

文件分类：研发文档
文件编号：SHD-【产品代号】-【序列号】
版本：V1.0

密级：☐ O 公开
☒ S 保密
☐ TS 机密

智能家居领域语言云编译 SDK

技术使用手册

共 14 页

文件审批：

	部门/职位	签名	日期
编制			
标审			
审核			
会签			
会签			
会签			
批准			

文件修改记录：

版本	修改内容概要	修改人	批准人	生效日期
V1.0	新建	尹佩佩		2023/2/6
V1.1	添加章节 4 部署环境	尹佩佩		2023/7/25

目 录

1 引用	1
1.1 jar 包复制到本项目	1
1.2 maven	1
2 接口定义	1
2.1 编译(syncCompile)	1
2.2 语句拆分(getTokenList)	7
3 demo 示例	14
4 部署环境	14

家居领域语言编译器云编译 SDK 技术使用手册

1 引用

1.1 jar 包复制到本项目

```
<dependency>
  <groupId>com.utyun</groupId>
  <artifactId>home-cloud-compiler-sdk</artifactId>
  <version>1.05.00</version>
  <scope>system</scope>
  <systemPath>>${project.basedir}/lib/home-cloud-compiler-sdk.jar</systemPath>
</dependency>
```

如果缺少依赖，请参考 jar 包中的\META-INF\maven\com.utyun\home-cloud-compiler-sdk\pom.xml 解决依赖问题。

1.2 maven

```
<dependency>
  <groupId>com.utyun</groupId>
  <artifactId>home-cloud-compiler-sdk</artifactId>
  <version>1.05.00</version>
</dependency>
```

2 接口定义

2.1 编译(syncCompile)

接口定义	定义说明
接口名称	同步编译

2.1.1 接口入参

参数名称	类型	是否必须	备注
appKey	String	是	标识调用方，应用 key
userData	String	是	base64 编码的用户文件 (变量、语句等)

projectData	Object	否	工程文件（语句配置、编译配置）
systemData	Object	是	系统数据

json 示例：

```
{
  "userData": "PDzln7rmnKzkv6Hmga/ooag+PuWQjeensHzmj4",
  "projectData": {
    "configInfo": {
      "basicConfig": {
        "U_CONFIG_MAX_LOGIC_SEG_BYTES": 1,
        "U_CONFIG_MAX_LOGIC_TRIGGERED": 2
      },
      "shConfig": {
        "SH_CONFIG_LOGOUT_LEVEL": 3,
        "SH_CONFIG_MAX_LOGIC_ULC": 4,
        "SH_COFNIG_LOGIC_TYPE": 5
      },
      "deviceTable": [
        {
          "deviceName": "客厅西开关",
          "nodeNo": 0,
          "nodeInnerProductNo": 1,
          "channelNo": 2,
          "hostNo": 3
        }
      ]
    },
  },
}
```

```
"ulcFileInfo": [  
  {  
    "ulcId": "0",  
    "logicNoList": [  
      1,  
      2,  
      3  
    ]  
  }  
]  
,  
"systemData": {  
  "GlossaryInfo": {  
    "GlossaryDefine": [  
      {  
        "category": 0,  
        "level": 0,  
        "logicConversionRules": "\\\"%s\\\",\\\"\\\"\\\",  
        "logicRegularMatch": "或",  
        "name": "或",  
        "tokenConversionRules": "\\\"%s\\\",getTokenList(sz[1],\"LogicGlossary\\\")",  
        "tokenRegularMatch": "(或)",  
        "type": "LogicGlossary"  
      }  
    ],  
    "LuaData": {  
      "data": "LS0qKioqKioqKioqKioqKioqQysr5Lqk5LqS55qE5YWz6ZSu5=",  
      "type": 1
```

```
    }  
  },  
  "ShInfo": {  
    "FunctionDefine": [  
      {  
        "description": "切换函数",  
        "funKey": "9",  
        "funName": "Switch",  
        "parasList": [  
          {  
            "dataType": "int32",  
            "parasDescription": ""  
          },  
          {  
            "dataType": "int32",  
            "parasDescription": ""  
          },  
          {  
            "dataType": "int32",  
            "parasDescription": ""  
          },  
          {  
            "dataType": "int32",  
            "parasDescription": ""  
          }  
        ]  
      }  
    ]  
  }  
}
```

```

    },
    {
      "dataType": "int32",
      "parasDescription": ""
    },
    {
      "dataType": "int32",
      "parasDescription": ""
    },
    {
      "dataType": "int32",
      "parasDescription": ""
    }
  ],
  "returnDataType": "bool"
}

],
"ProductAttributeDefine": [
  {
    "productAttributeInfo": [
      {
        "attributeName": "左继电器",
        "attributeType": 2,
        "dataType": "int8",
        "valueDescription": "{断开=0,导通=1}"
      }
    ]
  },
  {
    "productName": "一开智能控制面板（433）"
  }
]

