

自然语言处理

# API 参考

文档版本 01  
发布日期 2024-04-24



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 安全声明

## 漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

# 目录

<b>1 使用前必读</b>	<b>1</b>
1.1 概述	1
1.2 调用说明	1
1.3 终端节点	1
1.4 约束与限制	1
1.5 基本概念	2
<b>2 API 概览</b>	<b>3</b>
<b>3 如何调用 API</b>	<b>4</b>
3.1 申请服务	4
3.2 构造请求	5
3.3 认证鉴权	7
3.4 返回结果	9
<b>4 API</b>	<b>11</b>
4.1 自然语言处理基础服务接口说明	11
4.1.1 分词	11
4.1.2 多粒度分词	20
4.1.3 依存句法分析	26
4.1.4 命名实体识别（基础版）	31
4.1.5 命名实体识别（领域版）	35
4.1.6 文本相似度（基础版）	41
4.1.7 文本相似度（高级版）	44
4.1.8 句向量	47
4.1.9 实体链接	50
4.1.10 关键词抽取	54
4.1.11 事件抽取	57
4.1.12 成分句法分析	62
4.2 语言生成服务接口说明	66
4.2.1 文本摘要（基础版）	66
4.2.2 文本摘要（领域版）	71
4.2.3 诗歌生成	75
4.3 语言理解服务接口说明	78
4.3.1 情感分析（基础版）	78

4.3.2 情感分析（领域版）.....	82
4.3.3 文本分类.....	85
4.3.4 属性级情感分析（领域版）.....	89
4.3.5 属性级情感分析（高级版）.....	93
4.3.6 实体级情感分析.....	98
4.3.7 意图理解.....	101
4.3.8 文档分类.....	106
4.4 机器翻译服务接口说明.....	110
4.4.1 文本翻译.....	110
4.4.2 语种识别.....	114
4.4.3 文档翻译任务创建.....	119
4.4.4 文档翻译状态查询.....	123
<b>5 数据结构.....</b>	<b>126</b>
5.1 公共请求参数.....	126
5.2 公共响应参数.....	126
<b>6 附录.....</b>	<b>128</b>
6.1 状态码.....	128
6.2 错误码.....	128
6.3 获取项目 ID.....	132
6.4 获取账号 ID.....	133
6.5 配置 OBS 访问权限.....	133
<b>7 修订记录.....</b>	<b>135</b>

# 1 使用前必读

## 1.1 概述

欢迎使用自然语言处理（Natural Language Processing），服务主要包括自然语言处理基础服务、语言生成服务、语言理解服务、机器翻译服务等功能。

在调用自然语言处理服务API之前，请确保已经充分了解自然语言处理服务相关概念，详细信息请参见[产品介绍](#)。

## 1.2 调用说明

自然语言处理服务提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

## 1.3 终端节点

终端节点即调用API的[请求地址](#)，不同服务不同区域的终端节点不同，您可以从[地区和终端节点](#)中查询所有服务的终端节点。

目前自然语言处理基础、语言生成、语言理解和机器翻译服务支持以下地区和终端节点：

表 1-1 自然语言处理基础、语言生成、语言理解和机器翻译服务

区域名称	区域	终端节点（Endpoint）	协议类型
华北-北京四	cn-north-4	nlp-ext.cn-north-4.myhuaweicloud.com	HTTPS

## 1.4 约束与限制

更详细的限制请参见具体API的说明和[产品介绍约束与限制](#)。

## 1.5 基本概念

- 账号  
用户注册华为云时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用创建的用户进行日常管理工作。
- 用户  
由账号创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。在[我的凭证](#)下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。
- 区域（Region）  
从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。  
详情请参见[区域和可用区](#)。
- 可用区（AZ，Availability Zone）  
一个AZ是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。
- 项目  
华为云的区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中购买资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中资源，使得资源的权限控制更加精确。

图 1-1 项目隔离模型



# 2 API 概览

自然语言处理提供了如下接口，方便用户对自然语言处理的使用。各类接口的说明如表2-1所示。

表 2-1 接口说明

接口类型	说明
自然语言处理基础接口	包含分词、多粒度分词、依存句法分析、命名实体识别、文本相似度和句向量、实体链接、关键词抽取、事件抽取接口等。
语言生成接口	包含文本摘要、诗歌生成接口。
语言理解接口	包含情感分析、文本分类、意图理解接口。
机器翻译接口	包含文本翻译、语种识别接口。

# 3 如何调用 API

## 3.1 申请服务

### 前提条件

已注册华为账号并开通华为云，完成实名认证，且账号不能处于欠费、冻结、被注销等异常状态。

### 操作步骤

1. 进入[自然语言处理NLP主页](#)，单击“立即使用”。  
进入自然语言处理控制台。
2. 在自然语言处理控制台“总览”页面，选择需要使用的服务，在操作列单击“开通服务”。
3. 在弹出的对话框中单击“确定”。  
服务开通成功后，开通状态显示“已开通”。

图 3-1 开通服务



### 📖 说明

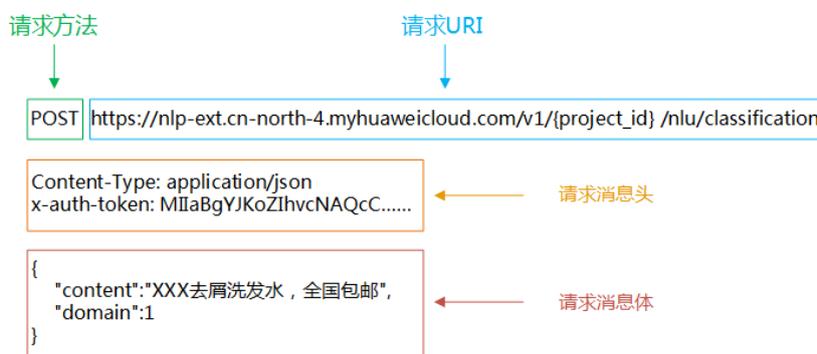
- NLP服务开通后，暂不支持关闭。开通服务时，计费规则默认为“按需计费”，按需计费时，不使用NLP服务，则不收费。如果您购买了套餐包，套餐包扣减规则请参见[价格计算器](#)计费详情页。
- 如未开通服务，直接调用NLP API会提示ModelArts.4204报错。

## 3.2 构造请求

本节介绍REST API请求的组成，讲述如何调用API接口。

请求示例如下所示，一个请求主要由请求URL、请求方法、请求消息头和请求消息体几部分组成，各个部分将在下文详细解释。

图 3-2 请求示例图



### 请求 URI

请求URI由如下部分组成。

**{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}**

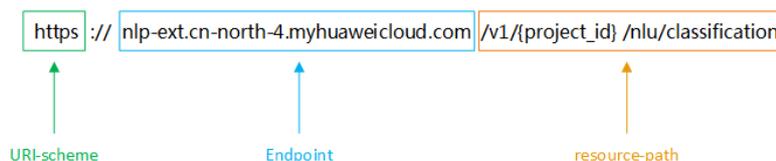
尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

- **URI-scheme**: 表示用于传输请求的协议，当前所有API均采用HTTPS协议。
- **Endpoint**: 指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从[地区和终端节点](#)中获取。例如自然语言处理（NLP）服务在“华北-北京四”区域的Endpoint为“nlp-ext.cn-north-4.myhuaweicloud.com”。
- **resource-path**: 资源路径，也即API访问路径。从具体API的URI模块获取，例如“文本分类”API的resource-path为“/v1/{project\_id}/nlu/classification”。其中{project\_id}为项目ID，请参考[获取项目ID](#)获取。
- **query-string**: 查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。当前NLP服务未使用该参数。

例如，您需要在“华北-北京四”区域调用文本分类API，则需使用“华北-北京四”区域的Endpoint（nlp-ext.cn-north-4.myhuaweicloud.com），并在[文本分类](#)的URI部分找到resource-path（/v1/{project\_id}/nlu/classification），拼接起来如下所示。

```
https://nlp-ext.cn-north-4.myhuaweicloud.com/v1/{project_id}/nlu/classification
```

图 3-3 URI 示意图



### 说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，而Endpoint在同一个区域也相同，所以简洁起见将这两部分省略。

## 请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

- **GET**：请求服务器返回指定资源。
- **PUT**：请求服务器更新指定资源。
- **POST**：请求服务器新增资源或执行特殊操作。
- **DELETE**：请求服务器删除指定资源，如删除对象等。
- **HEAD**：请求服务器资源头部。
- **PATCH**：请求服务器更新资源的部分内容。当资源不存在的时候，PATCH可能会去创建一个新的资源。

在[文本分类](#)的URI部分，您可以看到其请求方法为“POST”，则其请求为：

```
POST https://nlp-ext.cn-north-4.myhuaweicloud.com/v1/{project_id}/nlu/classification
```

## 请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

如下公共消息头需要添加到请求中。

- **Content-Type**：消息体的类型（格式），必选，默认取值为“application/json”，有其他取值时会在具体接口中专门说明。
- **X-Auth-Token**：用户Token，可选，当使用Token方式认证时，必须填充该字段。用户Token请参考[3.2 认证鉴权](#)。用户Token也就是调用IAM服务获取用户Token接口的响应值，该接口是唯一不需要认证的接口。

### 说明

API同时支持使用AK/SK认证，AK/SK认证是使用SDK对请求进行签名，签名过程会自动往请求中添加Authorization（签名认证信息）和X-Sdk-Date（请求发送的时间）请求头。AK/SK认证的详细说明请参见[AK/SK认证](#)。

对于获取用户Token接口，添加消息头后的请求如下所示。

```
POST https://nlp-ext.cn-north-4.myhuaweicloud.com/v1/{project_id}/nlu/classification
Content-Type: application/json
x-auth-token: MIlaBgYJKoZlhvcNAQcC.....
```

## 请求消息体

请求消息体通常以结构化格式发出，与请求消息头中Content-type对应，传递除请求消息头之外的内容。如果请求消息体中参数支持中文，则中文字符必须为UTF-8编码。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

对于[文本分类](#)接口，您可以从接口的请求部分看到所需的请求参数及参数说明。将消息体加入后的请求如下所示。

```
POST https://nlp-ext.cn-north-4.myhuaweicloud.com/v1/{project_id}/nlu/classification
Content-Type: application/json
x-auth-token: M1laBgYJKoZlhvcNAQcC.....
{
  "content": "XXX去屑洗发水，全国包邮",
  "domain": 1
}
```

到这里为止这个请求需要的内容就具备齐全了，您可以使用[curl](#)、[Postman](#)或直接编写代码等方式发送请求调用API。对于[文本分类](#)接口，您可以从响应消息部分看到返回参数及参数说明。

## 3.3 认证鉴权

调用NLP服务接口目前支持Token认证进行认证鉴权。

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- **Token认证：**通过Token认证调用请求。  
通过[终端节点](#)和Token调用API。不支持调用“中东-阿布扎比-OP5” Region的对象存储服务（OBS）、数据接入服务（DIS）、自然语言处理（NLP）、文字识别（OCR）和图像识别（Image Recognition）。
- **AK/SK认证：**通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。

### Token 认证

Token在计算机系统中代表令牌（临时）的意思，拥有Token就代表拥有某种权限。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限。

#### 说明

- Token的有效期为24小时，需要使用一个Token鉴权时，可以先缓存起来，避免频繁调用。
- 如果您的华为云账号已升级为华为账号，将不支持获取账号Token，建议您为自己创建一个IAM用户，授予该用户必要的权限，获取IAM用户Token。
- 如果您是第三方系统用户，直接使用联邦认证的用户名和密码获取Token，系统会提示密码错误。请先在华为云的登录页面，通过“忘记密码”功能，设置原华为云账号密码。
- 如果您开启了登录保护并设置登录保护为MFA验证，请参考[获取IAM用户Token（使用密码+虚拟MFA）](#)获取IAM用户Token。
- 如果需要获取具有Security Administrator权限的Token，请参见：[如何获取Security Administrator权限的Token](#)。
- 通过Postman获取用户Token示例请参见：[如何通过Postman获取用户Token](#)。

NLP服务所需Token获取可参考下文，如需了解更多获取Token方式，请参考[获取用户Token](#)。

获取用户Token接口请求构造如下，您可以从接口的请求部分看到所需的请求参数及参数说明。获取Token消息头只需填写“Content-Type”，将消息体加入后的请求如下所示。

加粗的斜体字段需要根据实际值填写，其中`username`为用户名，`domainname`为用户所属的账号名称，`*****`为用户登录密码。详细信息请参见[Token消息体中username, domain name和project name分别指的是什么?](#) 章节。

获取Token的终端节点和`projectname`需与NLP服务终端节点保持一致。当访问华北-北京四的终端节点（即`nlp-ext.cn-north-4.myhuaweicloud.com`时），获取Token请使用终端节点`https://iam.cn-north-4.myhuaweicloud.com`，`projectname`对应使用`cn-north-4`。

Token可通过调用[获取用户Token](#)接口获取，调用本服务API需要project级别的Token，即调用[获取用户Token](#)接口时，请求body中`auth.scope`的取值需要选择project，如下所示。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "projectname"
      }
    }
  }
}
```

如下图所示，返回的响应消息头中“x-subject-token”就是需要获取的用户Token。获取Token之后，您就可以使用Token认证调用NLP服务API。



## 响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”，“x-request-id”，NLP服务响应消息头无特殊用途，可用于定位问题使用。

## 响应消息体

响应消息体通常以结构化格式返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

对于[文本分类](#)接口，返回如下消息体，格式请具体参考[文本分类](#)响应消息部分。

```
{
  "result": {
    "content": "XXX去屑洗发水，全国包邮",
    "label": 0,
    "confidence": 0.5190434
  }
}
```

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

```
{
  "error_code": "NLP.0301",
  "error_msg": "query param error content.at least one of Chinese, English, or number;"
}
```

其中，error\_code表示错误码，error\_msg表示错误描述信息。

# 4 API

## 4.1 自然语言处理基础服务接口说明

### 4.1.1 分词

#### 功能介绍

对文本进行分词和词性标注处理。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[基础套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

#### 调试

您可以在[API Explorer](#)中调试该接口。

#### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

#### URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/segment
- 参数说明

表 4-1 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-2](#)所示。

表 4-2 请求 Body 参数

参数名	参数类型	必选	说明
text	String	是	待分词文本，中文长度为1~512，英文长度为1~2000，文本编码为UTF-8。
pos_switch	Integer	否	是否开启词性标注功能，1为开启，0为关闭，默认为关闭。
lang	String	否	支持的文本语言类型，目前支持中文（zh）和英文（en），默认为中文。
criterion	String	否	支持的分词规范。 中文分词标准目前支持PKU（北大分词标准）、CTB（宾州中文树库标准），默认为PKU。 英文分词标准默认为Penn TreeBank（宾州树库标准），不需要传入该参数。

## 响应消息

响应参数如[表4-3](#)所示。

表 4-3 响应参数

参数名	参数类型	说明
words	Array of words	分词结果，请参见 <a href="#">表4-4</a> 。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-4 Word 字段数据结构说明

参数名	参数类型	说明
content	String	词汇文本。
pos	String	词汇对应的词性。详细说明请参见表4-5、表4-6、表 (Penn TreeBank) 词性说明。

表 4-5 PKU 词性说明

一类词性	二类词性	三类词性
n: 名词	nr: 人名	<ul style="list-style-type: none"> <li>• nr1: 汉语姓氏</li> <li>• nr2: 汉语名字</li> <li>• nrj: 日语人名</li> <li>• nrf: 音译人名</li> </ul>
	ns: 地名	nsf: 音译地名
	nt: 机构团体名	-
	nz: 其它专名	-
	nl: 名词性惯用语	-
	ng: 名词性语素	-
t: 时间词	tg: 时间词性语素	-
s: 处所词	-	-
f: 方位词	-	-
v: 动词	vd: 副动词	-
	vn: 名动词	-
	vshi: 动词“是”	-
	vyou: 动词“有”	-
	vf: 趋向动词	-
	vx: 形式动词	-
	vi: 不及物动词(内动词)	-
	vl: 动词性惯用语	-
	vg: 动词性语素	-
a: 形容词	ad: 副形容词	-
	an: 名形容词	-

一类词性	二类词性	三类词性
	ag: 形容词性语素	-
	al: 形容词性惯用语	-
b: 区别词	bl: 区别词性惯用语	-
z: 状态词	-	-
r: 代词	rr: 人称代词	-
	rz: 指示代词	<ul style="list-style-type: none"> <li>● rzt: 时间指示代词</li> <li>● rzs: 处所指示代词</li> <li>● rzv: 谓词性指示代词</li> </ul>
	ry: 疑问代词	<ul style="list-style-type: none"> <li>● ryt: 时间疑问代词</li> <li>● rys: 处所疑问代词</li> <li>● ryv: 谓词性疑问代词</li> </ul>
	rg: 代词性语素	-
m: 数词	mq: 数量词	-
	mg: 甲、乙、丙、丁、戊、己、庚、辛、壬、癸	-
q: 量词	qv: 动量词	-
	qt: 时量词	-
d: 副词	-	-
p: 介词	pba: 介词“把”	-
	pbei: 介词“被”	-
c: 连词	cc: 并列连词	-
u: 助词	uzhe: 着	-
	ule: 了、喽	-
	uguo: 过	-
	ude1: 的、底	-
	ude2: 地	-
	ude3: 得	-
	usuo: 所	-
	udeng: 等、等等、云云	-
uyy: 一样、一般似的、般	-	

一类词性	二类词性	三类词性
	udh: 的话	-
	uls: 来讲、来说而言、说来	-
	uzhi: 之	-
	ulian: 连 (“连小学生都会”)	-
e: 叹词	-	-
y: 语气词(delete yg)	-	-
o: 拟声词	-	-
h: 前缀	-	-
k: 后缀	-	-
x: 字符串	xe: Email字符串	-
	xs: 微博会话分隔符	-
	xm: 表情符号	-
	xu: 网址URL	-
w: 标点符号	wkz: 左括号, 全角: ( [ [ { 《 【 【 ‹ 半角: 角: ( [ { <	-
	wky: 右括号, 全角: 角: ) ] ] } 》 ] ] ) 半角: 角: ) ] { >	-
	wyz: 左引号, 全角: “ ‘ 『	-
	wyy: 右引号, 全角: ” , ’ 』	-
	wj: 句号, 全角: 。	-
	ww: 问号, 全角: ? 半角: 角: ?	-
	wt: 叹号, 全角: ! 半角: 角: !	-
	wd: 逗号, 全角: , 半角: 角: ,	-
	wf: 分号, 全角: ; 半角: 角: ;	-
	wn: 顿号, 全角: 、	-

一类词性	二类词性	三类词性
	wm: 冒号, 全角: : 半角: :	-
	ws: 省略号, 全角: …… …	-
	wp: 破折号, 全角: —— - - —— - 半角: --- ----	-
	wb: 百分号千分号, 全角: % ‰ 半角: %	-
	wh: 单位符号, 全角: ¥ \$ > °C 半角: \$	-

表 4-6 CTB 词性说明

词性	名称	示例
AD	副词	不也就
AS	动态助词	了著过
BA	把字结构	将把
CC	并列连接词	和与
CD	限定数量词	一两三
CS	从属连接词	虽然 如果 如果
DEC	补语或名词化	的之
DEG	关联或所有格	的之
DER	补语短语“得”	得
DEV	方式“地”	地
DT	限定词	这各全
ETC	等等	等 等等
FW	外来词	A E B
IJ	感叹词	唉呀 哈拉
JJ	名词修饰词	大新小
LB	长“被”结构	被为受
LC	方位词	中上时
M	量词	个年美元

词性	名称	示例
MSP	其他助词	所 而 来
NN	名词	经济 企业 人
NR	专有名词	有限 公司
NT	时间名词	目前 去年
OD	数词	第一 第二 首
ON	拟声词	〇
P	介词	在 对 以
PN	代词	他 我 自己
PU	标点符号	， 。
SB	短“被”结构	被 遭
SP	句末助词	了 的 吗
VA	谓词性形容词	大多 好
VC	系动词	是 为 非
VE	主要动词“有”	有 没有 无
VV	动词	说 要 会

表 4-7 Penn TreeBank 词性说明

词性	名称	示例
CC	并列连接词	and, but, or
CD	基数	one, two
DT	限定词	a, the
EX	存在型there	there
FW	外文单词	mea, culpa
IN	介词/从属, 连接词	of, in, by
JJ	形容词	yellow
JJR	形容词, 比较级	bigger
JJS	形容词, 最高级	wildest
LS	列表项标记	1, 2, One
MD	情态动词	can, could, might

词性	名称	示例
NN	名词, 可数或不可数	llama
NNS	名词, 复数	llamas
NNP	专有名词, 单数	IBM
NNPS	专有名词, 复数	Carolinas
PDT	前位限定词	all, both
POS	所有格结束词	's
PRP	人称代名词	I, me, you,
PRP\$	物主代词, 所有格代名词	my, your, yours
RB	副词	quickly
RBR	副词, 比较级	faster
RBS	副词, 最高级	fastest
RP	小品词	up, off
SYM	符号 (数学或科学)	+, %, &
TO	to	to
UH	感叹词	ah, oops
VB	动词, 基本形态	eat
VBD	动词, 过去式	ate
VBG	动词, 动名词/现在分词	eating
VBN	动词, 过去分词	eaten
VBP	动词, 非第三人称单数现在式	eat
VBZ	动词, 第三人称单数现在式	eats
WDT	wh-限定词	which, that
WP	wh-代词	what, who
WP\$	所有格wh-代词	whose
WRB	wh-副词	how, where
PU	标点符号	, . :

## 请求示例

- 请求示例 (开启词性标注功能进行分词, 支持PKU分词规范)

 说明

“endpoint”、“project\_id”、“token”等请求参数获取方式可参考[快速入门](#)，参数详情请见[构造请求](#)。

