

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

技术使用手册

共 14 页

文件审批：

	部门/职位	签名	日期
编制	软件研发部	谢东晓	2021/12/09
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			
V1.1	getTokenList 返回数据结构不一致，更新文档		李宇哲	2021/12/16
V1.2	接口路径、入参、返回数据结构修改		谢东晓	2022/04/26
V1.3	更新为生产环境的路径		谢东晓	2022/07/14
V1.4	接口入参、返回数据结构修改		尹佩佩	2023/1/12

目 录

1 概述	1
2 设计概述	1
2.1 设计简述	1
2.2 系统结构设计	1
2.3 主要业务流程图	2
2.4 约束和假定	3
3 接口定义	3
3.1 业务管理-编译管理	3
4 对接调用流程	17
4.1 注册账号	17
4.2 获取接口调用凭证 access-token	17
4.3 使用 access-token 调用接口	18

家居领域语言编译器云编译技术使用手册

1 概述

领域语言是在标准语言的基础上进行衍生扩展,形成的行业、领域里特有的方言(规则)。由用户语言基本语法或者家居规范定义的术语构成。它将用户语言的描述能力更加精准的用于家居领域,方便家居行业的使用。

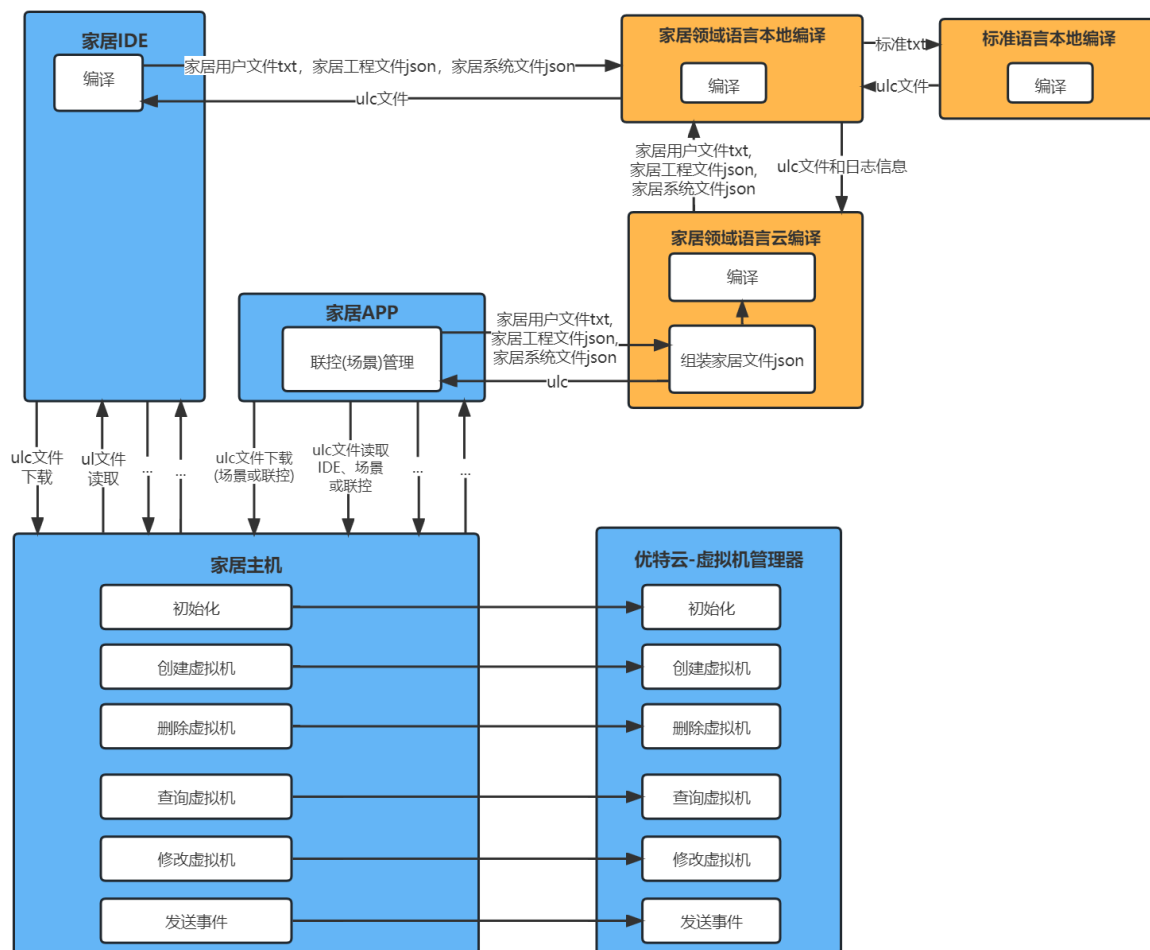
智能家居领域语言,是优特云在智能家居业务领域,为优特物联公司提供的业务语言。用户可使用偏口语化的智能照明术语语言,再通过智能家居云编译成标准语言,标准语言再通过标准编译器编译形成虚拟机的可执行文件,最后在网关设备上执行,达到家居设备之间的本地化联动控制。