```



```

    }
  ]
}
}
}

```

2.1.2 接口返参

参数名称	类型	是否必须	备注
compileName	String	是	编译名称
compileInfo	Object	否	编译结果信息：ulc 文件内容和对应的逻辑编号
compileLogs	Object	是	编译 log

Json 示例：

```

{
  "compileName": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "compileInfo": [{
    "ulcId": "1",
    "logicNoList": [
      1,
      2
    ],
    "ulcBase64Content": "AAAAAAAAAAAAAAAAAAAAAAAAAAAA"
  }],
  "compileLogs": [{
    "logicNo": -1,
    "code": 0,
    "module": -1,
    "level": -1,
    "rowNo": -1,
    "pos": -1,
    "message": "家居编译成功",
    "keywords": {}
  }]
}

```

}

2.2 语句拆分(getTokenList)

接口定义	定义说明
接口名称	语句的拆分

2.2.1 接口入参

参数名称	类型	是否必须	备注
appKey	String	是	标识调用方, 应用 key
userData	String	是	用户文件(变量、语句等)
projectData	Object	否	工程文件(语句配置、编译配置)
eventCondition	String	否	触发条件字符串
partCondition	String	否	参与条件字符串
trueResult	String	否	真结果
falseResult	String	否	假结果
systemData	Object	是	系统数据

Json 示例:

```
{
  "userData": "PDzln7rmnKzkv6Hmga/ooag+PuWQjeensHzmj4",
  "projectData": {
    "configInfo": {
      "basicConfig": {
        "U_CONFIG_MAX_LOGIC_SEG_BYTES": 1,
        "U_CONFIG_MAX_LOGIC_TRIGGERED": 2
      },
      "shConfig": {
        "SH_CONFIG_LOGOUT_LEVEL": 3,

```

```
"SH_CONFIG_MAX_LOGIC_ULC": 4,

"SH_COFNIG_LOGIC_TYPE": 5

},

"deviceTable": [{

    "deviceName": "客厅西开关",

    "nodeNo": 0,

    "nodeInnerProductNo": 1,

    "channelNo": 2,

    "hostNo": 3

}]

},

"ulcFileInfo": [{

    "ulcId": "0",

    "logicNoList": [

        1,

        2,

        3

    ]

}

},

"eventCondition": "qwe",

"partCondition": "asd",

"trueResult": "123",

"falseResult": "456",

"systemData": {

    "GlossaryInfo": {

        "GlossaryDefine": [{

            "category": 0,
```

```

    "level": 0,

    "logicConversionRules": "\"%s\\",\"|\"",

    "logicRegularMatch": "或",

    "name": "或",

    "tokenConversionRules": "\"%s\\",getTokenList(sz[1],\"LogicGlossary\")",

    "tokenRegularMatch": "(或)",

    "type": "LogicGlossary"

  }],

  "LuaData": {

    "data": "LS0qKioqKioqKioqKioqKioqQysr5Lqk5LqS55qE5YWz6ZSu5=",

    "type": 1

  }

},

"ShInfo": {

  "FunctionDefine": [{

    "description": "切换函数",

    "funKey": "9",

    "funName": "Switch",

    "parasList": [{

      "dataType": "int32",

      "parasDescription": ""

    },

    {

      "dataType": "int32",

      "parasDescription": ""

    },

    {

      "dataType": "int32",

```

-10- UL-T010

```

    "attributeType": 2,

    "dataType": "int8",

    "valueDescription": "{断开=0,导通=1}"

  }

],

"productName": "一开智能控制面板（433）"

}]

}

}

}

```

备注:

类型	描述
dataType	"bool","int8","int16","int32","int64","float","double","string"
	布尔类型
	Int8 类型
	Int16 类型
	Int32 类型
	Int64 类型
	单精度浮点型
	双精度浮点型
	字符串类型

举例如下:



newTokenlist.json

2.2.2 接口返参

参数名称	类型	是否必须	备注
shTokenInfo	Object	否	拆分信息
logs	Object	是	拆分日志

Json 示例:

```
{
```

```

    "shTokenInfo":
    "{
      \"eventCondition\": {
        \"Pos\": 0,
        \"Children\": [
          {
            \"Pos\": 0,
            \"Children\": [],
            \"TokenType\": \"Variable\",
            \"SrcValue\": \"客厅开关_上键\"
          },
          {
            \"Pos\": 7,
            \"Children\": [],
            \"TokenType\": \"RelayGlossary\",
            \"SrcValue\": \"不等于\"
          },
          {
            \"Pos\": 10,
            \"Children\": [],
            \"TokenType\": \"Const\",
            \"SrcValue\": \"1\"
          }
        ],
        \"TokenType\": \"Expression\",
        \"SrcValue\": \"客厅开关_上键不等于1\"
      },
      \"partCondition\": {
        \"trueResult\": {
          \"Pos\": 11,
          \"Children\": [
            {
              \"Pos\": 11,
              \"Children\": [],
              \"TokenType\": \"Variable\",
              \"SrcValue\": \"客厅开关_上键\"
            },
            {
              \"Pos\": 18,
              \"Children\": [],
              \"TokenType\": \"RelayGlossary\",
              \"SrcValue\": \"等于\"
            },
            {
              \"Pos\": 20,
              \"Children\": [],
              \"TokenType\": \"Const\",
              \"SrcValue\": \"2\"
            }
          ],
          \"TokenType\": \"Expression\",
          \"SrcValue\": \"客厅开关_上键等于 2\"
        },
        \"falseResult\": {}
      }
    }
  ]
}