```
Request Header:  
Content-Type: application/json  
X-Auth-Token:  
MIINRwYJKoZlHvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
```

```
Request Body:  
{  
  "text": "今天天气真好",  
  "pos_switch": 1,  
  "lang": "zh",  
  "criterion": "PKU"  
}
```

- Python3语言请求代码示例（开启词性标注功能进行分词，支持PKU分词规范）

```
# -*- coding: utf-8 -*-  
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests  
import requests  
import json  
  
def nlp_demo():  
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/segment' # endpoint和project_id需替换  
    token = '用户对应region的token'  
    header = {  
        'Content-Type': 'application/json',  
        'X-Auth-Token': token  
    }  
    body = {  
        'text': '今天天气真好',  
        'pos_switch': 1,  
        'lang': 'zh',  
        'criterion': 'PKU'  
    }  
    resp = requests.post(url, data=json.dumps(body), headers=header)  
    print(resp.json())  
  
if __name__ == '__main__':  
    nlp_demo()
```

- Java语言请求代码示例（开启词性标注功能进行分词，支持PKU分词规范）

```
import java.io.BufferedReader;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStreamWriter;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
/**  
 * 此demo仅供测试使用，建议使用sdk  
 */  
public class NLPDemo {  
    public void nlpDemo() {  
        try {  
            //endpoint和projectId需要替换成实际信息。  
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/segment");  
            String token = "对应region的token";  
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
            connection.setRequestMethod("POST");  
            connection.setDoInput(true);  
            connection.setDoOutput(true);  
            connection.addRequestProperty("Content-Type", "application/json");  
            connection.addRequestProperty("X-Auth-Token", token);  
  
            //输入参数  
            String text = "订单记录怎么删除";
```

```
String body = "{\"text\":\"" + text + "\",\"pos_switch\":1,\"lang\":\"zh\",\"criterion\":\"PKU\"}";

OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
osw.append(body);
osw.flush();
InputStream is = connection.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
while (br.ready()) {
    System.out.println(br.readLine());
}
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "words": [
    {
      "content": "今天",
      "pos": "t"
    },
    {
      "content": "天气",
      "pos": "n"
    },
    {
      "content": "真",
      "pos": "d"
    },
    {
      "content": "好",
      "pos": "a"
    }
  ]
}
```

- 失败响应示例

```
{
  "error_code": "NLP.0301",
  "error_msg": "The length of text should be in the range of 1-512"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

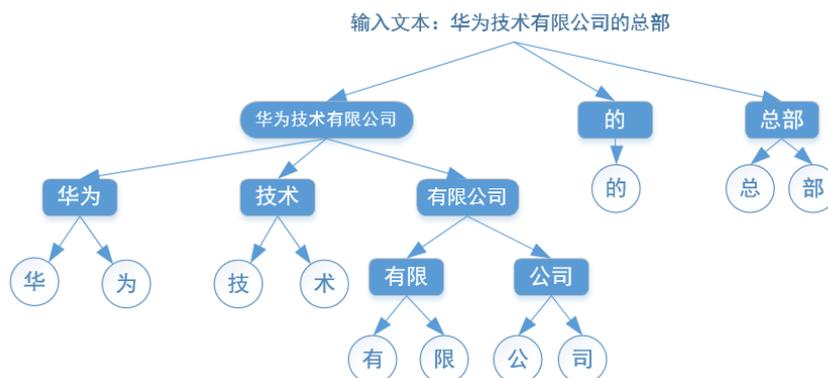
## 4.1.2 多粒度分词

### 功能介绍

给定一个句子输入，输出不同粒度的所有单词的层次结构。

以“华为技术有限公司的总部”为例，多粒度分词得到的层次结构如下图所示。其中白色圆形节点为字符单元，蓝色圆角矩形节点为词汇单元。

图 4-1 多粒度分词



本API免费调用，调用限制为2次/秒。

## 调试

您可以在[API Explorer](#)中调试该接口。

## 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/multi-grained-segment
- 参数说明

表 4-8 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如表 [请求参数](#) 所示。

表 4-9 请求参数

参数名	参数类型	必选	说明
text	String	是	待分析文本，长度为1~64，文本编码为UTF-8。
lang	String	否	支持的文本语言类型，目前支持中文（zh）和英文（en），默认为中文。
granularity	Integer	否	分词粒度，1为最细粒度，2为最粗粒度，其他情况默认返回全部粒度分词树结果。

## 响应消息

响应参数如表4-10所示。

表 4-10 响应参数

参数名	参数类型	说明
result	Array of <b>node objects</b> 或 Array of strings	分词结果。默认返回全部粒度分词树结果，如果选择了分词粒度，则返回对应粒度的词汇列表结果。
error_code	String	调用失败时的错误码。具体参见 <b>错误码</b> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-11 node 字段数据结构说明

参数名	参数类型	说明
content	String	该节点对应的文本内容，并基于文本的unicode编码，做归一化处理。 例如：中文标点“，”会映射到英文标点“,”。
type	String	节点类型，包括 WORD-词汇类型，CHAR-字符类型。
sub_contents	Array of node objects	子节点列表。

## 请求示例

- 请求示例1（多粒度分词，分词粒度为最粗粒度）

```
POST https://{endpoint}/v1/{project_id}/nlp-fundamental/multi-grained-segment
```

Request Header:

```
Content-Type: application/json
```

```
X-Auth-Token:
```

```
MIINRwYJKoZIHvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...
```

Request Body:

```
{
  "text": "华为技术有限公司的总部",
  "lang": "zh",
  "granularity": 2
}
```

- Python3语言请求代码示例（多粒度分词，分词粒度为最细粒度）

```
# -*- coding: utf-8 -*-
```

```
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
```

```
import json
```

```
def nlp_demo():
```

```
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/multi-grained-segment' # endpoint和  
    project_id需替换
```

```
    token = '用户对应region的token'
```

```
    header = {
```

```
        'Content-Type': 'application/json',
```

```
        'X-Auth-Token': token
```

```
    }
```

```
    body = {
```

```
        'text': '华为技术有限公司的总部',
```

```
        'granularity': 1
```

```
    }
```

```
    resp = requests.post(url, data=json.dumps(body), headers=header)
```

```
    print(resp.json())
```

```
if __name__ == '__main__':
```

```
    nlp_demo()
```

- Java语言请求代码示例（多粒度分词，分词粒度为最细粒度）

```
import java.io.BufferedReader;
```

```
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
```

```
import java.io.OutputStreamWriter;
```

```
import java.net.HttpURLConnection;
```

```
import java.net.URL;
```

```
/**
```

```
 * 此demo仅供测试使用，建议使用sdk
```

```
 */
```

```
public class NLPDemo {
```

```
    public void nlpDemo() {
```

```
        try {
```

```
            //endpoint和projectId需要替换成实际信息。
```

```
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/multi-grained-  
segment");
```

```
            String token = "对应region的token";
```

```
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
```

```
            connection.setRequestMethod("POST");
```

```
            connection.setDoInput(true);
```

```
            connection.setDoOutput(true);
```

```
            connection.addRequestProperty("Content-Type", "application/json");
```

```
            connection.addRequestProperty("X-Auth-Token", token);
```

```
            //输入参数
```

```
            String text = "华为技术有限公司的总部";
```

```
            String body = "{\"text\":\"" + text + "\",\"granularity\":1}";
```

```

        OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
        osw.append(body);
        osw.flush();
        InputStream is = connection.getInputStream();
        BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
        while (br.ready()) {
            System.out.println(br.readLine());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}

```

- 请求示例2（多粒度分词）

POST [https://{endpoint}/v1/{project\\_id}/nlp-fundamental/multi-grained-segment](https://{endpoint}/v1/{project_id}/nlp-fundamental/multi-grained-segment)

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZlHvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBGkqhkiG...

Request Body:

```

{
  "text": "华为技术有限公司的总部",
  "lang": "zh"
}

```

 说明

“endpoint”、“project\_id”、“token”等请求参数获取方式可参考[快速入门](#)，参数详情请见[构造请求](#)。

## 响应示例

- 成功响应示例1

```

{
  "result": [
    "华为技术有限公司",
    "的",
    "总部"
  ]
}

```

- 成功响应示例2

```

{
  "result": [
    {
      "content": "华为技术有限公司",
      "sub_contents": [
        {
          "content": "华为",
          "sub_contents": [
            {
              "content": "华",
              "type": "CHAR"
            },
            {
              "content": "为",
              "type": "CHAR"
            }
          ]
        }
      ]
    }
  ],
  "type": "WORD"
}

```

```
},
{
  "content": "技术",
  "sub_contents": [
    {
      "content": "技",
      "type": "CHAR"
    },
    {
      "content": "术",
      "type": "CHAR"
    }
  ],
  "type": "WORD"
},
{
  "content": "有限公司",
  "sub_contents": [
    {
      "content": "有限",
      "sub_contents": [
        {
          "content": "有",
          "type": "CHAR"
        },
        {
          "content": "限",
          "type": "CHAR"
        }
      ],
      "type": "WORD"
    },
    {
      "content": "公司",
      "sub_contents": [
        {
          "content": "公",
          "type": "CHAR"
        },
        {
          "content": "司",
          "type": "CHAR"
        }
      ],
      "type": "WORD"
    }
  ],
  "type": "WORD"
},
{
  "content": "的",
  "sub_contents": [
    {
      "content": "的",
      "type": "CHAR"
    }
  ],
  "type": "WORD"
},
{
  "content": "总部",
  "sub_contents": [
    {
      "content": "总",
      "type": "CHAR"
    }
  ],

```

```
{
  "content": "部",
  "type": "CHAR"
},
{
  "type": "WORD"
}
]
```

- 失败响应示例

```
{
  "error_code": "NLP.0301",
  "error_msg": "the length of the text must between 1-64"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.1.3 依存句法分析

### 功能介绍

识别句子中词汇与词汇之间的相互依存关系。

本API免费调用，调用限制为2次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/dependency-parser
- 参数说明

表 4-12 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如表 [请求参数](#) 所示。

表 4-13 请求参数

参数名	参数类型	必选	说明
text	String	是	待分析文本，长度为1~32，文本编码为UTF-8。
lang	String	否	支持的文本语言类型，目前只支持中文（zh），默认为中文。

## 响应消息

响应参数如表 [响应参数](#) 所示。

表 4-14 响应参数

参数名	参数类型	说明
words	Array of words objects	依存句法分析结果。 请参见表 <a href="#">words数据结构说明</a> 。
error_code	String	调用失败时的错误码，具体参见 <a href="#">6.2 错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-15 words 数据结构说明

参数名	参数类型	说明
id	Integer	词汇id，从1开始计数。
word	String	词汇内容。
pos	String	词性，请参见 <a href="#">表4-6</a> 。
head_word_id	Integer	头节点id。如果是根节点，默认为0。

参数名	参数类型	说明
dependency_label	String	词与头节点的依存关系，使用汉语开放依存句法树库（CODT）依存关系标签集合。详细说明请参见 <a href="#">表 (CODT)关系类型说明</a> 。

表 4-16 (CODT)关系类型说明

关系标签	说明	例句	标注结果
root	sentence root (根节点)	我 爱 妈妈	(Root ->爱), 其中Root为虚拟词
sasubj-obj	same subject and object (同主语同宾语)	我 一直 在 研究 和 思考 这 个 问 题。	(研究->思考)
sasubj	same subject (同主语)	我 走 进 操 场 打 篮 球。	(走进->打)
dfsubj	different subject (不同主语)	这 书 太 贵 了, 我 打 算 买 另 一 本	(贵->打算)
subj	subject (主语)	我 爱 妈 妈	(我<-爱)
subj-in	subject inside a subject-predicate predicate (主谓谓语句中的内部主语)	他 确 实 头 疼	(头<-疼)
obj	object (宾语)	我 爱 妈 妈	(爱->妈妈)
pred	predicate (谓语)	命 令 他 扫 地	(他->扫地)
att	attribute modifier (定语)	国 家 主 席	(国家<-主席)
adv	adverbial modifier (状语)	非 常 喜 欢	(非常<-喜欢)
cmp	complement modifier (补语)	洗 干 净 手	(洗->干净)
coo	coordination construction (并列结构)	鲜 花 和 掌 声	(鲜花->掌声)
pobj	preposition object (介宾)	在 家 看 书	(在->家)
iobj	indirect-object (间宾)	给 他 书	(给->他)
de	de-construction (“的”字结构)	这 是 他 的	(他<-的)

关系标签	说明	例句	标注结果
adjct	adjunct (附加成分)	我走了	(走->了)
app	appellation (称呼)	老师, 你好	(老师<-好)
exp	explanation (进一步解释)	王刚 (红星小学校长)	(王刚->校长)
punc	punctuation (标点)	我爱妈妈。	(爱->。)
frag	fragment (片段)	你, 我, 中国。	(你->我->中国)
repet	repetition (重复)	你吃, 吃饭了吗?	(吃->吃)

## 请求示例

调用API前, 请先[构造请求](#), 通过[获取用户Token](#)认证鉴权。

- 请求示例 (分析"张三买电脑"的句法依存关系, 支持的语言类型为中文)

POST https://{endpoint}/v1/{project\_id}/nlp-fundamental/dependency-parser

Request Header:

Content-Type: application/json

X-Auth-Token:

MIIFbwYJKoZlhwNAQcCoIIFYDCCBVwCAQExDTALBgIghkgBZQMEAgEwggNBgkqhkiG9...

Request Body: {

"text": "张三买电脑",

"lang": "zh"

}

- Python3语言请求代码示例 (分析"张三买电脑"的句法依存关系, 支持的语言类型为中文)

```
# -*- coding: utf-8 -*-
```

```
# 此demo仅供测试使用, 建议使用sdk。需提前安装requests, 执行pip install requests
```

```
import requests
```

```
import json
```

```
def nlp_demo():
```

```
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/dependency-parser' # endpoint和project_id需替换
```

```
    token = '用户对应region的token'
```

```
    header = {
```

```
        'Content-Type': 'application/json',
```

```
        'X-Auth-Token': token
```

```
    }
```

```
    body = {
```

```
        'text': '张三买电脑',
```

```
        'lang': 'zh'
```

```
    }
```

```
    resp = requests.post(url, data=json.dumps(body), headers=header)
```

```
    print(resp.json())
```

```
if __name__ == '__main__':
```

```
    nlp_demo()
```

- Java语言请求代码示例 (分析"订单记录怎么删除"的句法依存关系, 支持的语言类型为中文)

```
import java.io.BufferedReader;
```

```
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
```

```
import java.io.OutputStreamWriter;
```

```
import java.net.HttpURLConnection;
```

```
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/dependency-parser");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String text = "订单记录怎么删除";
            String body = "{\"text\":\"" + text + "\",\"lang\":\"zh\"}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        NLPDemo nlpDemo = new NLPDemo();
        nlpDemo.nlpDemo();
    }
}
```

## 响应示例

- 成功响应示例

```
{
  "words": [
    {
      "id": 1,
      "word": "张三",
      "pos": "NR",
      "head_word_id": 2,
      "dependency_label": "subj"
    },
    {
      "id": 2,
      "word": "买",
      "pos": "VV",
      "head_word_id": 0,
      "dependency_label": "root"
    },
    {
      "id": 3,
      "word": "电脑",
      "pos": "NN",
      "head_word_id": 2,
      "dependency_label": "obj"
    }
  ]
}
```

```
    ]  
  }  
  • 失败响应示例  
  {  
    "error_code": "NLP0301",  
    "error_message": "Missing parameters:text"  
  }
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.1.4 命名实体识别（基础版）

### 功能介绍

对文本进行命名实体识别分析，目前支持人名、地名、时间、组织机构类实体的识别。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[基础套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/ner
- 参数说明

表 4-17 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-18](#)所示。

表 4-18 请求参数

参数名	参数类型	必选	说明
text	String	是	待分析文本，中文长度为1~512，英文和西班牙文长度为1~2000，文本编码为UTF-8。
lang	String	否	支持的文本语言类型，目前支持中文（zh）、英文（en）、和西班牙文（es），默认为中文。

## 响应消息

响应参数如[表4-19](#)所示。

表 4-19 响应参数

参数名	参数类型	说明
named_entities	Array of named_entity objects	命名实体识别结果。 请参见 <a href="#">表4-20</a> 。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-20 named\_entity 数据结构说明

参数名	参数类型	说明
word	String	实体文本。

参数名	参数类型	说明
tag	String	实体类型，枚举类型。 <ul style="list-style-type: none"> <li>中文支持人名“nr”，地名“ns”，机构名“nt”，时间“t”。</li> <li>英文支持人名“per”，地名“loc”，机构名“org”，时间“t”。</li> <li>西班牙文支持人名“per”，地名“loc”，机构名“org”，时间“t”。</li> </ul>
offset	Integer	实体文本在待分析文本中的起始位置。
len	Integer	实体文本长度。

## 请求示例

- 请求示例（识别分析命名实体）

POST https://{endpoint}/v1/{project\_id}/nlp-fundamental/ner

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgguVBGkqhkiG...

