

API接口描述					数据模型定义		
名称	类型	描述	请求参数	响应内容	名称	描述	数据模型
/agent/invoke	POST	提交异步运行智能体并返回任务ID	request=AgentRequest(user_id: str session_id: str task_id: str query: str system_message: Optional[str])	response = { "user_id": user_id, "session_id": session_id, "task_id": task_id }	AgentRequest	客户端发起的运行智能体的请求数据	class AgentRequest(BaseModel): # 用户唯一标识 user_id: str # 会话唯一标识 session_id: str # 任务唯一标识 task_id: str # 用户的问题 query: str # 系统提示词 system_message: Optional[str] = "你会使用工具来帮助用户。如果工具使用被拒绝, 请提示用户。"
/agent/resume	POST	提交异步恢复被中断的智能体运行返回任务ID	request=InterruptResponse(user_id: str session_id: str task_id: str response_type: str args: Optional[Dict[str, Any]] = Non e)	response = { "user_id": user_id, "session_id": session_id, "task_id": task_id }	LongMemRequest	客户端发起的写入长期记忆的请求数据	LongMemRequest(BaseModel): # 用户唯一标识 user_id: str # 写入的内容 memory_info: str
/system/info	GET	获取当前系统内全部的会话状态信息	-	response = SystemInfoResponse(sessions_count= sessions_count, active_users= active_users)	AgentResponse	运行智能体后返回的响应数据	class AgentResponse(BaseModel): # 会话唯一标识 session_id: str # 任务唯一标识 task_id: str # 三个状态: interrupted, completed, error status: str # 时间戳 timestamp: float = Field(default_factory=lambda: time.time()) # error时的提示消息 message: Optional[str] = None # completed时的结果消息 result: Optional[Dict[str, Any]] = None # interrupted时的中断消息 interrupt_data: Optional[Dict[str, Any]] = None
/agent/active/sessionid/{user_id}	GET	获取指定用户ID的最近一次修改的会话ID	user_id	response = ActiveSessionInfoResponse(active_session_id=active_session_id)			class InterruptResponse(BaseModel): # 用户唯一标识 user_id: str # 会话唯一标识 session_id: str # 任务唯一标识 task_id: str # 中断响应类型 response_type: str # 如果是edit, response类型, 可能需要额外的参数 args: Optional[Dict[str, Any]] = None
/agent/sessionids/{user_id}	GET	获取指定用户ID的所有session_id	user_id	response = SessionInfoResponse(session_ids=session_ids)			class SystemInfoResponse(BaseModel): # 当前系统内会话总数 sessions_count: int # 系统内当前活跃的用户和会话 active_users: List[str]
/agent/tasks/{user_id}/{session_id}	GET	获取指定用户ID和会话ID下所有任务的ID和状态值	user_id, session_id	response = TaskInfoResponse(task_ids=task_ids)	SessionInfoResponse	系统内的会话状态响应数据	class SessionInfoResponse(BaseModel): # 当前用户的所有的session_id session_ids: List[str]
/agent/status/{user_id}/{session_id}/{task_id}	GET	获取指定用户当前会话和任务ID的对应的状态数据	user_id, session_id, task_id	response = SessionStatusResponse(user_id=user_id, session_id=session_id, task_id=task_id, status=status, last_query=last_query, last_updated=last_updated, last_response=last_response)	TaskInfoResponse	所有会话ID响应数据	class TaskInfoResponse(BaseModel): # 当前用户会话的所有task_ids task_ids: List[str]
/agent/write/longterm	POST	写入指定用户的长期记忆	request=LongMemRequest(user_id: str memory_info: str)	response = { "status": "success", "memory_id": result.get("memory_id"), "message": result.get("message", "记忆存储成功") }	ActiveSessionInfoResponse	当前最近一次更新的会话ID响应	class ActiveSessionInfoResponse(BaseModel): # 最近一次更新的会话ID active_session_id: str
/agent/session/{user_id}/{session_id}	DELETE	删除指定用户指定会话	user_id, session_id	response = { "status": "success", "message": f"用户 {user_id}:{session_id} 的会话已删除" }	SessionStatusResponse	会话状态详情响应数据	class SessionStatusResponse(BaseModel): # 用户唯一标识 user_id: str # 会话唯一标识 session_id: Optional[str] = None # 任务唯一标识 task_id: str # 状态: not_found, idle, running, interrupted, completed, error status: str # error时的提示消息 message: Optional[str] = None # 上次查询 last_query: Optional[str] = None # 上次更新时间 last_updated: Optional[float] = None # 上次响应 last_response: Optional[AgentResponse] = None
/agent/session/{user_id}/{session_id}/{task_id}	DELETE	删除指定用户指定会话的指定任务	user_id, session_id, task_id	response = { "status": "success", "message": f"用户 {user_id}:{session_id}:{task_id} 的任务已删除" }			