_code("print(\"hello\")",envs={"foo":"bar"})
```

指定代码执行超时时间

您可以为代码运行指定超时时间，在超过指定时间后终止代码运行。

Python

```
response = sandbox.run_code("print(\"hello\")",timeout=60)  # 单位:秒
```

浏览器操作

最近更新时间：2025-09-28 18:50:42

前提条件

在执行下文所有代码前，请先按照 配置环境变量 部分完成环境变量设置。
使用浏览器沙箱实例前，您需要在 Agent 沙箱服务控制台 创建对应名称的浏览器沙箱工具，并将名称填入下列代码的 template 参数中。

浏览器操作

您在使用浏览器沙箱实例时，可以通过 `live url` 查看浏览器界面， `cdp url` 操作浏览器页面。
您可以使用 Playwright SDK 来通过 `cdp url` 操作浏览器页面，下面的代码演示了如何通过 playwright sdk 跳转到腾讯官网，您可以在 Playwright SDK 文档 中找到更多用法。

Python

```
from e2b import Sandbox
# 创建一个浏览器沙箱
sandbox = Sandbox.create(template="browser-v1")
# 拼接live url
live_url = f"https://{sandbox.get_host(9000)}/novnc/vnc_lite.html?
access_token=
{sandbox._envd_access_token}&path=websockify%3Faccess_token%3D{sandbox._
envd_access_token}"
print(live_url)
# 拼接cdp url
cdp_url = f"https://{sandbox.get_host(9000)}/cdp?access_token=
{sandbox._envd_access_token}"
# 使用playwright通过cdp_url来操作浏览器
from playwright.sync_api import sync_playwright
with sync_playwright() as playwright:
    # 连接到沙箱浏览器实例
    browser = playwright.chromium.connect_over_cdp(
        cdp_url,
        headers={
            "X-Access-Token":str(sandbox._envd_access_token)
        }
    )
    # 获取浏览器Context
    context = browser.contexts[0]
```

```
    # 获取浏览器第一个页面
    page = context.pages[0]
    # 使第一个页面导航到指定网页
    page.goto("http://www.tencent.com")
    # 输出页面的标题
    print(page.title())
```