Request Body:

```
{
  "text": "昨天程序员李小明来到北京参加开发者大赛，在比赛中表现优异，赢得了第一名。",
  "lang": "zh"
}
```

- Python3语言请求代码示例（识别分析命名实体）

```
# -*- coding: utf-8 -*-
```

```
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
```

```
import json
```

```
def nlp_demo():
```

```
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/ner' # endpoint和project_id需替换
```

```
    token = '用户对应region的token'
```

```
    header = {
```

```
        'Content-Type': 'application/json',
```

```
        'X-Auth-Token': token
```

```
    }
```

```
    body = {
```

```
        'text': '昨天程序员李小明来到北京参加开发者大赛，在比赛中表现优异，赢得了第一名。',
```

```
        'lang': 'zh'
```

```
    }
```

```
    resp = requests.post(url, data=json.dumps(body), headers=header)
```

```
    print(resp.json())
```

```
if __name__ == '__main__':
```

```
    nlp_demo()
```

- Java语言请求代码示例（识别分析命名实体）

```
import java.io.BufferedReader;
```

```
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
```

```
import java.io.OutputStreamWriter;
```

```
import java.net.HttpURLConnection;
```

```
import java.net.URL;
```

```
/**
```

```
 * 此demo仅供测试使用，建议使用sdk
```

```
*/
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/ner");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String text = "昨天程序员李小明来到北京参加开发者大赛，在比赛中表现优异，赢得了第一名。";
            String body = "{\"text\":\"" + text + "\",\"lang\":\"zh\"}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        NLPDemo nlpDemo = new NLPDemo();
        nlpDemo.nlpDemo();
    }
}
```

## 响应示例

- 成功响应示例

```
{
  "named_entities": [
    {
      "word": "昨天",
      "tag": "t",
      "offset": 0,
      "len": 2
    },
    {
      "word": "李小明",
      "tag": "nr",
      "offset": 5,
      "len": 3
    },
    {
      "word": "北京",
      "tag": "ns",
      "offset": 10,
      "len": 2
    }
  ]
}
```

- 失败响应示例

```
{
  "error_code": "NLP.0301",
  "error_msg": "The length of text should be in the range of 1-2000."
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.1.5 命名实体识别（领域版）

### 功能介绍

对文本进行命名实体识别分析，目前支持通用、商务和娱乐领域。

- 通用领域：支持人名、地名、组织机构、时间点、日期、百分比、货币额度、序数词、计量规格词、民族、职业、邮箱、国家、节日的实体的识别。
- 商务领域：支持公司名、品牌名、职业、职位、邮箱、手机号码、电话号码、IP地址、身份证号、网址、专业的实体的识别。
- 娱乐领域：支持电影名、动漫、书名、互联网、歌名、产品名、电视剧名、电视节目名的实体的识别。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[领域套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/ner/domain
- 参数说明

表 4-21 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如表4-22所示。

表 4-22 请求参数

参数名	参数类型	必选	说明
text	String	是	待分析文本，长度为1~512，文本编码为UTF-8。
lang	String	否	支持的文本语言类型，目前只支持中文（zh），默认为中文。
domain	String	否	支持的领域类型，目前支持通用（general）领域、商务（business）领域、娱乐（entertainment）领域，默认为general。

## 响应消息

响应参数如表4-23所示。

表 4-23 响应参数

参数名	参数类型	说明
named_entities	Array of named_entity objects	命名实体识别结果，请参见表4-24。
error_code	String	调用失败时的错误码，具体参见错误码。调用成功时无此字段。
error_msg	String	调用失败时的错误信息。调用成功时无此字段。

表 4-24 named\_entity 数据结构说明

参数名	参数类型	说明
word	String	实体文本。

参数名	参数类型	说明
tag	String	实体类型，枚举类型。 <ul style="list-style-type: none"> <li>通用领域：支持人名nr，地名ns，机构名nt，时间点tpt，日期day，百分比pct，货币额度mny，序数词ord，计量规格词qtt，民族race，职业job，邮箱email，国家coun，节日fest。</li> <li>商务领域：支持公司名com、品牌名bra、职业job、职位post、邮箱email、手机号码cell、电话号码tele、IP地址ip、身份证号id、网址web。</li> <li>娱乐领域：支持电影名mov、动漫anime、书名book、互联网int、歌名song、产品名pro、电视剧名dra、电视节目名tv。</li> </ul>
offset	Integer	实体文本在待分析文本中的起始位置。
len	Integer	实体文本长度。

## 请求示例

- 请求示例1（识别分析命名实体，支持的领域类型为通用领域）

POST https://{endpoint}/v1/{project\_id}/nlp-fundamental/ner/domain

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZlhvcNAQcCoIIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:

```
{
  "text": "昨天程序员李小明来到北京参加开发者大赛，在比赛中表现优异，赢得了第一名。",
  "lang": "zh",
  "domain": "general"
}
```

- Python3语言请求代码示例（识别分析命名实体，支持的语言类型为中文）

```
# -*- coding: utf-8 -*-
```

```
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
```

```
import json
```

```
def nlp_demo():
```

```
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/ner/domain' # endpoint和project_id需替换
```

```
    token = '用户对应region的token'
```

```
    header = {
```

```
        'Content-Type': 'application/json',
```

```
        'X-Auth-Token': token
```

```
    }
```

```
    body = {
```

```
        'text': '昨天程序员李小明来到北京参加开发者大赛，在比赛中表现优异，赢得了第一名。',
```

```
        'lang': 'zh'
```

```
    }
```

```
    resp = requests.post(url, data=json.dumps(body), headers=header)
```

```
    print(resp.json())
```

```
if __name__ == '__main__':
    nlp_demo()
```

- **Java语言请求代码示例（识别分析命名实体，支持的领域类型为通用领域）**

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/ner/domain");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String text = "昨天程序员李小明来到北京参加开发者大赛，在比赛中表现优异，赢得了第一名。";
            String body = "{\"text\":\"" + text + "\",\"lang\":\"zh\",\"domain\":\"general\"}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        NLPDemo nlpDemo = new NLPDemo();
        nlpDemo.nlpDemo();
    }
}
```

- **请求示例2（识别分析命名实体，支持商务领域）**

POST https://{endpoint}/v1/{project\_id}/nlp-fundamental/ner/domain

Request Header:

Content-Type: application/json  
X-Auth-Token:

MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBGkqhkiG...

Request Body:

```
{
  "text": "程序员小明是华为的员工，邮箱是xiaoming@xx.com，电话12345678。",
  "lang": "zh",
  "domain": "business"
}
```

- **请求示例3（识别分析命名实体，支持娱乐领域）**

POST https://{endpoint}/v1/{project\_id}/nlp-fundamental/ner/domain

Request Header:

```
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:
{
  "text": "我早上看了动画《海贼王》，下午学唱《童年》，晚上在安徽卫视看《甄嬛传》。",
  "lang": "zh",
  "domain": "entertainment"
}
```

## 响应示例

- 成功响应示例1

```
{
  "named_entities": [
    {
      "len": 2,
      "offset": 0,
      "tag": "day",
      "word": "昨天"
    },
    {
      "len": 3,
      "offset": 2,
      "tag": "job",
      "word": "程序员"
    },
    {
      "len": 3,
      "offset": 5,
      "tag": "nr",
      "word": "李小明"
    },
    {
      "len": 2,
      "offset": 10,
      "tag": "ns",
      "word": "北京"
    },
    {
      "len": 2,
      "offset": 32,
      "tag": "ord",
      "word": "第一"
    }
  ]
}
```

- 失败响应示例1

```
{
  "error_code": "NLP.0301",
  "error_msg": "The length of text should be in the range of 1-512."
}
```

- 成功响应示例2

```
{
  "named_entities": [
    {
      "len": 3,
      "offset": 0,
      "tag": "job",
      "word": "程序员"
    },
    {
      "len": 2,
      "offset": 6,
      "tag": "com",
      "word": "华为"
    }
  ]
}
```

```
    },  
    {  
      "len": 15,  
      "offset": 15,  
      "tag": "email",  
      "word": "xiaoming@xx.com"  
    },  
    {  
      "len": 8,  
      "offset": 33,  
      "tag": "tele",  
      "word": "12345678"  
    }  
  ]  
}
```

- 失败响应示例2

```
{  
  "error_code": "NLP.0301",  
  "error_msg": "The length of text should be in the range of 1-512."  
}
```

- 成功响应示例3

```
{  
  "named_entities": [  
    {  
      "len": 3,  
      "offset": 8,  
      "tag": "anime",  
      "word": "海贼王"  
    },  
    {  
      "len": 2,  
      "offset": 18,  
      "tag": "song",  
      "word": "童年"  
    },  
    {  
      "len": 4,  
      "offset": 25,  
      "tag": "pro",  
      "word": "安徽卫视"  
    },  
    {  
      "len": 3,  
      "offset": 31,  
      "tag": "dra",  
      "word": "甄嬛传"  
    }  
  ]  
}
```

- 失败响应示例3

```
{  
  "error_code": "NLP.0301",  
  "error_msg": "The length of text should be in the range of 1-512."  
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.1.6 文本相似度（基础版）

### 功能介绍

对文本进行语义相似度计算。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[基础套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

也可使用文本相似度（高级版）接口，详情请见[文本相似度（高级版）](#)。文本相似度基础版和高级版基于不同算法实现，对相同文本，基础版和高级版的结果有所差别。根据测试数据，高级版效果一般优于基础版。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/text-similarity
- 参数说明

表 4-25 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

### 请求消息

请求参数如[表4-26](#)所示。

表 4-26 请求参数

参数名	参数类型	必选	说明
text1	String	是	待计算文本1，中文长度1~512，英文长度1~2000，文本编码为UTF-8。
text2	String	是	待计算文本2，中文长度1~512，英文长度1~2000，文本编码为UTF-8。
lang	String	否	支持的文本语言类型，默认为“zh”。目前支持中文和英文，对应的“lang”输入分别为“zh”，“en”。

## 响应消息

响应参数如表4-27所示。

表 4-27 响应参数

参数名	参数类型	说明
similarity	Float	相似度得分，范围在0~1，默认小数点后保留8位。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。调用成功时无此字段。
error_msg	String	调用失败时的错误信息。调用成功时无此字段。

## 请求示例

- 请求示例（计算文本语义相似度）

```
POST https://{endpoint}/v1/{project_id}/nlp-fundamental/text-similarity
Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:
{
  "text1": "今天天气晴朗",
  "text2": "今天天气晴",
  "lang": "zh"
}
```

- Python3语言请求代码示例（计算文本语义相似度）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json
```

```
def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/text-similarity' # endpoint和project_id需
    替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'text1': '订单记录怎么删除',
        'text2': '如何删除浏览记录',
        'lang': 'zh'
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例（计算文本语义相似度）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/text-similarity");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String text1 = "订单记录怎么删除";
            String text2 = "如何删除浏览记录";
            String lang = "zh";
            String body = "{\"text1\":\"" + text1 + "\",\"text2\":\"" + text2 + "\",\"lang\":\"" + lang + "\"}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        NLPDemo nlpDemo = new NLPDemo();
        nlpDemo.nlpDemo();
    }
}
```

## 响应示例

- 成功响应示例

```
{  
  "similarity":0.73369961  
}
```

- 失败响应示例

```
{  
  "error_code": "NLP.0301",  
  "error_msg": "argument valid error:text2.text2 for text similarity should be between 1 to 512 ;text2.must not be blank;"  
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.1.7 文本相似度（高级版）

### 功能介绍

对文本语义相似度计算。

具体Endpoint请参见[终端节点](#)。

本API免费调用，调用限制为2次/秒。

文本相似度基础版和高级版基于不同算法实现，对相同文本，基础版和高级版的结果有所差别。根据测试数据，高级版效果一般优于基础版。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/text-similarity/advance
- 参数说明

表 4-28 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-29](#)所示。

表 4-29 请求参数

参数名	参数类型	必选	说明
text1	String	是	待计算文本1，长度1~64，文本编码为UTF-8。
text2	String	是	待计算文本2，长度1~64，文本编码为UTF-8。
lang	String	否	支持的文本语言类型，目前只支持中文（zh），默认为中文。

## 响应消息

响应参数如[表4-30](#)所示。

表 4-30 响应参数

参数名	参数类型	说明
similarity	Float	相似度得分，范围在0~1，默认小数点后保留8位。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

## 请求示例

- 请求示例（计算文本语义相似度）

POST [https://{endpoint}/v1/{project\\_id}/nlp-fundamental/text-similarity/advance](https://{endpoint}/v1/{project_id}/nlp-fundamental/text-similarity/advance)

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:

```
{
```

```
"text1": "杭州有啥好玩的",  
"text2": "杭州哪里好玩",  
"lang": "zh"  
}
```

- Python3语言请求代码示例（计算文本语义相似度）

```
# -*- coding: utf-8 -*-  
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests  
import requests  
import json  
  
def nlp_demo():  
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/text-similarity/advance' # endpoint和  
    project_id需替换  
    token = '用户对应region的token'  
    header = {  
        'Content-Type': 'application/json',  
        'X-Auth-Token': token  
    }  
    body = {  
        'text1': '订单记录怎么删除',  
        'text2': '如何删除浏览记录',  
        'lang': 'zh'  
    }  
    resp = requests.post(url, data=json.dumps(body), headers=header)  
    print(resp.json())  
  
if __name__ == '__main__':  
    nlp_demo()
```

- Java语言请求代码示例（计算文本语义相似度）

```
import java.io.BufferedReader;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStreamWriter;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
/**  
 * 此demo仅供测试使用，建议使用sdk  
 */  
public class NLPDemo {  
    public void nlpDemo() {  
        try {  
            //endpoint和projectId需要替换成实际信息。  
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/text-similarity/  
advance");  
            String token = "对应region的token";  
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
            connection.setRequestMethod("POST");  
            connection.setDoInput(true);  
            connection.setDoOutput(true);  
            connection.addRequestProperty("Content-Type", "application/json");  
            connection.addRequestProperty("X-Auth-Token", token);  
  
            //输入参数  
            String text1 = "订单记录怎么删除";  
            String text2 = "如何删除浏览记录";  
            String lang = "zh";  
            String body = "{\"text1\": \"\" + text1 + \"\", \"text2\": \"\" + text2 + \"\", \"lang\": \"\" + lang + \"\"";  
  
            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),  
"UTF-8");  
            osw.append(body);  
            osw.flush();  
            InputStream is = connection.getInputStream();  
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));  
            while (br.ready()) {  
                System.out.println(br.readLine());  
            }  
        }  
    }  
}
```

```
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}  
  
public static void main(String[] args) {  
    NLPDemo nlpDemo = new NLPDemo();  
    nlpDemo.nlpDemo();  
}  
}
```

## 响应示例

- 成功响应示例

```
{  
  "similarity":0.93171031  
}
```

- 失败响应示例

```
{  
  "error_code": "NLP.0301",  
  "error_msg": "text1 must not be null. text2 must not be null"  
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.1.8 句向量

### 功能介绍

输入句子，返回对应的句向量。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[基础套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/sentence-embedding
- 参数说明

表 4-31 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数表4-32所示。

表 4-32 请求参数

参数名	参数类型	必选	说明
sentences	Array of strings	是	文本列表，文本长度为1~512，列表大小为1~1000，文本编码为UTF-8，目前只支持中文。
domain	String	否	计算句向量时使用的模型名，目前只支持general，默认为general。

## 响应消息

响应参数如表4-33所示。

表 4-33 响应参数

参数名	参数类型	说明
vectors	Array of floats	句向量结果列表，按输入句子顺序返回句向量，句向量维度默认为100。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

## 请求示例

- 请求示例（计算句向量）  
POST https://{endpoint}/v1/{project\_id}/nlp-fundamental/sentence-embedding  
Request Header:

```
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:
{
  "sentences":["今天张三来学校了","今天李四来学校了"],
  "domain":"general"
}
```

- Python3语言请求代码示例（计算句向量）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/sentence-embedding' # endpoint和
    project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'sentences': ['今天张三来学校了', '今天李四来学校了'],
        'domain': 'general'
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例（计算句向量）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/sentence-embedding");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String sentences = "[" + "今天张三来学校了" + "," + "今天李四来学校了" + "]";
            String body = "{\"sentences\": " + sentences + ", \"domain\": \"general\"}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
```

```
        System.out.println(br.readLine());
    }
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "vectors": [
    [0.1331, 0.0488, 0.2441, 0.2514, -0.6771, 0.4782, 0.6759, 0.015, 0.0064, -0.6326, 0.3958, -0.6848,
    0.1118, -0.1391, 0.4804, 0.9294, 0.1004, 0.2414, 0.2477, -0.8162, 1.2052, -0.6719, -0.47, -0.1946,
    -0.0606, 0.473, 0.0247, -0.3857, 1.1637, -0.6092, -0.5512, -0.2389, -0.2168, 0.1673, -0.4124, -0.1196,
    -0.7147, 1.1774, -0.8166, 0.1285, -0.3136, 0.4687, -0.5939, -0.4579, 0.1857, 0.049, -0.5936, -0.4554,
    -0.1878, 0.017],
    [0.0833, -0.0731, 0.298, 0.0085, -0.6858, 0.529, 0.887, 0.1772, -0.118, -0.7559, 0.1995, -0.6415,
    0.3014, 0.2061, 0.9727, 0.9089, 0.1603, 0.3773, -0.146, -0.6429, 1.4808, -0.7797, -0.6061, -0.0854,
    -0.1324, 0.3183, 0.3378, -0.4552, 1.4929, -0.7543, -0.6089, -0.1906, -0.1892, 0.0628, -0.4675, -0.2478,
    -0.7632, 1.1876, -1.0734, -0.0954, -0.2896, 0.5757, -0.5601, -0.2595, 0.3831, 0.4729, -0.8736, -0.4378,
    -0.2519, 0.0448]
  ]
}
```

- 失败响应示例

```
{
  "error_code": "NLP.0301",
  "error_msg": "argument valid error: sentence must not be blank and sentence length 1-512"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.1.9 实体链接

### 功能介绍

针对通用领域的文本进行实体链接分析，识别出其中的实体，并返回实体相关信息。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API以[定制版API定价](#)按需计费，**不支持使用套餐包**，使用时请在[自然语言处理价格计算器](#) 按需计费-自然语言处理定制版API中查看费用详情。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

## 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/entity-linking
- 参数说明

表 4-34 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-35](#)所示。

表 4-35 请求参数

参数名	参数类型	必选	说明
text	String	是	待分析文本，长度为1~50，文本编码为UTF-8。
lang	String	否	支持的文本语言类型，目前只支持中文（zh），默认为中文。

## 响应消息

响应参数如[表4-36](#)所示。

表 4-36 响应参数

参数名	参数类型	说明
entities	Array of entities	实体链接结果，请参见 <a href="#">表4-37</a> 。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。

参数名	参数类型	说明
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-37 entity 数据结构说明

参数名	参数类型	说明
mention	String	实体指称。
offset	Integer	偏移量。
entity_title	String	指称链接到的实体名称。

## 请求示例

- 请求示例（分析“李娜唱的青藏高原真好听”的实体链接）

POST https://{endpoint}/v1/{project\_id}/nlp-fundamental/entity-linking

Request Header:

```
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZlhvcNAQcCoIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgguVBgkqhkiG...
```

Request Body:

```
{
  "text": "李娜唱的青藏高原真好听",
  "lang": "zh"
}
```
- Python3语言请求代码示例（分析“李娜唱的青藏高原真好听”的实体链接）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/entity-linking' # endpoint和project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'text': '李娜唱的青藏高原真好听',
        'lang': 'zh'
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```
- Java语言请求代码示例（分析“李娜唱的青藏高原真好听”的实体链接）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
```

```
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/entity-linking");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String text = "李娜唱的青藏高原真好听";
            String body = "{\"text\":\"" + text + "\",\"lang\":\"zh\"}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        NLPDemo nlpDemo = new NLPDemo();
        nlpDemo.nlpDemo();
    }
}
```

## 响应示例

- 成功响应示例

```
{
  "entities": [
    {
      "mention": "李娜",
      "offset": 0,
      "entity_title": "李娜（流行歌手、佛门女弟子）"
    },
    {
      "mention": "青藏高原",
      "offset": 4,
      "entity_title": "青藏高原（李娜演唱歌曲）"
    }
  ]
}
```

- 失败响应示例

```
{
  "error_code": "NLP.0301",
  "error_msg": "argument valid error:text.text for entity-linking should between 1 and 50;"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.1.10 关键词抽取

### 功能介绍

根据指定文本，抽取其中最能够反映文本主题或者意思的词汇。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[基础套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/keyword-extraction
- 参数说明

表 4-38 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

### 请求消息

请求参数如表 [请求参数](#) 所示。

表 4-39 请求参数

参数名	参数类型	必选	说明
text	String	是	待分析文本，长度为1~512，文本编码为UTF-8。
limit	integer	否	返回关键词的最大数量，默认为5。 如果请求文本的词汇数量小于该值，则返回实际词汇数量。 如果词汇数量为浮点数，则limit取该浮点数向下取整的结果。
lang	String	否	支持的文本语言类型，目前只支持中文（zh），默认为中文。

## 响应消息

响应参数如表4-40所示。

表 4-40 响应参数

参数名	参数类型	说明
words	Array of String	关键词列表。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