```

备注：

返回标准语句的拆分字段信息

字段	含义
SrcValue	所有子节点的 srcvalue 组合
Pos	所有子节点中取最小 pos, 没有子节点则取当前节点 pos
TokenType	“Variable”, “Const”, “Operator”, “Function”, “Identifier”, “Expression”

	变量类型	"Variable"
	常量类型	"Const"
	操作符类型	"Operator"
	函数类型	"Function"
	未知类型	"Identifer"
	表达式类型	"Expression"

举例:



newTokenlist.json

返回术语语句的拆分字段信息

字段	含义																																								
Src Value	所有子节点的 srcvalue 组合																																								
Pos	所有子节点中取最小 pos,没有子节点则取当前节点 pos																																								
TokenType	<table border="1"> <tr> <th colspan="2">术语类型</th></tr> <tr> <td>类型</td><td>类型 (字符串)</td></tr> <tr> <td>未知类型</td><td>Identifer</td></tr> <tr> <td>变量</td><td>Variable</td></tr> <tr> <td>常量</td><td>Const</td></tr> <tr> <td>延时术语</td><td>DelayGlossary</td></tr> <tr> <td>取消延时术语</td><td>CancelDelayGlossary</td></tr> <tr> <td>时间点术语</td><td>TimeDotGlossary</td></tr> <tr> <td>时间段术语</td><td>TimeRangeGlossary</td></tr> <tr> <td>星期术语</td><td>WeekGlossary</td></tr> <tr> <td>节假日术语</td><td>HolidayGlossary</td></tr> <tr> <td>灯亮术语</td><td>LightOnGlossary</td></tr> <tr> <td>灯灭术语</td><td>LightOffGlossary</td></tr> <tr> <td>常量值术语</td><td>ConstGlossary</td></tr> <tr> <td>逻辑运算符术语</td><td>LogicGlossary</td></tr> <tr> <td>关系运算符术语</td><td>RelayGlossary</td></tr> <tr> <td>动作分隔符术语</td><td>ActionGlossary</td></tr> <tr> <td>场景术语</td><td>ScenceGlossary</td></tr> <tr> <td>置反术语</td><td>ReverseGlossary</td></tr> <tr> <td>表达式</td><td>Expression</td></tr> </table>	术语类型		类型	类型 (字符串)	未知类型	Identifer	变量	Variable	常量	Const	延时术语	DelayGlossary	取消延时术语	CancelDelayGlossary	时间点术语	TimeDotGlossary	时间段术语	TimeRangeGlossary	星期术语	WeekGlossary	节假日术语	HolidayGlossary	灯亮术语	LightOnGlossary	灯灭术语	LightOffGlossary	常量值术语	ConstGlossary	逻辑运算符术语	LogicGlossary	关系运算符术语	RelayGlossary	动作分隔符术语	ActionGlossary	场景术语	ScenceGlossary	置反术语	ReverseGlossary	表达式	Expression
术语类型																																									
类型	类型 (字符串)																																								
未知类型	Identifer																																								
变量	Variable																																								
常量	Const																																								
延时术语	DelayGlossary																																								
取消延时术语	CancelDelayGlossary																																								
时间点术语	TimeDotGlossary																																								
时间段术语	TimeRangeGlossary																																								
星期术语	WeekGlossary																																								
节假日术语	HolidayGlossary																																								
灯亮术语	LightOnGlossary																																								
灯灭术语	LightOffGlossary																																								
常量值术语	ConstGlossary																																								
逻辑运算符术语	LogicGlossary																																								
关系运算符术语	RelayGlossary																																								
动作分隔符术语	ActionGlossary																																								
场景术语	ScenceGlossary																																								
置反术语	ReverseGlossary																																								
表达式	Expression																																								
children	子节点																																								

举例:



getTokenList-Glossary.json

3 demo 示例

家居云编译 sdk 调用 demo 示例详见以下附件。



home-compiler-sdk-demo.rar

4 部署环境

由于家居编译器内核动态库是在 ubuntu16.04.12 系统下进行编译生成的, 因此运行环境在 ubuntu16.04.12 系统及以上(包括 ubuntu16.04.12)基本是可以的, alpine 等其它操作系统都是不行的。