

uniapp 安卓文字转拼音原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
插件方法	2
汉字转拼音	2.1
是否是多音字	2.2
获取首字母	2.3
转换为繁体字	2.4
转换为简体字	2.5
判断是否是繁体字	2.6
判断是否包含中文	2.7

使用方法

介绍

安卓文字转拼音插件，支持转换为声调模式和非声调模式，支持繁体和简体互相转换

联系作者

关注微信公众号可联系作者



插件地址

<https://ext.dcloud.net.cn/plugin?id=14026>

用法

在需要使用插件的页面加载以下代码

```
const module = uni.requireNativePlugin("leven-pinyin-PinyinModule");
```

更新日志

2023-8-11

首次发布

汉字转拼音

方法名

toPinyin

用法

- 用法如下：

```
module.toPinyin({
  content: "这是一条需要转换的中文内容",
  separator: " ",
  format: 3
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
content	String	是	无	要转换的内容
separator	String	否	空格	分隔符
format	Integer	否	1	转换格式，1.不带拼音，2.数字代表拼音，3.带拼音

回调

- 示例

```
{
  "data": {
    "result": "zhè shì yī tiáo xū yào zhuǎn huàn de zhōng wén nèi róng"
  },
  "message": "转换完成",
}
```

```
    "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.result	String	转换后的数据
code	Integer	返回类型，0.成功

是否是多音字

方法名

hasMulti

用法

- 用法如下：

```
let charArr = ["重", "天"];
let index = uni.$lv.number.random(0, 1);
module.hasMulti({
  content: charArr[index]
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
content	String	是	无	要验证的内容

回调

- 示例

```
{
  "data": {
    "result": true
  },
  "message": "查询完成",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

是否是多音字

参数名	参数类型	参数描述
data	Object	数据对象
data.result	Boolean	转换后的数据
code	Integer	返回类型，0.成功

获取首字母

方法名

getShortPinyin

用法

- 用法如下：

```
module.getShortPinyin({
  content: "这是一条需要转换的中文内容",
  isUpper: true,
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
content	String	是	无	要转换的内容
isUpper	Boolean	否	false	是否转为大写

回调

- 示例

```
{
  "data": {
    "result": "ZSYTXYZHDZWNR"
  },
  "message": "查询完成",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

获取首字母

参数名	参数类型	参数描述
data	Object	数据对象
data.result	String	转换后的数据
code	Integer	返回类型，0.成功

转换为繁体字

方法名

toTraditional

用法

- 用法如下：

```
module.toTraditional({
  content: "这是一条需要转换的中文内容"
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
content	String	是	无	要转换的内容

回调

- 示例

```
{
  "data": {
    "result": "這是一條需要轉換的中文内容"
  },
  "message": "查询完成",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

转换为繁体字

参数名	参数类型	参数描述
data.result	String	转换后的数据
code	Integer	返回类型，0.成功

转换为简体字

方法名

toSimplified

用法

- 用法如下：

```
module.toSimplified({
  content: "我們的大中國啊，好大的一個家"
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
content	String	是	无	要转换的内容

回调

- 示例

```
{
  "data": {
    "result": "我们的大中国啊，好大的一个家"
  },
  "message": "查询完成",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

转换为简体字

参数名	参数类型	参数描述
data.result	String	转换后的数据
code	Integer	返回类型，0.成功

判断是否是繁体字

方法名

isTraditional

用法

- 用法如下：

```
let charArr = ["国", "國"];
let index = uni.$lv.number.random(0, 1);
module.isTraditional({
  content: charArr[index]
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
content	String	是	无	要验证的内容

回调

- 示例

```
{
  "data": {
    "result": true
  },
  "message": "查询完成",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

判断是否是繁体字

参数名	参数类型	参数描述
data	Object	数据对象
data.result	Boolean	转换后的数据
code	Integer	返回类型，0.成功

判断是否包含中文

方法名

containsChinese

用法

- 用法如下：

```
let charArr = ["this is test", "这里有中文, oh good!"];
let index = uni.$lv.number.random(0, 1);
module.containsChinese({
  content: charArr[index]
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
content	String	是	无	要验证的内容

回调

- 示例

```
{
  "data": {
    "result": true
  },
  "message": "查询完成",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

判断是否包含中文

参数名	参数类型	参数描述
data	Object	数据对象
data.result	Boolean	转换后的数据
code	Integer	返回类型，0.成功