## 请求示例

- 请求示例（抽取关键词，且返回关键词的最大数量为8）

```
POST https://{endpoint}/v1/{project_id}/nlp-fundamental/keyword-extraction

Request Header:
Content-Type: application/json
X-Auth-Token:
MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:
{
  "text": "华为技术有限公司成立于1987年，总部位于广东省深圳市龙岗区。华为是全球领先的信息与通信技术（ICT）解决方案供应商，专注于ICT领域，坚持稳健经营、持续创新、开放合作，在电信运营商、企业、终端和云计算等领域构筑了端到端的解决方案优势，为运营商客户、企业客户和消费者提供有竞争力的ICT解决方案、产品和服务，并致力于实现未来信息社会、构建更美好的全联接世界。",
  "limit": 8,
  "lang": "zh"
}
```
- Python3语言请求代码示例（抽取关键词，且返回关键词的最大数量为8）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/keyword-extraction' # endpoint和
    project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'text': '华为技术有限公司成立于1987年，总部位于广东省深圳市龙岗区。华为是全球领先的信息与通
        信技术（ICT）解决方案供应商，专注于ICT领域，坚持稳健经营、持续创新、开放合作，在电信运营商、企
        业、终端和云计算等领域构筑了端到端的解决方案优势，为运营商客户、企业客户和消费者提供有竞争力
        的ICT解决方案、产品和服务，并致力于实现未来信息社会、构建更美好的全联接世界。',
        'limit': 8,
        'lang': 'zh'
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例（抽取关键词，且返回关键词的最大数量为5）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/keyword-extraction");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String text = "华为技术有限公司成立于1987年，总部位于广东省深圳市龙岗区。华为是全球领先
            的信息与通信技术（ICT）解决方案供应商，专注于ICT领域，坚持稳健经营、持续创新、开放合作，在电信
            运营商、企业、终端和云计算等领域构筑了端到端的解决方案优势，为运营商客户、企业客户和消费者提
            供有竞争力的ICT解决方案、产品和服务，并致力于实现未来信息社会、构建更美好的全联接世界。";
            String body = "{\"text\":\"" + text + "\",\"limit\": 5}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
            "UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "words": [
    "解决方案",
    "ICT",
    "信息",
    "企业",
    "领域",
    "技术",
    "运营商",
    "华为"
  ]
}
```

- 失败响应示例

```
{
  "error_code": "NLP.0301",
  "error_msg": "argument valid error:lang.only support language:zh;"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.1.11 事件抽取

### 功能介绍

事件抽取是指从自然语言文本中抽取指定类型的事件以及相关实体信息，并形成结构化数据输出的文本处理技术。

目前只支持金融公告中会议召开、聘任、辞职、股票增持、股票减持5类事件以及相关要素的抽取。

本API免费调用，调用限制为2次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

## 📖 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/event-extraction
- 参数说明

表 4-41 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表 请求参数](#)所示。

表 4-42 请求参数

参数名	参数类型	必选	说明
text	String	是	待分析文本，长度为1~256，文本编码为UTF-8。

## 响应消息

响应参数如[表4-43](#)所示。

表 4-43 响应参数

参数名	参数类型	说明
events	Array of events	事件抽取结果。 请参见 <a href="#">表4-44</a> 。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-44 events 数据结构说明

参数名	参数类型	说明
argument	Array of arguments	事件元素列表。
event_trigger	String	事件触发词。触发词是事件描述中最能代表事件发生的词汇，决定事件类别的重要特征。
event_type	String	事件类型。
trigger_span	List of Integer	事件触发词在待分析文本中的起始和终止位置。

表 4-45 events.argument 数据结构说明

参数名	参数类型	说明
role	String	元素角色。元素角色指的是事件元素在事件中扮演的角色，是事件元素与事件的语义关系。
span	List of Integer	实体文本在待分析文本中的起始和终止位置。
word	String	实体文本。

表 4-46 事件模板

事件类型	模板元素	说明
会议召开	ORG	公司名称
	Time	会议时间
	Place	会议地点
	Name	会议名称
聘任	ORG	公司名称
	Person	聘任对象
	Job	职位
	Time-Period	任期
辞职	ORG	公司名称
	Person	辞职对象
	Job	职位

事件类型	模板元素	说明
	Time	辞职时间
股票增持	Obj	发行股票的公司
	Sub	增持主体(个人或组织)
	Time-Ending	增持日期
	Increasing-Num	增持股数
	Increasing-Rate	增持比例
	After-Num	增持后持股数
	After-Rate	增持后持股比例
股票减持	Obj	发行股票的公司
	Sub	减持主体(个人或组织)
	Time-Ending	减持日期
	Decreasing-Num	减持股数
	Decreasing-Rate	减持比例
	After-Num	减持后持股数
	After-Rate	减持后持股比例

## 请求示例

- 请求示例（抽取事件）

POST https://{endpoint}/v1/{project\_id}/nlp-fundamental/event-extraction

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:

```
{
  "text": "中国XXX股份有限公司（以下简称“公司”）第五届董事会第一次会议上，同意选举董事黄XX为审计委员会委员。审计委员会任期与本届董事会任期一致"
}
```

- Python3语言请求代码示例（抽取事件）

```
# -*- coding: utf-8 -*-
```

```
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
```

```
import json
```

```
def nlp_demo():
```

```
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/event-extraction' # endpoint和project_id需替换
```

```
    token = '用户对应region的token'
```

```
    header = {
```

```
        'Content-Type': 'application/json',
```

```
        'X-Auth-Token': token
```

```
    }
```

```
    body = {
```

```
'text': '中国XXX股份有限公司（以下简称“公司”）第五届董事会第一次会议上，同意选举董事黄XX为审计委员会委员。审计委员会任期与本届董事会任期一致'
}
resp = requests.post(url, data=json.dumps(body), headers=header)
print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- **Java语言请求代码示例（抽取事件）**

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/event-extraction");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String text = "中国XXX股份有限公司（以下简称“公司”）第五届董事会第一次会议上，同意选举董事黄XX为审计委员会委员。审计委员会任期与本届董事会任期一致";
            String body = "{\"text\":\"" + text + "\"}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        NLPDemo nlpDemo = new NLPDemo();
        nlpDemo.nlpDemo();
    }
}
```

## 响应示例

- **成功响应示例**

```
{
  "events": [
    {
      "argument": [
        {
          "role": "ORG",
          "span": [0,11],
          "word": "中国XXX股份有限公司"
        }
      ]
    }
  ]
}
```

```
    },
    {
      "role": "Person",
      "span": [40,43],
      "word": "黄XX"
    },
    {
      "role": "Job",
      "span": [44,51],
      "word": "审计委员会委员"
    },
    {
      "role": "Time-Period",
      "span": [57,69],
      "word": "任期与本届董事会任期一致"
    }
  ],
  "event_trigger": "选举",
  "event_type": "聘任",
  "trigger_span": [36,38]
}
]
```

- 失败响应示例

```
{
  "error_code": "NLP.0301",
  "error_msg": "text should be between 1 to 256."
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.1.12 成分句法分析

### 功能介绍

识别句子中的成分以及成分之间的层次包含关系。

本API免费调用，调用限制为2次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlp-fundamental/constituency-parser
- 参数说明

表 4-47 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如表 [请求参数](#) 所示。

表 4-48 请求参数

参数名	参数类型	必选	说明
text	String	是	待分析文本，长度为1~32，文本编码为UTF-8。
lang	String	否	支持的文本语言类型，目前只支持中文（zh），默认为中文。

## 响应消息

响应参数如表 [响应参数](#) 所示。

表 4-49 响应参数

参数名	参数类型	说明
tree	String	成分句法分析结果，用嵌套括号的形式表示一棵树。括号内的第一个元素为子树的标签，如果是叶子节点则用"_"代替。第二个元素为子树，如果是叶子节点则为字符串。
error_code	String	调用失败时的错误码，具体参见 <a href="#">6.2 错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-50 成分标签说明

成分标签	说明
ADJP	adjective phrase (形容词短语)
ADVP	adverbial phrase headed by AD (副词开头的副词短语)
CLP	classifier phrase (量词短语)
CP	clause headed by complementizer (补语性嵌套句的从属连词引起的分句)
DNP	phrase formed by “XP+DEG” (XP+DEG结构构成的短语)
DP	determiner phrase (限定词短语)
DVP	phrase formed by “XP+DEV” (XP+DEV结构构成的短语)
FRAG	fragment (片段语)
INTJ	interjection (感叹词)
IP	simple clause headed by INFL (INFL为开头的简单从句)
LCP	phrase formed by “XP+LC” (XP+LC结构构成的短语)
LST	list marker (用于解释说明性的列表标记短语)
NP	noun phrase (名词短语)
PP	preposition phrase (介词短语)
PRN	parenthetical (插入语)
QP	quantifier phrase (量词短语)
UCP	unidentical coordination phrase (非一致性并列短语)
VCD	coordinated verb compound (并列动词复合词)
VCP	verb compounds formed by “VV+VC” (VV+VC结构构成的动词复合词)
VNV	verb compounds formed by “A-not-A” or “A-one-A” (A-not-A或A-one-A结构构成的动词复合词)
VP	verb phrase (动词短语)
VPT	potential form V-de-R or V-bu-R (构成V-的-R或V-不-R的动词复合词)
VRD	verb resultative compound (动词结果性复合词)

成分标签	说明
VSB	verb compounds formed by a modifier + a head ( 定语+核心词构成的动词复合词 )
WHPP	wh-prepositional phrase ( wh开头的介词短语 )

## 请求示例

- 请求示例 ( 分析"张三买电脑"的成分句法 )

```
POST https://{endpoint}/v1/{project_id}/nlp-fundamental/constituency-parser

Request Header:
Content-Type: application/json
X-Auth-Token:
MIIFbwYJKoZIHvcNAQcCoIIFYDCCBVwCAQExDTALBglghkgBZQMEAgEwggNBgkqhkiG9...
```

```
Request Body: {
  "text": "张三买电脑",
  "lang": "zh"
}
```
- Python3语言请求代码示例 ( 分析"张三买电脑"的成分句法 )

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用, 建议使用sdk。需提前安装requests, 执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlp-fundamental/constituency-parser' # endpoint和project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'text': '张三买电脑',
        'lang': 'zh'
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```
- Java语言请求代码示例 ( 分析"张三买电脑"的成分句法 )

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用, 建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlp-fundamental/constituency-parser");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
```

```
connection.addRequestProperty("Content-Type", "application/json");
connection.addRequestProperty("X-Auth-Token", token);

//输入参数
String text = "张三买电脑";
String body = "{\"text\":\"" + text + "\",\"lang\":\"zh\"}";

OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
osw.append(body);
osw.flush();
InputStream is = connection.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
while (br.ready()) {
    System.out.println(br.readLine());
}
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "tree": "(IP (NP ( _ 张三))(VP ( _ 买)(NP ( _ 电脑))))"
}
```

- 失败响应示例

```
{
  "error_code": "NLP.0301",
  "error_message": "Missing parameters:text"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.2 语言生成服务接口说明

### 4.2.1 文本摘要（基础版）

#### 功能介绍

对文本生成摘要。

文本摘要（基础版）和文本摘要（领域版）基于不同算法实现，对相同文本，基础版和领域版的结果有所差别。根据测试数据，领域版效果一般优于基础版。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用**基础套餐包**，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

## 调试

您可以在[API Explorer](#)中调试该接口。

## 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlg/summarization
- 参数说明

表 4-51 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-52](#)所示。

表 4-52 请求参数

参数名	参数类型	必选	说明
lang	String	否	支持的文本语言类型，目前支持中文（zh）和英文（en），默认为中文。
title	String	否	文本标题（目前仅支持UTF-8编码），长度为0~1000。
content	String	是	文本正文（目前仅支持UTF-8编码），长度为1~10000。 <b>说明</b> 文本语言类型需和lang字段保持一致。

参数名	参数类型	必选	说明
length_limit	Float	否	生成摘要的长度限制。 <ul style="list-style-type: none"> <li>length_limit &gt; 1, 则返回结果为字数不小于该值且最接近该值的摘要。</li> <li>0 &lt;= length_limit &lt;= 1, 则返回结果为长度百分比不小于该值且最接近该值的摘要。</li> </ul> 默认数值为0.3。

## 响应消息

响应参数如表4-53所示。

表 4-53 响应参数

参数名	参数类型	说明
summary	String	根据文本请求体，返回摘要结果。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

## 请求示例

- 请求示例（生成文本摘要，长度限制为0.3）

Request Header:

Content-Type: application/json  
X-Auth-Token:

MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG... //用户认证获取的token值，获取方法详细请参考“如何调用API”章节中的“认证鉴权”页面。

Request Body:

```
{
  "length_limit":0.3,
  "title":"华为“刀片式基站”获2018年度国家科学技术进步奖一等奖",
  "lang": "zh",
  "content": "华为刀片式基站解决方案是华为在深入理解客户诉求基础上，指引业界的创新解决方案。
```

该方案采用统一的模块化设计，实现基站主要元素如射频、基带、电源、电池、微波、传输的刀片化，不同模块间的任意快速拼装组合无缝拼装，能灵活安装在抱杆、铁塔、墙面或者屋顶，“0”站址无需机房机柜，使基站的安装像拼装乐高积木一样简单便捷。刀片式基站同时支持2G/3G/4G等多制式，在多频多模网络发展策略下可以高效利用宝贵的站点资源，大幅降低站点获取难度和减少站点租金，帮助运营商应对移动网络快速增长的容量需求。同时刀片式基站采用自然散热，满足室外55℃高温环境，IP65防护等级，无需机房机柜和空调，其高能效和环境友好的特性，帮助运营商打造绿色移动网络。自2012年推出以来，刀片式基站全球累计已发货超1500万片，在全球超过170个国家310张运营商网络中成功商用部署。全球客户高度肯定了华为创新刀片式基站解决方案，它不但打破了传统机柜站点占地面积大、运维复杂的建站模式，而且还有效地提高了站点的部署效率，特别在密集城区、高铁场景下解决站点空间受限、实现快速部署、降低租赁成本等方面效果显著，同时也为乡村广覆盖场景提供最简单站点方案。在5G时代，华为围绕客户需求持续创新，在2018全球移动宽带论坛上，华为亦推出Super Blade Site——面向5G全室外站解决方案，其中包含最新支持5G容量要求的室外基带单元Blade BBU和有源天线一体化产品Blade AAU，该解决方

案进一步匹配5G的新技术要求和容量要求，极大降低5G引入对天面空间的需求，加速Massive MIMO部署，帮助运营商布局5G网络。华为Super Blade Site在2018全球移动宽带论坛上展出华为无线网络研发总裁郇舟剑表示，“华为一直致力于围绕客户需求持续创新，刀片式基站就是典型的例子。刀片式基站解决客户获取站址难题，帮助运营商快速建站，满足移动业务的迅猛增长的需求，也给客户带来商业成功。同时，面对即将到来的5G时代，华为持续创新，全室外刀片基站解决方案，将助力5G快速规模商用，帮助客户取得更大的商业成功。”国家科学技术进步奖，是国务院设立的国家科学技术奖5大奖项（国家最高科学技术奖、国家自然科学奖、国家技术发明奖、国家科学技术进步奖、国际科学技术合作奖）之一。该奖项授予在技术研究、技术开发、技术创新、推广应用先进科学技术成果、促进高新技术产业化，以及完成重大科学技术工程、计划等过程中做出创造性贡献的中国公民和组织。”

- Python3语言请求代码示例（生成文本摘要，长度限制为0.3）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlg/summarization' # endpoint和project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'length_limit': 0.3,
        'title': '华为“刀片式基站”获2018年度国家科学技术进步奖一等奖',
        'lang': 'zh',
        'content': '华为刀片式基站解决方案是华为在深入理解客户诉求基础上，指引业界的创新解决方案。该方案采用统一的模块化设计，实现基站主要元素如射频、基带、电源、电池、微波、传输的刀片化，不同模块间的任意快速拼装组合无缝拼装，能灵活安装在抱杆，铁塔，墙面或者屋顶，“0”站址无需机房机柜，使基站的安装像拼装乐高积木一样简单便捷。刀片式基站同时支持2G/3G/4G等多制式，在多频多模网络发展策略下可以高效利用宝贵的站点资源，大幅降低站点获取难度和减少站点租金，帮助运营商应对移动网络快速增长的容量需求。同时刀片式基站采用自然散热，满足室外55℃高温环境，IP65防护等级，无需机房机柜和空调，其高能效和环境友好的特性，帮助运营商打造绿色移动网络。自2012年推出以来，刀片式基站全球累计已发货超1500万片，在全球超过170个国家310张运营商网络中成功商用部署。全球客户高度肯定了华为创新刀片式基站解决方案，它不但打破了传统机柜站点占地面积大、运维复杂的建站模式，而且还有效地提高了站点的部署效率，特别在密集城区、高铁场景下解决站点空间受限、实现快速部署、降低租赁成本等方面效果显著，同时也为乡村广覆盖场景提供最简单站点方案。在5G时代，华为围绕客户需求持续创新，在2018全球移动宽带论坛上，华为亦推出Super Blade Site——面向5G全室外站解决方案，其中包含最新支持5G容量要求的室外基带单元Blade BBU和有源天线一体化产品Blade AAU，该解决方案进一步匹配5G的新技术要求和容量要求，极大降低5G引入对天面空间的需求，加速Massive MIMO部署，帮助运营商布局5G网络。华为Super Blade Site在2018全球移动宽带论坛上展出华为无线网络研发总裁郇舟剑表示，“华为一直致力于围绕客户需求持续创新，刀片式基站就是典型的例子。刀片式基站解决客户获取站址难题，帮助运营商快速建站，满足移动业务的迅猛增长的需求，也给客户带来商业成功。同时，面对即将到来的5G时代，华为持续创新，全室外刀片基站解决方案，将助力5G快速规模商用，帮助客户取得更大的商业成功。”国家科学技术进步奖，是国务院设立的国家科学技术奖5大奖项（国家最高科学技术奖、国家自然科学奖、国家技术发明奖、国家科学技术进步奖、国际科学技术合作奖）之一。该奖项授予在技术研究、技术开发、技术创新、推广应用先进科学技术成果、促进高新技术产业化，以及完成重大科学技术工程、计划等过程中做出创造性贡献的中国公民和组织。'
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例（生成文本摘要，长度限制为0.3）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
```

```
try {
    //endpoint和projectId需要替换成实际信息。
    URL url = new URL("https://{endpoint}/v1/{project_id}/nlg/summarization");
    String token = "对应region的token";
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    connection.setRequestMethod("POST");
    connection.setDoInput(true);
    connection.setDoOutput(true);
    connection.addRequestProperty("Content-Type", "application/json");
    connection.addRequestProperty("X-Auth-Token", token);

    //输入参数
    String title = "小德3-0费德勒首进澳网决赛 天王遭遇08年首败";
    String content = "搜狐体育讯 北京时间1月25日,2008年首项大满贯赛事--澳大利亚网球公开赛进入第12日角逐。在罗德-拉沃尔球场进行的男单第二场半决赛争夺中,大赛3号种子、塞尔维亚天才少年德约科维奇出人意料的直落三盘,以总比分3-0横扫大赛头号种子、瑞士天王费德勒,职业生涯首进澳网男单决赛,同时这也是德约科维奇第二次杀入大满贯赛事男单决赛。三盘的具体比分为7-5、6-3和7-6(5)。费德勒无缘卫冕,这是瑞士天王在2008年输掉的首场比赛,也是他自从2005年法网之后,首度无缘大满贯赛事男单决赛;瑞士天王在墨尔本的19连胜就此终结。德约科维奇在男单决赛中的对手将是法国黑人小将特松加。您认为费天王失利的原因是什么?作为世界头号选手,以及三届澳网冠军,费德勒这两年在墨尔本的表现相当不错。本届澳网,费德勒的状态很一般,特别是第三轮他经过五盘苦战才力擒塞尔维亚眼镜侠提普萨勒维奇。在此之前,费德勒连续10项大满贯闯入决赛;连续15项大满贯晋级四强。德约科维奇已经创下个人在澳网的最佳战绩,这位塞尔维亚天才少年渴望连续第二年挺进大满贯赛事男单决赛。在此之前,费德勒与德约科维奇总共共有过六次交锋。";
    String body = "{\"title\":\"\" + title + "\",\"content\":\"\" + content + "\",\"length_limit\":" + 0.3 + "\",\"lang\":\"zh\"}";

    OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
    osw.append(body);
    osw.flush();
    InputStream is = connection.getInputStream();
    BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
    while (br.ready()) {
        System.out.println(br.readLine());
    }
} catch (Exception e) {
    e.printStackTrace();
}

}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "summary": "华为刀片式基站解决方案是华为在深入理解客户诉求基础上,指引业界的创新解决方案。自2012年推出以来,刀片式基站全球累计已发货超1500万片,在全球超过170个国家310张运营商网络中成功商用部署。在5G时代,华为围绕客户需求持续创新,在2018全球移动宽带论坛上,华为亦推出Super Blade Site——面向5G全室外站解决方案,其中包含最新支持5G容量要求的室外基带单元Blade BBU和有源天线一体化产品Blade AAU,该解决方案进一步匹配5G的新技术要求和容量要求,极大降低5G引入对天面空间的需求,加速Massive MIMO部署,帮助运营商布局5G网络。华为Super Blade Site在2018全球移动宽带论坛上展出华为无线网络研发总裁郇舟剑表示,“华为一直致力于围绕客户需求持续创新,刀片式基站就是典型的例子。”国家科学技术进步奖,是国务院设立的国家科学技术奖5大奖项(国家最高科学技术奖、国家自然科学奖、国家技术发明奖、国家科学技术进步奖、国际科学技术合作奖)之一。”
}
```

