

文档规范
1、文档使用 OGNL 规范描述参数结构 2、字段采用unit2.0对话API标准，见UNIBOT2.0对话接口 http://ai.baidu.com/docs#/UNIT-v2-API/top 3所有参数都使用 UTF8 编码

输入参数说明

参数名称	类型	说明	举例	由开发者填充	是否允许 置空
version	string	由 unibot 定义。版本号，每次参数调整都应该有一个新的版本号。	3333	√	否
bot_id	string	(场景id) chatbot的id，唯一标识，固定为5	5	√	否
log_id	string	请求的唯一标识，由调用方定义，unibot 不做唯一性验证。	7758521	√	否
bot_session	string(json)	(由 unibot 定义，bot 的 session 信息，client 直接传递不需要了解其内容。) 对于 chatbot 需要记录历史信息放入 session 中，目前 (2018-4-26) 只存在单轮，所以只需要记录单轮的交互就好。	""	1)由开发者填充 2) 由API根据session_id 填充优先实现1。	(是，置 空时表示 清空
bot_session.session_id	string	(session_id，将来可能需要通过bot的日志，统计我们处理了多少个“会话”。) session_id，初次传入chatbot中session为空，需要产生唯一的session_id。	见bot_session		
bot_session.interactions	list<object>	(历史交互序列，即历史 request/response 序列，序列的每一个元素称作一次交互 (interaction)，随交互进行而交替插入，格式与上述不断增长直到发生清空操作。) 目前 (同上) 只包含单轮操作，所以这里只会存在一个interaction。	见bot_session		
bot_session.interactions[i].interaction_id	string	(第1次交互的唯一标识。)	见bot_session		
bot_session.interactions[i].timestamp	string	根据unibase算法产生唯一标识	见bot_session		
bot_session.interactions[i].request	object	时间戳	见bot_session		
bot_session.interactions[i].response	object	第1次交互的 request，结构参考【输入接口】中的request	见bot_session		
request	object	第1次交互的 response，结构参考【输出接口】中的response	见bot_session		
request	object	本轮请求体。		√	否
request.query	string	(本轮请求query: query_info.type为TEXT时，为常规的文本型query (query_info.type为EVENT时，为一组K-V (json)，且其中必须包含一个名为 [event_name] 的key，其他自便。)	见request	√	
request.query_info	object	chatbot中只会为TEXT	见request	√	否
request.query_info.type	string	本轮请求query的附加信息。	见request	√	否
request.query_info.type	enum	TEXT chatbot中只会为TEXT	见request	√	否

输出参数说明

参数名称	类型	说明，“”标识说明信息与输入接口相同	默认值
version	string	-	
bot_id	string	-	
log_id	string	-	
bot_session	string (json)	与【输入接口】中的bot_session相同，这里不展开	
interaction_id	string	为本轮请求+应答之组合，生成的id	
response	object	本轮应答体	
response.status	int	状态码	
response.msg	string	错误信息	"ok"
response.qu_res	object	置空	
response.action_list	list<object>	置空	
response.action_list[i].confidence	float	置信度 (match程度)	
response.action_list[i].action_id	string	目前用solr数据库时的qid填充	
response.action_list[i].custom_reply	string	置空	
response.action_list[i].refine_detail	string	置空	
response.action_list[i].say	string	应答话术，chatbot里为回复	
response.action_list[i].type	string	chat - 聊天话术 chatbot中为这个	"chat"
response.schema	object	默认置空	{}
response.schema.intent	string	意图名称，现在默认置空	BUILD_CHAT (闲聊)
response.schema.intent_confidence	float	置信度	1

请求方式

框架指定了 http 的通信协议，用户可以通过 http post 的方式进行请求，示例如下： curl -d request_data http://ip:port/anyq，其中request_data是把输入接口序列化为字符串，参考如下： curl -d '{"log_id": "7758521", "version": "00003333", "request": {"query_info": {"type": "TEXT"}, "query": "今天天气怎么样"}, "bot_id": "5", "bot_session":""}' 10.14.25.15:8991/anyq