2 设计概述

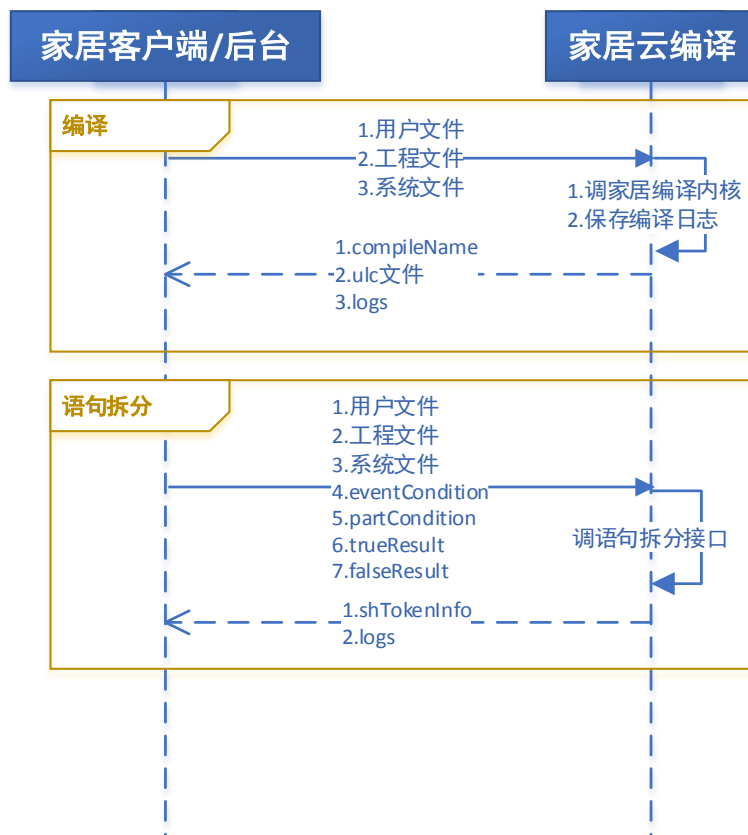
2.1 设计简述

本软件在总体上采用面向对象结合结构化设计的思路,使用设计工具进行设计,对模块进行功能分解和细化,模块采用基于 MVC 的分层结构设计,分为控制层、业务层、数据层。

2.2 系统结构设计



2.3 主要业务流程图



2.4 约束和假定

无

3 接口定义

3.1 业务管理-编译管理

3.1.1 syncCompile

接口定义	定义说明
接口名称	同步编译
请求方式	POST
接口路径	/compile/syncCompile
请求格式	Request body

■ 请求参数

Headers

参数名称	参数值	是否必须	备注
------	-----	------	----

Content-Type	application/json	是	
--------------	------------------	---	--

Body

参数名称	类型	是否必须	备注
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": ""  
        }  
      ]  
    }  
  ]  
}
```

```
        "parasDescription": ""
    },
    {
        "dataType": "int32",
        "parasDescription": ""
    },
    {
        "dataType": "int32",
        "parasDescription": ""
    },
    {
        "dataType": "int32",
        "parasDescription": ""
    },
    {
        "dataType": "int32",
        "parasDescription": ""
    }
],
"returnDataType": "bool"
}
],
"ProductAttributeDefine": [
    {
        "productAttributeInfo": [
            {
                "attributeName": "左继电器",
                "attributeType": 2,
```

■ 返回数据 (Json)

3.1.2 getTokenList

接口定义	定义说明
接口名称	语句的拆分
请求方式	POST
接口路径	/compile/getTokenList
请求格式	Request body

■ 请求参数

Headers

参数名称	参数值	是否必须	备注
Content-Type	application/json	是	

Body

参数名称	类型	是否必须	备注
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

    ]

},

{

    "ulcId": "1",

    "logicNoList": [

        4,

        5,
```

6

```

    ]
  }
]
},
"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": [{

```

UL-T010

```
        "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}"
    },
    {
        "attributeName": "中继继电器",
        "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”																
	<table> <tr> <td>布尔类型</td><td>“bool”</td></tr> <tr> <td>Int8 类型</td><td>”int8”</td></tr> <tr> <td>Int16 类型</td><td>”int16”</td></tr> <tr> <td>Int32 类型</td><td>”int32”</td></tr> <tr> <td>Int64 类型</td><td>”int64”</td></tr> <tr> <td>单精度浮点型</td><td>”float”</td></tr> <tr> <td>双精度浮点型</td><td>”double”</td></tr> <tr> <td>字符串类型</td><td>”string”</td></tr> </table>	布尔类型	“bool”	Int8 类型	”int8”	Int16 类型	”int16”	Int32 类型	”int32”	Int64 类型	”int64”	单精度浮点型	”float”	双精度浮点型	”double”	字符串类型	”string”
布尔类型	“bool”																
Int8 类型	”int8”																
Int16 类型	”int16”																
Int32 类型	”int32”																
Int64 类型	”int64”																
单精度浮点型	”float”																
双精度浮点型	”double”																
字符串类型	”string”																

举例如下：



newTokenlist.json

■ 返回数据（String）

{

"utData": {

"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":{}}},

```

```

    "logs": [
      {
        "logicNo": -1,
        "code": 0,
        "module": -1,
        "level": -1,
        "rowNo": -1,
        "pos": -1,
        "message": "家居语句拆分成功",
        "keywords": {}
      }
    ]
  },
  "utCode": 0,
  "utMsg": ""
}

```

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	“Identifier”	
	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																																						
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	术语类型 <table> <tr> <th>类型</th><th>类型（字符串）</th></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>常数值术语</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>	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	常数值术语	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
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																																						
常数值术语	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-Glo
ssary.json