- 失败响应示例

```
{
  "error_code": "NLP.3201",
  "error_msg": "parameter error."
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.2.2 文本摘要（领域版）

### 功能介绍

根据不同领域的语料训练特定领域的摘要模型，能有效提升文本摘要的效果。

文本摘要（基础版）和文本摘要（领域版）基于不同算法实现，对相同文本，基础版和领域版的结果有所差别。根据测试数据，领域版效果一般优于基础版。

本API免费调用，调用限制为2次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
POST /v1/{project\_id}/nlg/summarization/domain
- 参数说明

表 4-54 URI 参数说明

名称	是否必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

### 请求消息

请求参数如下所示。

表 4-55 请求参数

名称	是否必选	参数类型	说明
lang	否	String	支持的文本语言类型，目前支持中文（zh），默认为中文。
title	否	String	文本标题（目前仅支持UTF-8编码），长度不超过1000字。
content	是	String	文本正文（目前仅支持UTF-8编码），长度1~10000字。
length_limit	否	float	生成摘要的长度限制。 <ul style="list-style-type: none"> <li>length_limit &gt; 1，则为摘要的具体字数。</li> <li>0 &lt;= length_limit &lt;= 1，则为生成摘要占原文长度的百分比。</li> </ul> 默认数值为0.3。
type	否	Integer	支持的领域类型，取值如下（目前只支持通用领域），默认为通用领域： 0：通用领域

## 响应消息

响应参数如表4-56所示。

表 4-56 响应参数

名称	参数类型	说明
summary	String	根据文本请求体，返回摘要结果。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

## 请求示例

- 请求示例（生成文本摘要，长度限制为0.3）

POST https://{endpoint}/v1/{project\_id}/nlg/summarization/domain

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZlhcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG... //用户认证获取的token值，获取方法详细请参考“如何调用API”章节中的“认证鉴权”页面。

Request Body:

```
{
```

```
"length_limit":0.3,
"title":"华为“刀片式基站”获2018年度国家科学技术进步奖一等奖",
"lang":"zh",
"content":"华为刀片式基站解决方案是华为在深入理解客户诉求基础上，引领业界的创新解决方案。该方案采用统一的模块化设计，实现基站主要元素如射频、基带、电源、电池、微波、传输的刀片化，不同模块间的任意快速拼装组合无缝拼装，能灵活安装在抱杆，铁塔，墙面或者屋顶，“0”站址无需机房机柜，使基站的安装像拼装乐高积木一样简单便捷。刀片式基站同时支持2G/3G/4G等多制式，在多频多模网络发展策略下可以高效利用宝贵的站点资源，大幅降低站点获取难度和减少站点租金，帮助运营商应对移动网络快速增长的容量需求。同时刀片式基站采用自然散热，满足室外55℃高温环境，IP65防护等级，无需机房机柜和空调，其高能效和环境友好的特性，帮助运营商打造绿色移动网络。自2012年推出以来，刀片式基站全球累计已发货超1500万片，在全球超过170个国家310张运营商网络中成功商用部署。全球客户高度肯定了华为创新刀片式基站解决方案，它不但打破了传统机柜站点占地面积大、运维复杂的建站模式，而且还有效地提高了站点的部署效率，特别在密集城区、高铁场景下解决站点空间受限、实现快速部署、降低租赁成本等方面效果显著，同时也为乡村广覆盖场景提供最简单站点方案。在5G时代，华为围绕客户需求持续创新，在2018全球移动宽带论坛上，华为亦推出Super Blade Site——面向5G全室外站解决方案，其中包含最新支持5G容量要求的室外基带单元Blade BBU和有源天线一体化产品Blade AAU，该解决方案进一步匹配5G的最新技术要求和容量要求，极大降低5G引入对天面空间的需求，加速Massive MIMO部署，帮助运营商布局5G网络。华为Super Blade Site在2018全球移动宽带论坛上展出华为无线网络研发总裁邬舟剑表示，“华为一直致力于围绕客户需求持续创新，刀片式基站就是典型的例子。刀片式基站解决客户获取站址难题，帮助运营商快速建站，满足移动业务的迅猛增长的需求，也给客户带来商业成功。同时，面对即将到来的5G时代，华为持续创新，全室外刀片基站解决方案，将助力5G快速规模商用，帮助客户取得更大的商业成功。”国家科学技术进步奖，是国务院设立的国家科学技术奖5大奖项（国家最高科学技术奖、国家自然科学奖、国家技术发明奖、国家科学技术进步奖、国际科学技术合作奖）之一。该奖项授予在技术研究、技术开发、技术创新、推广应用先进科学技术成果、促进高新技术产业化，以及完成重大科学技术工程、计划等过程中做出创造性贡献的中国公民和组织。”，
"type":0
}
```

- Python3语言请求代码示例（生成文本摘要，长度限制为0.3）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlg/summarization/domain' # endpoint和project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'length_limit': 0.3,
        'title': '华为“刀片式基站”获2018年度国家科学技术进步奖一等奖',
        'lang': 'zh',
        'content': '华为刀片式基站解决方案是华为在深入理解客户诉求基础上，引领业界的创新解决方案。该方案采用统一的模块化设计，实现基站主要元素如射频、基带、电源、电池、微波、传输的刀片化，不同模块间的任意快速拼装组合无缝拼装，能灵活安装在抱杆，铁塔，墙面或者屋顶，“0”站址无需机房机柜，使基站的安装像拼装乐高积木一样简单便捷。刀片式基站同时支持2G/3G/4G等多制式，在多频多模网络发展策略下可以高效利用宝贵的站点资源，大幅降低站点获取难度和减少站点租金，帮助运营商应对移动网络快速增长的容量需求。同时刀片式基站采用自然散热，满足室外55℃高温环境，IP65防护等级，无需机房机柜和空调，其高能效和环境友好的特性，帮助运营商打造绿色移动网络。自2012年推出以来，刀片式基站全球累计已发货超1500万片，在全球超过170个国家310张运营商网络中成功商用部署。全球客户高度肯定了华为创新刀片式基站解决方案，它不但打破了传统机柜站点占地面积大、运维复杂的建站模式，而且还有效地提高了站点的部署效率，特别在密集城区、高铁场景下解决站点空间受限、实现快速部署、降低租赁成本等方面效果显著，同时也为乡村广覆盖场景提供最简单站点方案。在5G时代，华为围绕客户需求持续创新，在2018全球移动宽带论坛上，华为亦推出Super Blade Site——面向5G全室外站解决方案，其中包含最新支持5G容量要求的室外基带单元Blade BBU和有源天线一体化产品Blade AAU，该解决方案进一步匹配5G的最新技术要求和容量要求，极大降低5G引入对天面空间的需求，加速Massive MIMO部署，帮助运营商布局5G网络。华为Super Blade Site在2018全球移动宽带论坛上展出华为无线网络研发总裁邬舟剑表示，“华为一直致力于围绕客户需求持续创新，刀片式基站就是典型的例子。刀片式基站解决客户获取站址难题，帮助运营商快速建站，满足移动业务的迅猛增长的需求，也给客户带来商业成功。同时，面对即将到来的5G时代，华为持续创新，全室外刀片基站解决方案，将助力5G快速规模商用，帮助客户取得更大的商业成功。”国家科学技术进步奖，是国务院设立的国家科学技术奖5大奖项（国家最高科学技术奖、国家自然科学奖、国家技术发明奖、国家科学技术进步奖、国际科学技术合作奖）之一。该奖项授予在技术研究、技术开发、技术创新、推广应用先进科学技术成果、促进高新技术产业化，以及完成重大科学技术工程、计划等过程中做出创造性贡献的中国公民和组织。',
        'type': 0
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
```

```
print(resp.json())
```

```
if __name__ == '__main__':  
    nlp_demo()
```

- Java语言请求代码示例（生成文本摘要，长度限制为0.3）

```
import java.io.BufferedReader;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStreamWriter;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
/**  
 * 此demo仅供测试使用，建议使用sdk  
 */  
public class NLPDemo {  
    public void nlpDemo() {  
        try {  
            //endpoint和projectId需要替换成实际信息。  
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlg/summarization/domain");  
            String token = "对应region的token";  
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
            connection.setRequestMethod("POST");  
            connection.setDoInput(true);  
            connection.setDoOutput(true);  
            connection.addRequestProperty("Content-Type", "application/json");  
            connection.addRequestProperty("X-Auth-Token", token);  
  
            //输入参数  
            String title = "小德3-0费德勒首进澳网决赛 天王遭遇08年首败";  
            String content = "搜狐体育讯 北京时间1月25日,2008年首项大满贯赛事--澳大利亚网球公开赛进入第12日角逐。在罗德-拉沃尔球场进行的男单第二场半决赛争夺中,大赛3号种子、塞尔维亚天才少年德约科维奇出人意料的直落三盘,以总比分3-0横扫大赛头号种子、瑞士天王费德勒,职业生涯首进澳网男单决赛,同时这也是德约科维奇第二次杀入大满贯赛事男单决赛。三盘的具体比分为7-5、6-3和7-6(5)。费德勒无缘卫冕,这是瑞士天王在2008年输掉的首场比赛,也是他自从2005年法网之后,首度无缘大满贯赛事男单决赛;瑞士天王在墨尔本的19连胜就此终结。德约科维奇在男单决赛中的对手将是法国黑人小将特松加。您认为费天王失利的原因是什么?作为世界头号选手,以及三届澳网冠军,费德勒这两年在墨尔本的表现相当不错。本届澳网,费德勒的状态很一般,特别是第三轮他经过五盘苦战才力擒塞尔维亚眼镜侠提普萨勒维奇。在此之前,费德勒连续10项大满贯闯入决赛;连续15项大满贯晋级四强。德约科维奇已经创下个人在澳网的最佳战绩,这位塞尔维亚天才少年渴望连续第二年挺进大满贯赛事男单决赛。在此之前,费德勒与德约科维奇总共有过六次交锋。";  
            String body = "{\"title\":\"" + title + "\",\"content\":\"" + content + "\",\"length_limit\":" + 0.3, \"lang\": \"zh\", \"type\": 0}";  
  
            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),  
"UTF-8");  
            osw.append(body);  
            osw.flush();  
            InputStream is = connection.getInputStream();  
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));  
            while (br.ready()) {  
                System.out.println(br.readLine());  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
  
    public static void main(String[] args) {  
        NLPDemo nlpDemo = new NLPDemo();  
        nlpDemo.nlpDemo();  
    }  
}
```

## 响应示例

- 成功响应示例

```
{  
  "summary": "华为刀片式基站解决方案是华为在深入理解客户诉求基础上，引领业界的创新解决方案。该方案采用统一的模块化设计，实现基站主要元素如射频、基带、电源、电池、微波、传输的刀片化，不同模块间的任意快速拼装组合无缝拼装，能灵活安装在抱杆，铁塔，墙面或者屋顶，“0”站址无需机房机柜，使基站的安装像拼装乐高积木一样简单便捷。刀片式基站同时支持2G/3G/4G等多制式，在多频多模网络发展策略下可以高效利用宝贵的站点资源，大幅降低站点获取难度和减少站点租金，帮助运营商应对移动网络快速增长的容量需求。该奖项授予在技术研究、技术开发、技术创新、推广应用先进科学技术成果、促进高新技术产业化，以及完成重大科学技术工程、计划等过程中做出创造性贡献的中国公民和组织。”  
}
```

- 失败响应示例

```
{  
  "error_code": "NLP.3104",  
  "error_msg": "content is empty."  
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.2.3 诗歌生成

### 功能介绍

根据用户的输入生成诗歌。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API以**定制版API定价**按需计费，**不支持使用套餐包**，使用时请在[自然语言处理价格计算器](#) 按需计费-自然语言处理定制版API中查看费用详情。

本API调用限制为2次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
POST /v1/{project\_id}/nlg/poem

- 参数说明

表 4-57 URI 参数说明

名称	是否必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如下所示。

表 4-58 请求参数

名称	是否必选	参数类型	说明
type	是	Integer	诗歌类型，取值如下： <ul style="list-style-type: none"> <li>• 0：五言绝句</li> <li>• 1：七言绝句</li> <li>• 2：五言律诗</li> <li>• 3：七言律诗</li> </ul>
title	是	String	诗歌标题，目前仅支持UTF-8编码，仅支持中文，长度为1-10。
acrostic	否	boolean	藏头诗，取值如下。 <ul style="list-style-type: none"> <li>• true：为藏头诗</li> <li>• false：为非藏头诗</li> </ul> 默认取值为false。

## 响应消息

响应参数如[表4-59](#)所示。

表 4-59 响应参数

名称	参数类型	说明
poem	Array[String]	根据文本请求体，返回生成的诗歌。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

## 请求示例

- 请求示例（生成标题为春色，类型为七言律诗的藏头诗）

POST https://{endpoint}/v1/{project\_id}/nlg/poem

Request Header:

Content-Type: application/json

X-Auth-Token:

MIINRwYJKoZihvcNAQcCoIIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBgkqhkiG...  
//用户认证获取的token值，获取方法详细请参考“如何调用API”章节中的“认证鉴权”页面。

Request Body:

```
{
  "type": 3,
  "title": "春色",
  "acrostic": true
}
```

- Python3语言请求代码示例（生成标题为春色，类型为七言律诗的藏头诗）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlg/poem' # endpoint和project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'type': 3,
        'title': '春色',
        'acrostic': True
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例（生成标题为夏日，类型为五言绝句的藏头诗）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlg/poem");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String title = "夏日";
            String body = "{\"title\":\""+title+"\",\"type\": 0,\"acrostic\":true}";
```

```
        OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
        osw.append(body);
        osw.flush();
        InputStream is = connection.getInputStream();
        BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
        while (br.ready()) {
            System.out.println(br.readLine());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "poem": [
    "春色",
    "春风吹梦到湖山，陡觉欣荣触眼看。",
    "色与梅花相避远，名随竹叶共阑干。",
    "一年好处几人赏，半夜诗成独倚栏。",
    "老去自知才思尽，只将华发镜中弹。"
  ]
}
```

- 失败响应示例

```
{
  "error_code": "NLP.3104",
  "error_msg": "poem_type is null or empty."
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.3 语言理解服务接口说明

### 4.3.1 情感分析（基础版）

#### 功能介绍

针对通用领域的用户评论进行情感分析。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[基础套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

## 调试

您可以在[API Explorer](#)中调试该接口。

## 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

### 📖 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlu/sentiment
- 参数说明

表 4-60 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-61](#)所示。

表 4-61 请求参数

参数名	参数类型	必选	说明
content	String	是	待分析文本。文本编码要求为utf-8。 限定180个字符以内，文本长度超过180个字符时，只检测前180个字符。
lang	String	否	支持的文本语言类型，目前支持中文（zh）和英文（en），默认为中文。

## 响应消息

响应参数如[表4-62](#)所示。

表 4-62 响应参数

参数名	参数类型	说明
result	Result object	调用成功时的返回情感信息。 调用失败时无此字段。 请参见表4-63。
error_code	String	调用失败时的错误码。具体参见错误码。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-63 result 字段数据结构说明

参数名	参数类型	说明
content	String	待分析文本。
label	Integer	正负标签。 <ul style="list-style-type: none"> <li>1: positive</li> <li>0: negative</li> </ul>
confidence	Float	标签label的置信度。小数点精确到（6）位。

## 请求示例

- 请求示例（分析情感）

POST https://{endpoint}/v1/{project\_id}/nlu/sentiment

Request Header:

Content-Type: application/json

X-Auth-

Token:MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBgkqhkiG...

Request Body:

```
{
  "content": "浑浑噩噩的头脑、失魂落魄的身体..."
}
```

- Python3语言请求代码示例（分析用户评论为“浑浑噩噩的头脑、失魂落魄的身体...”的情感）

```
# -*- coding: utf-8 -*-
```

```
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
```

```
import json
```

```
def nlp_demo():
```

```
    url = 'https://{endpoint}/v1/{project_id}/nlu/sentiment' # endpoint和project_id需替换
```

```
    token = '用户对应region的token'
```

```
    header = {
```

```
        'Content-Type': 'application/json',
```

```
        'X-Auth-Token': token
```

```
    }
```

```
    body = {
```

```
        'content': '浑浑噩噩的头脑、失魂落魄的身体...'
```

```
}  
resp = requests.post(url, data=json.dumps(body), headers=header)  
print(resp.json())  
  
if __name__ == '__main__':  
    nlp_demo()
```

- Java语言请求代码示例（分析用户评论为“你好，房先生，我喜欢这款车。”的情感）

```
import java.io.BufferedReader;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStreamWriter;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
/**  
 * 此demo仅供测试使用，建议使用sdk  
 */  
public class NLPDemo {  
    public void nlpDemo() {  
        try {  
            //endpoint和projectId需要替换成实际信息。  
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlu/sentiment");  
            String token = "对应region的token";  
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
            connection.setRequestMethod("POST");  
            connection.setDoInput(true);  
            connection.setDoOutput(true);  
            connection.addRequestProperty("Content-Type", "application/json");  
            connection.addRequestProperty("X-Auth-Token", token);  
  
            //输入参数  
            String content = "你好，房先生，我喜欢这款车。";  
            String body = "{\"content\":\"" + content + "\"}";  
  
            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),  
"UTF-8");  
            osw.append(body);  
            osw.flush();  
            InputStream is = connection.getInputStream();  
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));  
            while (br.ready()) {  
                System.out.println(br.readLine());  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
  
    public static void main(String[] args) {  
        NLPDemo nlpDemo = new NLPDemo();  
        nlpDemo.nlpDemo();  
    }  
}
```

## 响应示例

- 成功响应示例

```
{  
  "result": {  
    "content": "浑浑噩噩的头脑、失魂落魄的身体…",  
    "label": 0,  
    "confidence": 0.90706205  
  }  
}
```

- 失败响应示例

```
"error_code": "NLP.0101",  
"error_msg": "Authentication failed. Please verify the token"  
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.3.2 情感分析（领域版）

### 功能介绍

领域情感分析，针对电商，汽车领域，自适应领域（通用、电商、汽车）的用户评论进行情感分析。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[领域套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
POST /v1/{project\_id}/nlu/sentiment/domain
- 参数说明

表 4-64 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如表4-65所示。

表 4-65 请求参数

参数名	参数类型	必选	说明
content	String	是	待分析文本。文本编码要求为utf-8，目前仅支持中文。 <ul style="list-style-type: none"> <li>type为0（自适应领域），1（电商领域评论）时，限定200个字符以内，文本长度超过200个字符时，只检测前200个字符。</li> <li>type为2（汽车领域评论）时，限定400个字符以内，文本长度超过400个字符时，只检测前400个字符。</li> </ul>
type	Integer	否	取值如下，默认值为0： 0：自适应领域，根据输入内容自动识别适应领域。 1：电商领域，适用于电商领域评论。 2：汽车领域，适用于汽车领域评论。

## 响应消息

响应参数如表4-66所示。

表 4-66 响应参数

参数名	参数类型	说明
result	Result object	调用成功时的返回情感信息。 调用失败时无此字段。 请参见表4-67。
error_code	String	调用失败时的错误码。具体参见错误码。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-67 result 字段数据结构说明

参数名	参数类型	说明
content	String	待分析文本。

参数名	参数类型	说明
label	Integer	该文本的分析结果标签，取值如下： <ul style="list-style-type: none"> <li>1: positive</li> <li>0: negative</li> </ul>
confidence	Float	标签label的置信度。小数点精确到（6）位。

## 请求示例

- 请求示例（分析汽车领域用户评论为“很不错的一款车，可以改变生活方式。”的情感）

```
POST https://{endpoint}/v1/{project_id}/nlu/sentiment/domain

Request Header:
Content-Type: application/json
X-Auth-Token:MIINRwYJKoZlhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:
{
  "content": "很不错的一款车，可以改变生活方式。",
  "type": 2
}
```
- Python3语言请求代码示例（分析汽车领域用户评论为“浑浑噩噩的头脑、失魂落魄的身体...”的情感）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlu/sentiment/domain' # endpoint和project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'content': '浑浑噩噩的头脑、失魂落魄的身体...'
        'type': 2
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```
- Java语言请求代码示例（分析自适应领域用户评论为“你好，房先生，我喜欢这款车。”的情感）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
```

```
try {
    //endpoint和projectId需要替换成实际信息。
    URL url = new URL("https://{endpoint}/v1/{project_id}/nlu/sentiment/domain");
    String token = "对应region的token";
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    connection.setRequestMethod("POST");
    connection.setDoInput(true);
    connection.setDoOutput(true);
    connection.addRequestProperty("Content-Type", "application/json");
    connection.addRequestProperty("X-Auth-Token", token);

    //输入参数
    String content = "你好，房先生，我喜欢这款车。";
    String body = "{\"content\":\"" + content + "\", \"type\":\"0\"}";

    OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
    osw.append(body);
    osw.flush();
    InputStream is = connection.getInputStream();
    BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
    while (br.ready()) {
        System.out.println(br.readLine());
    }
} catch (Exception e) {
    e.printStackTrace();
}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
```

## 响应示例

- 成功响应示例

```
{
  "result": {
    "content": "很不错的一款车，可以改变生活方式。",
    "label": 1,
    "confidence": 0.96399385
  }
}
```

- 失败响应示例

```
{
  "error_code": "NLP.0101",
  "error_msg": "Authentication failed. Please verify the token"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.3.3 文本分类

### 功能介绍

针对广告领域的自动分类，判断是否是广告。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[基础套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

使用前请进入NLP服务控制台开通本API。

本API调用限制为20次/秒。

## 调试

您可以在[API Explorer](#)中调试该接口。

## 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlu/classification
- 参数说明

表 4-68 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-69](#)所示。

表 4-69 请求参数

参数名	参数类型	必选	说明
content	String	是	待分析文本，UTF-8编码，限定400个字符以内，文本长度超过400个字符时，只检测前400个字符，目前仅支持中文。
domain	Integer	否	文本分类适用领域。默认为1，表示广告检测。目前只支持广告检测。

## 响应消息

响应参数如表4-70所示。

表 4-70 响应参数

参数名	参数类型	说明
result	Result object	调用成功时的返回情感信息。 调用失败时无此字段。 请参见表4-71。
error_code	String	调用失败时的错误码。具体参见错误码。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-71 result 字段数据结构说明

参数名	参数类型	说明
content	String	待分析文本。
label	Integer	正负分类标签。 <ul style="list-style-type: none"> <li>1: 广告</li> <li>0: 非广告</li> </ul>
confidence	Float	标签label的置信度。

## 请求示例

- 请求示例（判断“XXX去屑洗发水，全国包邮”是否是广告）

POST https://{endpoint}/v1/{project\_id}/nlu/classification

Request Header:

Content-Type: application/json

X-Auth-

Token:MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:

```
{
  "content": "XXX去屑洗发水，全国包邮",
  "domain": 1
}
```

- Python3语言请求代码示例（判断“XXX去屑洗发水，全国包邮”是否是广告）

```
# -*- coding: utf-8 -*-
```

```
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
```

```
import json
```

```
def nlp_demo():
```

```
    url = 'https://{endpoint}/v1/{project_id}/nlu/classification' # endpoint和project_id需替换
```

```
    token = '用户对应region的token'
```

```
header = {
    'Content-Type': 'application/json',
    'X-Auth-Token': token
}
body = {
    'content': 'XXX去屑洗发水, 全国包邮',
    'domain': 1
}
resp = requests.post(url, data=json.dumps(body), headers=header)
print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例（判断“XXX去屑洗发水，全国包邮”是否是广告）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlu/classification");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String content = "XXX去屑洗发水, 全国包邮";
            String body = "{\"content\":\"" + content + "\", \"domain\":1}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        NLPDemo nlpDemo = new NLPDemo();
        nlpDemo.nlpDemo();
    }
}
```

## 响应示例

- 成功响应示例

```
{
  "result": {
    "content": "XXX去屑洗发水, 全国包邮",
    "label": 1,
```

```
    "confidence": 0.74237967  
  }  
}
```

- 失败响应示例

```
{  
  "error_code": "NLP.0101",  
  "error_msg": "Authentication failed. Please verify the token"  
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.3.4 属性级情感分析（领域版）

### 功能介绍

属性级情感分析（领域版），本产品适用于评论文本的属性级正负面分析，编码方式 UTF-8。建议对数据预处理，对于文本为空的内容进行过滤，文本长度不超过1000字。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[领域套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
POST /v1/{project\_id}/nlu/aspect-sentiment
- 参数说明

表 4-72 URI 参数说明

名称	是否必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-73](#)所示。

表 4-73 请求参数

名称	是否必选	参数类型	说明
content	是	String	待分析文本。仅支持中文，文本编码要求为utf-8。建议文本长度1000个字符以内。
type	是	integer	情感分析适用领域，取值如下： “1”：手机领域

## 响应消息

响应参数如[表4-74](#)所示。

表 4-74 响应参数

名称	参数类型	说明
text	string	待分析文本。
label	integer	该文本的整体情感标签，取值如下： <ul style="list-style-type: none"> <li>“0”：负向</li> <li>“1”：正向</li> </ul>
confidence	float	该文本整体情感label的置信度，小数点精确到3位。
aspect_opinions	Array of aspectOpinions	属性情感挖掘列表。 请参见 <a href="#">表4-75</a> 。
error_code	string	调用失败时的错误码。具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	string	调用失败时的错误信息。 调用成功时无此字段。

表 4-75 aspectOpinion 字段数据结构说明

名称	参数类型	说明
aspect_category	String	属性类别。 目前支持属性类别如下。 “手机领域”：【'整体','性价比','赠品','分期','配件','活动','品牌','物流派送','包装','游戏性能','系统性能','芯片','屏幕','电池','自拍','拍照','音质','散热','防水','信号','解锁','外观设计','握持手感','质感','颜色','内存/容量','客服/售后','其他'】。
label	integer	文本关于属性类别的情感标签，取值如下： <ul style="list-style-type: none"> <li>“0”：负向</li> <li>“1”：正向</li> </ul>
confidence	float	属性类别情感标签对应的置信度（备用，当前的置信度粒度较粗）。
aspect_term	String	属性词，与对应的描述词至少出现其中之一，可能为null。
opinion_term	String	描述词，与对应的属性词至少出现其中之一，可能为null。
tag	string	属性-描述词片段所对应的标签。如果分类为'其他'，则不给出标签，返回null。
span	integer列表	共4个数字，分别表示属性词和描述词在文本中的起始位置和结束位置。 如果属性词为null，则1, 2两位不展示；如果描述词为null，则3, 4位不展示。

## 请求示例

- 请求示例（分析手机领域用户评论为“运行很快，快充也不错。”的属性级情感）

```
POST https://nlp-ext.cn-north-4.myhuaweicloud.com/v1/{{project_id}}/nlu/aspect-sentiment
Request Header:
Content-Type: application/json
X-Auth-Token:MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBGkqhkiG...
Request Body:
{
  "content":"运行很快，快充也不错。",
  "type":1
}
```

- Python3语言请求代码示例（分析手机领域用户评论为“运行很快，快充也不错。”的属性级情感）

```
# *- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json

def nlp_demo():
```

```
url = 'https://{endpoint}/v1/{project_id}/nlu/aspect-sentiment' # endpoint和project_id需替换
token = '用户对应region的token'
header = {
    'Content-Type': 'application/json',
    'X-Auth-Token': token
}
body = {
    'content': '运行很快，快充也不错。',
    'type': 1
}
resp = requests.post(url, data=json.dumps(body), headers=header)
print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例（分析手机领域用户评论为“运行很快，快充也不错。”的属性级情感）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlu/aspect-sentiment");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String content = "运行很快，快充也不错。";
            String body = "{\"content\":\"" + content + "\", \"type\":1}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        NLPDemo nlpDemo = new NLPDemo();
        nlpDemo.nlpDemo();
    }
}
```

## 响应示例

- 成功响应示例

```
{
  "aspect_opinions": [
    {
      "aspect_category": "系统性能",
      "aspect_term": "运行",
      "confidence": 1.0,
      "label": 1,
      "opinion_term": "很快",
      "span": [
        0,
        1,
        2,
        3
      ],
      "tag": "系统强大"
    },
    {
      "aspect_category": "电池",
      "aspect_term": "快充",
      "confidence": 1.0,
      "label": 1,
      "opinion_term": "不错",
      "span": [
        5,
        6,
        8,
        9
      ],
      "tag": "充电快"
    }
  ],
  "confidence": 0.955,
  "label": 1,
  "text": "运行很快，快充也不错。"
}
```

- 失败响应示例

```
{
  "error_code": "NLP0301",
  "error_msg": "type must be 1 now"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.3.5 属性级情感分析（高级版）

### 功能介绍

属性情感分析高级版，本产品适用于评论文本的属性级正负面分析，文本长度不超过4096字，编码方式UTF-8。建议对数据预处理，对于文本为空的内容进行过滤。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API以**定制版API定价**按需计费，**不支持使用套餐包**，使用时请在[自然语言处理价格计算器](#) 按需计费-自然语言处理定制版API中查看费用详情。

本API调用限制为20次/秒。

## 调试

您可以在[API Explorer](#)中调试该接口。

## 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

### 📖 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlu/aspect-sentiment/advance
- 参数说明

表 4-76 URI 参数说明

名称	是否必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-77](#)所示。

表 4-77 请求参数

名称	是否必选	参数类型	说明
content	是	String	待分析文本。文本编码要求为utf-8，仅支持中文。限定4096个字符以内，建议文本长度300个字符以内。
type	是	integer	情感分析适用领域，取值如下。 <ul style="list-style-type: none"><li>• “1”：手机领域</li><li>• “2”：汽车领域</li></ul>

## 响应消息

响应参数如[表4-78](#)所示。

表 4-78 响应参数

名称	参数类型	说明
content	String	待分析文本。
label	integer	该文本的整体情感标签，取值如下。 <ul style="list-style-type: none"> <li>“0”：负向</li> <li>“1”：正向</li> </ul>
confidence	float	该文本整体情感label的置信度，小数点精确到3位。
aspect_opinions	Array of aspectOpinions	属性情感挖掘列表。 请参见表4-79。
error_code	string	调用失败时的错误码。具体参见错误码。 调用成功时无此字段。
error_msg	string	调用失败时的错误信息。 调用成功时无此字段。

表 4-79 aspectOpinion 字段数据结构说明

名称	参数类型	说明
aspect_category	String	属性类别。 目前支持属性类别： <ul style="list-style-type: none"> <li>“汽车领域”：【'动力','外观','内饰','空间','操控','舒适性','性价比','能耗'】</li> <li>“手机领域”：【'整体','内存','外形设计','屏幕','性价比','拍照','散热','电池','信号','指纹识别','音质','握持手感','活动配件赠品','防水','客服','物流派送','包装'】</li> </ul>
label	integer	文本关于属性类别的情感标签。 <ul style="list-style-type: none"> <li>“0”：负向</li> <li>“1”：正向</li> </ul>
confidence	float	属性类别情感标签对应的置信度。
aspect_term	String	属性描述词，预留参数，暂不支持。
opinion_term	String	观点描述词，预留参数，暂不支持。
tag	string	观点标签，预留参数，暂不支持。
span	integer列表	包含一个或多个描述属性类别的片段的起始位置和终止位置，预留参数，暂不支持。

## 请求示例

- 请求示例（分析汽车领域用户评论的属性级情感）

```
POST https://nlp-ext.cn-north-4.myhuaweicloud.com/v1/{project_id}/nlu/aspect-sentiment/advance
Request Header:
Content-Type: application/json
X-Auth-Token:MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBGkqhkiG...
Request Body:
{
  "content":"1、2档动力很肉，会抖。2、2档难挂，最怕半坡等红绿灯熄火。3、方向盘打死会有很大的异响，倒车最明显。4、踩油门很硬，而且踩下去很难走动，必须要用力大脚踩进去才把转速提上来。5、后备箱不能遥控开启。6、内饰接缝很是不整齐，而且用料很差。7、车内的音响设备太落后了，中控的cd机既然还不带usb播放功能。8、钥匙不好看太小气。9、锁车不会自动升车窗。先喷这么多，以后想起来再喷。",
  "type":2
}
```

- Python3语言请求代码示例（分析手机领域用户评论为“运行很快，快充也不错。”的属性级情感）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlu/aspect-sentiment/advance' # endpoint和project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'content': '运行很快，快充也不错。',
        'type': 1
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例（分析手机领域用户评论为“运行很快，快充也不错。”的属性级情感）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlu/aspect-sentiment/advance");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
```

```
String content = "运行很快, 快充也不错。";
String body = "{\"content\": \"" + content + "\", \"type\": 1}";

OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
osw.append(body);
osw.flush();
InputStream is = connection.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
while (br.ready()) {
    System.out.println(br.readLine());
}
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "content": "1、2档动力很肉, 会抖。2、2档难挂, 最怕半坡等红绿灯熄火。3、方向盘打死会有很大的异响, 倒车最明显。4、踩油门很硬, 而且踩下去很难走动, 必须要用力大脚踩进去才把转速提上来。5、后备箱不能遥控开启。6、内饰接缝很是不整齐, 而且用料很差。7、车内的音响设备太落后了, 中控的cd机既然还不带usb播放功能。8、钥匙不好看太小气。9、锁车不会自动升车窗。先喷这么多, 以后想起来再喷。",
  "label": 0,
  "confidence": 0.997282,
  "aspect_opinions": [
    {
      "aspect_category": "动力",
      "label": 0,
      "confidence": 0.988097
    },
    {
      "aspect_category": "内饰",
      "label": 0,
      "confidence": 0.988097
    },
    {
      "aspect_category": "操控",
      "label": 0,
      "confidence": 0.989935
    },
    {
      "aspect_category": "舒适性",
      "label": 0,
      "confidence": 0.964776
    }
  ]
}
```

- 失败响应示例

```
{
  "error_code": "NLPF.0301",
  "error_msg": "argument valid error:content.must not be null;content.must not be blank;"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

### 4.3.6 实体级情感分析

#### 功能介绍

实体级情感分析，本接口用于检测指定实体（entity）在文本（content）中的正负面分析，适用于金融方面公司实体正负面新闻的分析。

文本与实体长度的和不超过512字，编码方式UTF-8。建议对数据预处理，对于文本或实体为空的内容进行过滤。如果文本或实体为空，接口不支持。

本API免费调用，调用限制为2次/秒。

#### 调试

您可以在[API Explorer](#)中调试该接口。

#### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

##### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlu/entity-sentiment
- 参数说明

表 4-80 URI 参数说明

名称	是否必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

#### 请求消息

请求参数如[表4-81](#)所示。

表 4-81 请求参数

名称	是否必选	参数类型	说明
content	是	String	请求文本，用于检查指定实体（entity）在该请求文本中的正负面分析。 仅支持中文，文本编码要求为 utf-8。请求文本内容不能为空，且请求文本与请求实体的长度和不超过512字。长度超过512时，只检测前512个字符。
entity	是	String	请求实体，用于检查该请求实体在文本（content）中的正负面分析。 <b>说明</b> 不同实体（entity）在同一文本（content）的正负面分析会存在不同的分析结果。 仅支持中文，文本编码要求为 utf-8。请求实体内容不能为空，且请求文本与请求实体的长度和不超过512字。长度超过512时，只检测前512个字符。
type	是	integer	实体级情感分析适用领域，取值如下。 “3”：金融领域

## 响应消息

响应参数如表4-82所示。

表 4-82 响应参数

名称	参数类型	说明
confidence	double	该实体在文本中的情感label的置信度
content	String	响应文本。
entity	String	响应实体。
label	integer	该实体在文本中的情感标签，0表示负面，1表示非负面，2表示不相关。
error_code	string	调用失败时的错误码。具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	string	调用失败时的错误信息。 调用成功时无此字段。

## 请求示例

- 请求示例（检测金融领域中云端金融在杭州云端金融信息服务有限公司涉嫌非法吸取公众存款案例的正负面分析）

```
POST https://nlp-ext.cn-north-4.myhuaweicloud.com/v1/{project_id}/nlu/entity-sentiment
Request Header:
Content-Type: application/json
X-Auth-Token:MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgggVBGkqhkiG...
Request Body:
{
  "content": "杭州云端金融信息服务有限公司涉嫌非法吸取公众存款案例",
  "entity": "云端金融",
  "type": 3
}
```
- Python3语言请求代码示例（检测金融领域中云端金融在杭州云端金融信息服务有限公司涉嫌非法吸取公众存款案例的正负面分析）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlu/entity-sentiment' # endpoint和project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'content': '杭州云端金融信息服务有限公司涉嫌非法吸取公众存款案例',
        'entity': '云端金融',
        'type': 3
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```
- Java语言请求代码示例（检测金融领域中云端金融在杭州云端金融信息服务有限公司涉嫌非法吸取公众存款案例的正负面分析）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlu/entity-sentiment");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String content = "杭州云端金融信息服务有限公司涉嫌非法吸取公众存款案例";
            String entity = "云端金融";
```

```
String body = "{ \"content\": \"" + content + "\" , \"entity\": \"" + entity + "\" , \"type\": 3}";

OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
osw.append(body);
osw.flush();
InputStream is = connection.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
while (br.ready()) {
    System.out.println(br.readLine());
}
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "confidence": 0.999,
  "content": "杭州云端金融信息服务有限公司涉嫌非法吸取公众存款案例",
  "entity": "云端金融",
  "label": 0
}
```

- 失败响应示例

```
{
  "error_code": "NLP0301",
  "error_msg": "entity must not null"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.3.7 意图理解

### 功能介绍

针对天气类、报时、新闻类、笑话类、翻译类、提醒类、闹钟类、音乐类8个领域进行意图理解。意图理解包括对用户的问题，陈述进行领域识别以及对对应领域所包含的实体进行抽取。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[基础套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

## 调试

您可以在[API Explorer](#)中调试该接口。

## 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/nlu/semantic-parser
- 参数说明

表 4-83 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-84](#)所示。

表 4-84 请求参数

参数名	参数类型	必选	说明
text	String	是	待分析文本列表，UTF-8编码，限定32个字符以内，文本长度超过32个字符时，只检测前32个字符。
lang	String	否	支持的文本语言类型，目前只支持中文（zh），默认为中文。

## 响应消息

表 4-85 响应参数

参数名	参数类型	说明
result	Result object	调用成功时的返回意图信息。 调用失败时无此字段。 请参见表4-86。
error_code	String	调用失败时的错误码。具体参见错误码。 调用成功时无此字段。
error_message	String	调用失败时的错误信息。 调用成功时无此字段。

表 4-86 result 字段数据结构说明

参数名	参数类型	说明
text	String	返回待分析文本。
label	String	待分析文本的意图标签。标签共有以下9类： weather: 天气, time: 报时, news: 新闻, joke: 笑话, translation: 翻译, notification: 提醒, alarm: 闹钟, music: 音乐, others: 其它。
confidence	Float	标签label的置信度。
slots	Array of slot	slot数据结构, 请参见表4-87。

表 4-87 slot 字段数据结构说明

参数名	参数类型	说明
word	String	实体文本。

参数名	参数类型	说明
tag	String	实体类型。对于每个意图类别所支持的实体类型分别为： weather: date(日期), time(时间), location(位置) time: location(位置), timezone(时区) news: genre(风格) joke: genre(风格) translation: content(内容) notification: content(内容), date(日期), time(时间), singer(歌手) alarm: date(日期), time:(时间) music: singer(歌手), song(歌曲), content(内容)
offset	Integer	实体文本在待分析文本中的起始位置。
length	Integer	实体文本长度。
normalized_word	String	同义词或者其他标准表达的词，默认为原始的word。

## 请求示例

- 请求示例（意图理解）

POST https://{endpoint}/v1/{project\_id}/nlu/semantic-parser

Request Header:

Content-Type: application/json

X-Auth-

Token:MIINRwYJKoZihvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:

```
{
  "text": "来一首周杰伦的青花瓷",
  "lang": "zh"
}
```

- Python3语言请求代码示例（意图理解）

```
# -*- coding: utf-8 -*-
```

```
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
```

```
import json
```

```
def nlp_demo():
```

```
    url = 'https://{endpoint}/v1/{project_id}/nlu/semantic-parser' # endpoint和project_id需替换
```

```
    token = '用户对应region的token'
```

```
    header = {
```

```
        'Content-Type': 'application/json',
```

```
        'X-Auth-Token': token
```

```
    }
```

```
    body = {
```

```
        'text': '来一首周杰伦的青花瓷',
```

```
        'lang': 'zh'
```

```
    }
```

```
    resp = requests.post(url, data=json.dumps(body), headers=header)
```

```
    print(resp.json())
```

```
if __name__ == '__main__':  
    nlp_demo()
```

- Java语言请求代码示例（意图理解）

```
import java.io.BufferedReader;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStreamWriter;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
/**  
 * 此demo仅供测试使用，建议使用sdk  
 */  
public class NLPDemo {  
    public void nlpDemo() {  
        try {  
            //endpoint和projectId需要替换成实际信息。  
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlu/semantic-parser");  
            String token = "对应region的token";  
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
            connection.setRequestMethod("POST");  
            connection.setDoInput(true);  
            connection.setDoOutput(true);  
            connection.addRequestProperty("Content-Type", "application/json");  
            connection.addRequestProperty("X-Auth-Token", token);  
  
            //输入参数  
            String text = "来一首周杰伦的青花瓷";  
            String body = "{\"text\":\"" + text + "\",\"lang\":\"zh\"}";  
  
            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),  
"UTF-8");  
            osw.append(body);  
            osw.flush();  
            InputStream is = connection.getInputStream();  
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));  
            while (br.ready()) {  
                System.out.println(br.readLine());  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
  
    public static void main(String[] args) {  
        NLPDemo nlpDemo = new NLPDemo();  
        nlpDemo.nlpDemo();  
    }  
}
```

## 响应示例

- 成功响应示例

```
{  
  "result": {  
    "confidence": 1,  
    "label": "music",  
    "slots": [  
      {  
        "length": 3,  
        "normalized_word": "周杰伦",  
        "offset": 3,  
        "tag": "singer",  
        "word": "周杰伦"  
      },  
      {  
        "length": 3,
```

```
    "normalized_word": "青花瓷",  
    "offset": 7,  
    "tag": "song",  
    "word": "青花瓷"  
  }  
],  
"text": "来一首周杰伦的青花瓷"  
}
```

- 失败响应示例

```
{  
  "error_code": "NLP.0101",  
  "error_msg": "Authentication failed. Please verify the token"  
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.3.8 文档分类

### 功能介绍

文档分类接口对用户输入的文本自动分类，给文本具体的分类。用户只要提供待处理的文本，而不用关注具体实现。主要应用场景：新闻内容分类，广告识别等。

具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[基础套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
POST /v1/{project\_id}/nlu/doc-classification
- 参数说明

表 4-88 URI 参数说明

名称	是否必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数如[表4-89](#)所示。

表 4-89 请求参数

名称	是否必选	参数类型	说明
content	是	String	输入的文档，最大长度10000，长度超过10000字符截取前10000个字符。
lang	否	String	预留字段，支持的文本语言类型，当前只支持zh（中文），默认zh。

## 响应消息

响应参数如[表4-90](#)所示。

表 4-90 响应参数

名称	参数类型	说明
result	Array of Result object	调用成功时的返回标签列表。 调用失败时无此字段。 请参见 <a href="#">表4-91</a> 。
error_code	string	调用失败时的错误码。具体参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	string	调用失败时的错误信息。 调用成功时无此字段。

表 4-91 result 字段数据结构说明

参数名	参数类型	说明
label	String	输入的文档标签, 包括 sport (体育), anime (动漫), hospital (医院), history (历史), advertising (广告), entertainment (娱乐), religion (宗教), novel (小说), estate (地产), recruitment (招聘), education (教育), tourism (旅游), automobile (汽车), game (游戏), technology (科技), joke (笑话), health (健康), gourmet (美食), stock (股票), parenting (育儿), pornography (色情), finance (金融)
confidence	Float	标签label的置信度。0到1之间的数。

## 请求示例

- 请求示例 (文档分类)

POST https://nlp-ext.cn-north-4.myhuaweicloud.com/v1/{project\_id}/nlu/doc-classification

Request Header:

Content-Type: application/json

X-Auth-

Token:MIINRwYJKoZlhvcNAQcCoIIODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:

```
{
  "content": "中芯国际季报超预期, 带了波芯片股节奏, 不过, 现在的市场风险偏好, 不足以掀起普涨行情, 芯片股也是如此, 核心标的走慢牛, 少数猛股做主升浪, 大多数要死不活。前段时间, 芯片的最强分支是材料, 最近几天, 最强分支是封测, 猛股则是光刻胶龙头$容大感光(SZ300576)$与存储芯片封测龙头$深科技(SZ000021)$。原因很简单, 因为它们在芯片大跌时, 形态保持的最好, 之前也没爆炒过, 加上是新面孔。深科技创下了20年新高, 与14年的深圳华强很像, 可能是条潜在的大鱼, 未来几个月, 需要密切关注基本面的变化。封测炒一大波后, 可能会转向设计, 龙头韦尔随时可能新高, 而猛股则可能是$紫光国微(SZ002049)$, 即将历史新高主升浪。不管大盘是向上突破年线, 还是向下二次探底, 二季度拥抱核心科技都是最佳策略, 即使大盘调整, 核心科技的调整压力, 也会明显低于大消费与农业。//@似水年华:回复@似水年华:大盘调整期间, 走势抗跌的标的, 在大盘止跌后, 往往会成为大牛股。上轮芯片股大跌时, $容大感光(SZ300576)$深科技(SZ000021)$紫光国微(SZ002049)$跌幅最小、形态完好, 在大盘见底、芯片见底后, 也基本成为了最强的几只标的。形态最好的容大, 率先历史新高, 成为了新的材料龙头; 形态次之的深科技, 今天也创下了历史新高, 将成为新的封测龙头; 形态再次之的紫光国微, 即将历史新高, 可能成为新的设计龙头。查看对话",
  "lang": "zh"
}
```

- Python3语言请求代码示例 (文档分类)

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用, 建议使用sdk。需提前安装requests, 执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/nlu/doc-classification' # endpoint和project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'content': '中芯国际季报超预期, 带了波芯片股节奏, 不过, 现在的市场风险偏好, 不足以掀起普涨行情, 芯片股也是如此, 核心标的走慢牛, 少数猛股做主升浪, 大多数要死不活。前段时间, 芯片的最强分支是材料, 最近几天, 最强分支是封测, 猛股则是光刻胶龙头$容大感光(SZ300576)$与存储芯片封测龙头$深科技(SZ000021)$。原因很简单, 因为它们在芯片大跌时, 形态保持的最好, 之前也没爆炒过, 加上是新面孔。深科技创下了20年新高, 与14年的深圳华强很像, 可能是条潜在的大鱼, 未来几个月, 需要密切关注基本面的变化。封测炒一大波后, 可能会转向设计, 龙头韦尔随时可能新高, 而猛股则可能是$紫光'
    }
```

国微(SZ002049)\$, 即将历史新高主升浪。不管大盘是向上突破年线, 还是向下二次探底, 二季度拥抱核心科技都是最佳策略, 即使大盘调整, 核心科技的调整压力, 也会明显低于大消费与农业。//@似水年华:回复@似水年华:大盘调整期间, 走势抗跌的标的, 在大盘止跌后, 往往会成为大牛股。上轮芯片股大跌时, \$光大感光(SZ300576)\$深科技(SZ000021)\$紫光国微(SZ002049)\$跌幅最小、形态完好, 在大盘见底、芯片见底后, 也基本成为了最强的几只标的。形态最好的容大, 率先历史新高, 成为了新的材料龙头; 形态次之的深科技, 今天也创下了历史新高, 将成为新的封测龙头; 形态再次之的紫光国微, 即将历史新高, 可能成为新的设计龙头。查看对话',

```
'lang': 'zh'
}
resp = requests.post(url, data=json.dumps(body), headers=header)
print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例 (文档分类)

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用, 建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/nlu/doc-classification");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String content = "中芯国际季报超预期, 带了波芯片股节奏, 不过, 现在的市场风险偏好, 不足以掀起普涨行情, 芯片股也是如此, 核心标的走慢牛, 少数猛股做主升浪, 大多数要死不活。";
            String body = "{\"content\":\"" + content + "\",\"lang\":\"zh\"}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        NLPDemo nlpDemo = new NLPDemo();
        nlpDemo.nlpDemo();
    }
}
```

## 响应示例

- 成功响应示例

```
{
  "result": [
```

```
{
  "confidence": 1,
  "label": "stock"
}
```

- 失败响应示例

```
{
  "error_code": "NLP.0101",
  "error_msg": "Authentication failed. Please verify the token"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

# 4.4 机器翻译服务接口说明

## 4.4.1 文本翻译

### 功能介绍

文本翻译是为了实现语种间的转换。对于用户输入原始语种的文本，转换为目标语种的文本。本接口不支持对文档进行翻译，文档翻译请见[文档翻译任务创建](#)。

#### 📖 说明

文本翻译支持一次性提交多个任务，服务会根据资源排队处理。

文本翻译接口具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API按**字符数**阶梯计费，文本翻译**不支持**套餐包抵扣，用户可以在[自然语言处理价格计算器](#)价格详情页了解具体计费说明。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 📖 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/machine-translation/text-translation
- 参数说明

表 4-92 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数说明请参见[表4-93](#)。

表 4-93 请求参数说明

参数名	参数类型	必选	说明
text	String	是	待翻译文本，仅支持utf-8编码，长度不超过2000字符。一个汉字、英文字母、标点符号等，均计为一个字符。
from	String	是	翻译原语言，具体取值见 <a href="#">表4-94</a> 。
to	String	是	翻译目标语言，具体取值见 <a href="#">表4-94</a> 。
scene	String	否	默认为“common”，当前只有通用场景。

表 4-94 文本翻译支持的语言列表

语种名称	语言代码
阿拉伯语	ar
德语	de
俄语	ru
法语	fr
韩语	ko
葡萄牙语	pt
日语	ja
泰语	th

语种名称	语言代码
土耳其语	tr
西班牙语	es
英语	en
越南语	vi
中文（简体）	zh
中文（繁体）	zh-tw
自动检测输入语种并翻译成目标语种，您需要指定目标语种。	auto

## 响应消息

响应参数说明请参见[表4-95](#)。

**表 4-95** 响应参数说明

参数名	参数类型	说明
src_text	String	调用成功时表示翻译原文，编码格式为UTF-8。 调用失败时无此字段。
translated_text	String	调用成功时表示翻译译文，编码格式为UTF-8。 调用失败时无此字段。
from	String	调用成功时表示源语种（源语种输入为auto时，显示语种识别结果），编码格式为UTF-8。 调用失败时无此字段。
to	String	调用成功时表示目标语种，编码格式为UTF-8。 调用失败时无此字段。
error_code	String	调用失败时的错误码，具体请参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

## 请求示例

- 请求示例（文本翻译）

POST https://{endpoint}/v1/{project\_id}/machine-translation/text-translation

Request Header:

Content-Type:application/json

X-Auth-Token:

MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgguVBgkqhkiG...

```
Request Body:
{
  "text": "欢迎使用机器翻译服务",
  "from": "zh",
  "to": "en",
  "scene": "common"
}
```

- Python3语言请求代码示例（翻译中文"欢迎使用机器翻译服务"为英文）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/machine-translation/text-translation' # endpoint和
    project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'text': '欢迎使用机器翻译服务',
        'from': 'zh',
        'to': 'en',
        'scene': 'common'
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例（翻译英文"It is a good day"为中文）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/machine-translation/text-translation");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            String text = "It is a good day";
            String from = "en";
            String to = "zh";
            String body = "{\"text\":\"" + text + "\",\"from\":\"" + from + "\",\"to\":\"" + to + "\",\"scene\"
            :\"common\"}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
            "UTF-8");
            osw.append(body);
            osw.flush();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
InputStream is = connection.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
while (br.ready()) {
    System.out.println(br.readLine());
}
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "src_text": "欢迎使用机器翻译服务",
  "translated_text": "Welcome to use machine translation services",
  "from": "zh",
  "to": "en"
}
```

- 失败响应示例

```
{
  "error_code": "NLP0101",
  "error_msg": "Authentication failed. Verify the token."
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.4.2 语种识别

### 功能介绍

语种识别是为了识别文本所属的语种。对于用户输入的文本，返回识别出的所属语种。

语种识别接口具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API支持使用[基础套餐包](#)，购买时请在[自然语言处理价格计算器](#)中查看基础套餐包和领域套餐包支持的API范围。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

## 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/machine-translation/language-detection
- 参数说明

表 4-96 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数说明请参见[表4-97](#)。

表 4-97 请求参数说明

参数名	参数类型	必选	说明
text	String	是	需要识别语种的文本，具体支持的语种请参见 <a href="#">表3</a> 。 仅支持utf-8编码，长度不超过2000字符。一个汉字、英文字母、标点符号等，均计为一个字符。

表 4-98 语种识别支持的语言列表

语种名称	语言代码
阿拉伯语	ar
爱沙尼亚语	et
保加利亚语	bg
冰岛语	is
波兰语	pl
波斯尼亚语	bs

语种名称	语言代码
波斯语	fa
丹麦语	da
德语	de
俄语	ru
法语	fr
芬兰语	fi
高棉语	km
韩语	ko
加泰罗尼亚语	ca
捷克语	cs
克罗地亚语	hr
拉脱维亚语	lv
立陶宛语	lt
罗马尼亚语	ro
马耳他语	mt
马来西亚语	ms
北马其顿语	mk
孟加拉语	bn
缅甸语	my
南非荷兰语	af
挪威语	no
葡萄牙语	pt
日语	ja
瑞典语	sv
塞尔维亚语	sr
斯洛伐克语	sk
斯洛文尼亚语	sl
斯瓦希里语	sw
泰语	th
土耳其语	tr

语种名称	语言代码
威尔士语	cy
乌尔都语	ur
乌克兰语	uk
西班牙语	es
希伯来语	he
希腊语	el
匈牙利语	hu
意大利语	it
印地语	hi
印尼语	id
英语	en
越南语	vi
中文	zh
无法识别语种	unk

## 响应消息

响应参数说明请参见[表4-99](#)。

**表 4-99** 响应参数说明

参数名	参数类型	说明
detected_language	String	调用成功时表示调用结果，编码格式为UTF-8，语种取值请见 <a href="#">表4-98</a> 。 当输入文本过短或不明确时，识别结果可能不准确； 当输入文本包含多种语言时，会返回占比最高的语种。 调用失败时无此字段。
error_code	String	调用失败时的错误码，具体请参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

## 请求示例

- 请求示例（识别"欢迎使用机器翻译服务"的语种）

POST `https://{endpoint}/v1/{project_id}/machine-translation/language-detection`

Request Header:

Content-Type:application/json

X-Auth-Token:

MIINRwYJKoZIHvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:

```
{
  "text": "欢迎使用机器翻译服务"
}
```

- Python3语言请求代码示例（识别'欢迎使用机器翻译服务'的语种）

```
# -*- coding: utf-8 -*-
```

```
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
```

```
import json
```

```
def nlp_demo():
```

```
    url = 'https://{endpoint}/v1/{project_id}/machine-translation/language-detection' # endpoint和  
    project_id需替换
```

```
    token = '用户对应region的token'
```

```
    header = {
```

```
        'Content-Type': 'application/json',
```

```
        'X-Auth-Token': token
```

```
    }
```

```
    body = {
```

```
        'text': '欢迎使用机器翻译服务'
```

```
    }
```

```
    resp = requests.post(url, data=json.dumps(body), headers=header)
```

```
    print(resp.json())
```

```
if __name__ == '__main__':
```

```
    nlp_demo()
```

- Java语言请求代码示例（识别"It is a good day"的语种）

```
import java.io.BufferedReader;
```

```
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
```

```
import java.io.OutputStreamWriter;
```

```
import java.net.HttpURLConnection;
```

```
import java.net.URL;
```

```
/**
```

```
 * 此demo仅供测试使用，建议使用sdk
```

```
 */
```

```
public class NLPDemo {
```

```
    public void nlpDemo() {
```

```
        try {
```

```
            //endpoint和projectId需要替换成实际信息。
```

```
            URL url = new URL("https://{endpoint}/v1/{project_id}/machine-translation/language-  
detection");
```

```
            String token = "对应region的token";
```

```
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
```

```
            connection.setRequestMethod("POST");
```

```
            connection.setDoInput(true);
```

```
            connection.setDoOutput(true);
```

```
            connection.addRequestProperty("Content-Type", "application/json");
```

```
            connection.addRequestProperty("X-Auth-Token", token);
```

```
            //输入参数
```

```
            String text = "It is a good day";
```

```
            String body = "{\"text\":\"" + text + "\"}";
```

```
            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),  
"UTF-8");
```

```
            osw.append(body);
```

```
osw.flush();
InputStream is = connection.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
while (br.ready()) {
    System.out.println(br.readLine());
}
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "detected_language": "zh"
}
```

- 失败响应示例

```
{
  "error_code": "NLP0101",
  "error_msg": "Authentication failed. Verify the token."
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[错误码](#)。

## 4.4.3 文档翻译任务创建

### 功能介绍

由于文档翻译会需要较长的时间，因此翻译是异步的，也即接口分为创建翻译任务和查询任务状态两个接口。

创建翻译任务接口创建任务完成后返回，然后用户通过调用查询任务状态接口来获得翻译状态和临时URL。用户可以使用临时URL下载翻译好的文件，每个临时URL有效期为10分钟。翻译结果会保存24小时（从翻译完成的时间算起）。24小时后如果再访问，将会返回“task id is not found”错误。

文档翻译任务创建接口用于提交文档翻译任务，其中要翻译的文档保存在用户的OBS桶中。用户使用文档翻译服务时，服务需要拥有读取用户OBS桶权限，授权方法见[配置OBS访问权限](#)。当前仅支持翻译“docx”、“pptx”和“txt”格式的文档，其中txt格式文档只支持UTF-8编码格式。

文档翻译任务创建接口具体Endpoint请参见[终端节点](#)。

调用华为云NLP服务会产生费用，本API按[文档页数](#)阶梯计费，文本翻译[不支持](#)套餐包抵扣，用户可以在[自然语言处理价格计算器](#)价格详情页了解具体计费说明。

本API调用限制为2次/秒。

## 调试

您可以在[API Explorer](#)中调试该接口。

## 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

### 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

## URI

- URI格式  
POST /v1/{project\_id}/machine-translation/file-translation/jobs
- 参数说明

表 4-100 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。

## 请求消息

请求参数说明请参见[表4-101](#)。

表 4-101 请求参数说明

参数名	参数类型	必选	说明
url	String	是	存放在OBS的文档文件路径，私密文件推荐使用临时授权URL调用服务，如何获取OBS文件URL和临时授权URL请参见 <a href="#">配置OBS访问权限</a> 。OBS的region要和请求服务的region保持一致，region不一致则OBS不可用，即使obs是公开访问权限。存放在OBS的文档文件名必须是英文字母。
from	String	是	翻译原语言，文档翻译服务当前仅支持中英互译。
to	String	是	翻译目标语言，文档翻译服务当前仅支持中英互译。

参数名	参数类型	必选	说明
type	String	是	文档格式，当前仅支持翻译“docx”、“pptx”和“txt”格式的文档，其中txt格式文档只支持UTF-8编码格式。

表 4-102 支持的语言列表

语言 ( from )	语言 ( to )	说明
zh	en	中文翻译为英文。
en	zh	英文翻译为中文。

## 响应消息

响应参数说明请参见[表 响应参数说明](#)。

表 4-103 响应参数说明

参数名	参数类型	说明
job_id	String	创建的任务标识，如果创建任务成功时必存在。创建失败时无此参数。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此参数。
error_msg	String	调用失败时的错误信息。 调用成功时无此参数。

## 请求示例

- 请求示例（创建文档翻译任务）

```
POST https://{endpoint}/v1/{project_id}/machine-translation/file-translation/jobs
Request Header:
Content-Type:application/json
X-Auth-Token:
MIINRwYJKoZlhfvcNAQcCoIINODCCDTQCAQExDTALBgIghkgBZQMEAgEwgggVBgkqhkiG...

Request Body:
{
  "url": "https://****.obs.cn-north-4.huawei.com/***.docx",
  "from": "zh",
  "to": "en",
  "type": "docx"
}
```
- Python3语言请求代码示例（创建文档翻译任务）

```
# -*- coding: utf-8 -*-
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
import json

def nlp_demo():
    url = 'https://{endpoint}/v1/{project_id}/machine-translation/file-translation/jobs' # endpoint和
    project_id需替换
    token = '用户对应region的token'
    header = {
        'Content-Type': 'application/json',
        'X-Auth-Token': token
    }
    body = {
        'url': 'https://****.obs.cn-north-4.huawei.com/****.docx',
        'from': 'zh',
        'to': 'en',
        'type': 'docx'
    }
    resp = requests.post(url, data=json.dumps(body), headers=header)
    print(resp.json())

if __name__ == '__main__':
    nlp_demo()
```

- Java语言请求代码示例（创建文档翻译任务）

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

/**
 * 此demo仅供测试使用，建议使用sdk
 */
public class NLPDemo {
    public void nlpDemo() {
        try {
            //endpoint和projectId需要替换成实际信息。
            URL url = new URL("https://{endpoint}/v1/{project_id}/machine-translation/
jobs");
            String token = "对应region的token";
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);
            connection.addRequestProperty("Content-Type", "application/json");
            connection.addRequestProperty("X-Auth-Token", token);

            //输入参数
            //mtUrl请按照API说明，将文档上传至OBS后获取对应URL地址
            String mtUrl = "https://*.obs.cn-north-4.myhuaweicloud.com/*.docx";
            String from = "en";
            String to = "zh";
            String body = "{\"url\":\"" + mtUrl + "\", \"from\":\"" + from + "\", \"to\":\"" + to + "\", \"type
\": \"docx\"}";

            OutputStreamWriter osw = new OutputStreamWriter(connection.getOutputStream(),
"UTF-8");
            osw.append(body);
            osw.flush();
            InputStream is = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));
            while (br.ready()) {
                System.out.println(br.readLine());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
public static void main(String[] args) {
    NLPDemo nlpDemo = new NLPDemo();
    nlpDemo.nlpDemo();
}
}
```

## 响应示例

- 成功响应示例

```
{
  "job_id": "567e6536-****-****-****-826321939656"
}
```

- 失败响应示例

```
{
  "error_code": "NLP.0101",
  "error_msg": "Authentication failed. Verify the token."
}
```

## 4.4.4 文档翻译状态查询

### 功能介绍

该接口用于获取文档翻译识别状态以及临时url，临时url可以用于获取翻译后的文档，每个临时url有效期为十分钟。

本API调用限制为20次/秒。

### 调试

您可以在[API Explorer](#)中调试该接口。

### 前提条件

在使用本API之前，需要您完成服务申请和认证鉴权，具体操作流程请参见[申请服务](#)和[认证鉴权](#)章节。

#### 📖 说明

用户首次使用需要先[申请开通](#)。服务只需要开通一次即可，后面使用时无需再次申请。如未开通服务，调用服务时会提示ModelArts.4204报错，请在调用服务前先进入控制台开通服务，并注意开通服务区域与调用服务的区域保持一致。

### URI

- URI格式  
GET /v1/{project\_id}/machine-translation/file-translation/jobs/{job\_id}
- 参数说明

表 4-104 URI 参数说明

参数名	必选	说明
project_id	是	项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。
job_id	是	文档翻译任务标识符。通过 <a href="#">文档翻译任务创建</a> 获取。

## 请求消息

NA

## 响应消息

响应参数说明请参见[表 响应参数说明](#)。

**表 4-105** 响应参数说明

参数名	参数类型	说明
status	String	当前翻译状态。具体状态如下所示： <ul style="list-style-type: none"> <li>• WAITING 等待翻译。</li> <li>• BEGIN 开始翻译。</li> <li>• FINISHED 翻译已经完成。</li> <li>• ERROR 翻译过程中发生错误。</li> </ul> 调用失败时无此字段。
url	String	临时url，使用GET请求下载翻译结果，有效期十分钟。过期后请再次调用接口获取新的url。 调用失败时或翻译状态非FINISHED时无此字段。
error_code	String	调用失败时的错误码，具体参见 <a href="#">错误码</a> 。 调用成功时无此参数。
error_msg	String	调用失败时的错误信息。 调用成功时无此参数。

## 请求示例

- 请求示例（查询文档翻译状态）

```
GET https://{endpoint}/v1/{project_id}/machine-translation/file-translation/567e6536-****-****-****-826321939656
```

Request Header:

Content-Type:application/json

X-Auth-Token:

MIINRwYJKoZIHvcNAQcCoIIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

- Python3语言请求代码示例（查询文档翻译状态）

```
# -*- coding: utf-8 -*-
```

```
# 此demo仅供测试使用，建议使用sdk。需提前安装requests，执行pip install requests
```

```
import requests
```

```
def nlp_demo():
```

```
# endpoint和project_id需替换，job_id需要替换成[文档翻译任务创建]API返回的job_id信息。
```

```
url = 'https://{endpoint}/v1/{project_id}/machine-translation/file-translation/jobs/{job_id}'
```

```
token = '用户对应region的token'
```

```
header = {
```

```
    'Content-Type': 'application/json',
```

```
    'X-Auth-Token': token
```

```
}
```

```
resp = requests.get(url, headers=header, verify=False)
```

```
print(resp.json())
```

```
if __name__ == '__main__':  
    nlp_demo()
```

- Java语言请求代码示例（查询文档翻译状态）

```
import java.io.BufferedReader;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStreamWriter;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
/**  
 * 此demo仅供测试使用，建议使用sdk  
 */  
public class NLPDemo {  
    public void nlpDemo() {  
        try {  
            //endpoint和projectId需要替换成实际信息。  
            //job_id需要替换成[文档翻译任务创建]API返回的job_id信息。  
            URL url = new URL("https://{endpoint}/v1/{project_id}/machine-translation/file-translation/  
jobs/{job_id}");  
            String token = "对应region的token";  
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
            connection.setRequestMethod("GET");  
            connection.setDoInput(true);  
            connection.setDoOutput(true);  
            connection.addRequestProperty("Content-Type", "application/json");  
            connection.addRequestProperty("X-Auth-Token", token);  
  
            InputStream is = connection.getInputStream();  
            BufferedReader br = new BufferedReader(new InputStreamReader(is, "UTF-8"));  
            while (br.ready()) {  
                System.out.println(br.readLine());  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
  
    public static void main(String[] args) {  
        NLPDemo nlpDemo = new NLPDemo();  
        nlpDemo.nlpDemo();  
    }  
}
```

## 响应示例

- 成功响应示例

```
{  
  "status": "FINISHED",  
  "url": "https://****.obs.****.huawei.com:443/****?AccessKeyId=****&Expires=****&Signature=*****!"  
}
```

- 失败响应示例

```
{  
  "error_code": "NLP0101",  
  "error_msg": "Authentication failed. Verify the token."  
}
```

# 5 数据结构

## 5.1 公共请求参数

表 5-1 公共请求消息头

参数名	说明	必选	示例
Content-type	发送的实体的 MIME 类型。	是	application/json
Content-Length	请求 body 长度，单位为 Byte。	POST/PUT 请求为可选，GET 不包含。	3495
X-Auth-Token	用户 Token。	是	MIINRwYJKoZlhvc NAQcCoIINODCC DTQCAQExDTALB glghkgBZQMEAgE wgguVBgkqhkiG...
X-Language	请求语言类型。	否，默认为 zh-cn。	en-us

### 说明

其它 header 属性，请遵照 https 协议。

## 5.2 公共响应参数

表 5-2 公共响应消息头

参数名	说明
Content-Length	响应消息体的字节长度，单位为 Byte。

参数名	说明
Date	系统响应的时间。
Content-type	发送的实体的MIME类型。

# 6 附录

## 6.1 状态码

- 正常  
200
- 异常

返回值	说明
400 Bad Request	请求错误，具体返回错误码请参考 <a href="#">错误码</a> 。
401 Unauthorized	鉴权失败。
403 Forbidden	没有操作权限。
500 Internal Server Error	服务内部错误。
503 Service Unavailable	服务不可用。

## 6.2 错误码

调用接口出错后，将不会返回结果数据。调用方可根据每个接口对应的错误码来定位错误原因。当调用出错时，HTTPS请求返回一个4xx或5xx的HTTPS状态码。返回的消息体中是具体的错误代码及错误信息。

### 错误响应 Body 体格式说明

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

```
{  
  "error_msg": "The format of message is error",  
  "error_code": "AS.0001"  
}
```

其中，error\_code表示错误码，error\_msg表示错误描述信息。

## 错误码说明

自然语言处理错误码如表6-1所示，ModelArts的错误码请参考[错误码](#)。

当您调用API时，如果遇到“APIGW”开头的错误码，请参见[API网关错误码](#)进行处理。

表 6-1 错误码

状态码	错误码	错误信息	描述	处理措施
401	NLP.000	user has no privilege of beta!	用户不具备公测权限。	请根据界面提示，申请公测权限。
401	NLP.0101	Authentication failed. Please verify the token	认证失败。	检查Token是否正确，使用正确的Token重新尝试。 获取Token方式请见 <a href="#">API Explorer获取Token</a> 和 <a href="#">Postman获取Token</a> 。
401	NLP.0102	authentication info missing!	认证信息缺失。	补全正确的认证信息后重新尝试。
401	NLP.0103	get ops token failed	认证信息获取失败。	检查Token是否正确，使用正确的Token重新尝试。 获取Token方式请见 <a href="#">API Explorer获取Token</a> 和 <a href="#">Postman获取Token</a> 。
400	NLP.0301	query param error	请求参数异常。	检查请求参数字段名，修改正确后重新尝试。
400	NLP.0303	request method not support	http请求方式不支持。	检查请求方式，修改正确后重新尝试。
400	NLP.0304	http message not readable	http消息无法读取。	检查请求消息格式，修改正确后重新尝试。
400	NLP.0305	http media type is not supported	http媒体方式不支持。	检查请求媒体方式，修改正确后重新尝试。
500	NLP.0306	Http request timed out.	http请求超时。	服务后台错误，请联系技术支持。
500	NLP.0201	inner service exception	内部服务异常。	服务后台错误。您可以在 <a href="#">API Explorer</a> 中查看代码示例，调试接口。
500	NLP.0401	obsClient is null	OBSClient为空。	服务后台错误，请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
500	NLP.0 402	obs closed failed	OBSClient关闭失败。	服务后台错误，请联系技术支持。
500	NLP.0 403	upload file for obs failed	OBS上传文件失败。	服务后台错误，请联系技术支持。
500	NLP.0 405	download file for obs failed	OBS下载文件失败。	服务后台错误，请联系技术支持。
500	NLP.1 101	request deliver failed	服务下发失败。	服务后台错误，请联系技术支持。
500	NLP.1 102	response body parameter failed	服务返回参数解析出错。	服务后台错误，请联系技术支持。
500	NLP.1 201	purchase failed	购买套餐包失败。	服务后台错误，请联系技术支持。
400	NLP.1 202	purchase disabled	不可购买。	该产品暂时不允许购买，请尝试购买其他产品。
400	NLP.1 203	tenant is frozen!	用户账号欠费。	用户账号欠费，请充值后重新使用或购买。 充值请登录华为云官网，单击右上角的“账号名>费用与成本”，在“资金管理>充值”页面为自己的账号充值，确保当前账号未欠费。
400	NLP.1 204	the resource of the tenant is frozen	租户该资源被冻结	用户资源被冻结，请检查是否违规或者欠费。
400	NLP.1 205	can not repeat buy	不可重复购买	该资源不允许重复购买。
400	NLP.1 206	User is not verified	用户未实名	用户未实名，请实名制后重新使用或者购买。 请登录华为云官网，单击右上角的“账号中心>费用中心”，在左侧导航栏选择“实名认证”，进行实名认证。
400	NLP.1 207	this user have no permission	无购买权限	无购买权限，请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	NLP.3 201	illegal parameter.	参数错误。	服务后台错误，请联系技术支持。
500	NLP.3 202	template parse error.	模板解析错误。	服务后台错误，请联系技术支持。
500	NLP.3 203	formula calculation error	公式计算错误。	服务后台错误，请联系技术支持。
500	NLP.3 204	the template is not exist.	模板资源不存在。	服务后台错误，请联系技术支持。
500	NLP.4 101	w2v file is null	文件为空或者不存在。	服务后台错误，请联系技术支持。
500	NLP.4 201	NLPU Inner service exception	NLPU内部服务异常。	服务后台错误，请联系技术支持。
500	NLP.4 302	Response parameters cannot be resolved	内部数据解析错误。	服务后台错误，请联系技术支持。
500	NLP.6 101	Server inner error.	机器翻译服务内部异常。	请联系服务管理员。
400	NLP.6 302	Reaching concurrency limitation, please try it later.	达到流控上限。	请稍后重试。
400	NLP.6 303	Obtaining the file from the provided url failed.	无法从OBS桶获取文件。	请检查OBS桶是否为公共读状态。
400	NLP.6 304	Task is not found!	对应任务不存在。	请检查是否输入错误的job-id，或任务已经过期。
400	NLP.6 305	删除	不支持的文档格式。	请检查type字段指定的文档格式是否正确。
400	NLP.6 306	Empty file error.	无效文件。	请检查文件是否为空或文件大小为0byte。
400	NLP.6 307	File size is over the limit.	文档页数超过上限。	请减少单次请求的文档页数。当前页数上限为250页。
500	NLP.6 308	Task is overtime.	任务超时。	请重试。

状态码	错误码	错误信息	描述	处理措施
403	Model Arts.0 203	Invalid token.	无效的token.	重新生成一个token进行重试。 获取Token方式请见 <a href="#">API Explorer获取Token</a> 和 <a href="#">Postman获取Token</a> 。

## 6.3 获取项目 ID

在调用接口的时候，部分URL中需要填入项目ID，所以需要获取到项目ID。项目ID获取步骤如下：

### 从控制台获取项目 ID

1. 登录[管理控制台](#)。
2. 鼠标移动到右上角已登录的用户名上，在下拉列表中选择“我的凭证”。
3. 在“我的凭证”页面，可以查看用户名、账号名、项目ID，在项目列表中查看项目。

图 6-1 查看项目 ID



多项目时，展开“所属区域”，从“项目ID”列获取子项目ID。项目编码需与NLP服务终端节点保持一致，当访问华北-北京四的终端节点（即nlp-ext.cn-north-4.myhuaweicloud.com）时，项目对应需为cn-north-4。

### 调用 API 获取项目 ID

项目ID还可通过调用[查询指定条件下的项目信息](#)API获取。

获取项目ID的接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点，可以从[地区和终端节点](#)获取。接口的认证鉴权请参见[认证鉴权](#)。

响应示例如下，例如，自然语言处理服务部署的区域为“cn-north-4”，响应消息体中查找“name”为“cn-north-4”，其中projects下的“id”即为项目ID。

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "project_name",
      "description": "",
      "links": {
        "next": null,

```

```
    "previous": null,
    "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
  },
  "id": "a4a5d4098fb4474fa22cd05f897d6b99",
  "enabled": true
}
],
"links": {
  "next": null,
  "previous": null,
  "self": "https://www.example.com/v3/projects"
}
}
```

## 6.4 获取账号 ID

在调用接口的时候，部分URL中需要填入账号ID（domain-id），所以需要先在管理控制台上获取到账号ID。账号ID获取步骤如下：

1. 登录NLP控制台。
2. 单击用户名，在下拉列表中单击“我的凭证”。  
在“我的凭证”页面的项目列表中查看账号ID。

图 6-2 查看账号 ID



## 6.5 配置 OBS 访问权限

### OBS 服务

OBS全称Object Storage Service（对象存储服务），提供海量、安全、高可靠、低成本的数据存储能力，可供用户存储任意类型和大小数据。

EI企业智能服务对于图片、语音等多媒体文件支持直接使用华为云OBS服务的数据处理方式，以减少服务使用成本，降低服务的响应时长，提升服务使用的体验。

考虑到数据的安全，自然语言处理服务无法直接获取到用户数据，需要用户开启公共读授权。

### 开启公共读授权

开启公共授权，则数据全网可见，所有用户均可访问。一般私密数据不建议用此方法，数据隐私及过期时间请客户自行保证。

配置公共读可参考[配置标准桶策略](#)，将桶策略设置为“公共读”。

目前仅支持访问用户个人OBS下的文件的链接，不支持读取其他用户公共读的链接。

## 使用临时授权 URL 授权 NLP 服务访问

对于敏感信息，如个人私有数据，存在泄漏风险，建议使用临时授权URL替代OBS文件URL授权服务访问用户私有文件。

临时URL存在有效期，获取到临时URL后请及时使用。

如何上传文件并获取临时url可以参考以下代码。

```
String endPoint = "http://your-endpoint";
String ak = "*** Provide your Access Key ***";
String sk = "*** Provide your Secret Key ***";
// 创建ObsClient实例ObsClient
obsClient = new ObsClient(ak, sk, endPoint);
// URL有效期, 3600秒
longexpireSeconds = 3600L;
obsClient.putObject("bucketname", "objectname", new File("localfile"));
TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.GET,
expireSeconds);
request.setBucketName("bucketname");
request.setObjectKey("objectname");
// 获取临时url,临时授权URL格式如下:
// https://***.obs.cn-north-4.huawei.com/***.docx?AccessKeyId=***&Expires=***Signature=***
TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
System.out.println("\t" + response.getSignedUrl());
```

## 上传文件

可参考[上传对象](#)。

## 获取文件 URL

可参考[获取对象URL](#)。

# 7 修订记录

发布日期	修改说明
2024-01-17	修改示例代码
2021-02-26	删除 文本生成
2021-01-30	修改 <ul style="list-style-type: none"><li>• <a href="#">文本翻译</a></li></ul>
2020-08-03	新增 <ul style="list-style-type: none"><li>• <a href="#">事件抽取</a></li></ul>
2020-07-06	修改 <ul style="list-style-type: none"><li>• <a href="#">命名实体识别（领域版）</a></li></ul>
2019-12-13	新增 <ul style="list-style-type: none"><li>• <a href="#">诗歌生成</a></li><li>• 4.2.4-文本生成（即将下线）</li><li>• <a href="#">意图理解</a></li><li>• <a href="#">文档翻译任务创建</a></li><li>• <a href="#">文档翻译状态查询</a></li><li>• <a href="#">配置OBS访问权限</a></li></ul>
2019-08-05	新增 <ul style="list-style-type: none"><li>• <a href="#">命名实体识别（领域版）</a></li><li>• 4.3.4-意图理解（废弃）</li></ul> 删除 <ul style="list-style-type: none"><li>• 实体链接</li></ul>

2019-07-08	新增机器翻译内容： <ul style="list-style-type: none"><li>• <a href="#">使用前必读</a></li><li>• <a href="#">API概览</a></li><li>• <a href="#">机器翻译服务接口说明</a></li><li>• <a href="#">错误码</a></li></ul>
2019-06-28	刷新错误码。
2019-03-22	<ul style="list-style-type: none"><li>• 增加接口：<a href="#">文本相似度（基础版）句向量</a></li><li>• 老接口修改为废弃接口。<a href="#">句向量（废弃）</a></li></ul>
2019-01-30	第一次正式发布。