

编译

- 修订

更新内容	时间	编写人	版本
编译	2019-12-12	谢一铭	v1.0

一、云编译

业务系统根据业务生成[编译源文件](#) (./language-1bno00sg83psj)后，可通过调用[云编译3.0RestAPI](#) (./language-1blg6goo5mrmh)获得用户语言执行文件。

二、云编译调用方式

业务系统可通过多种方式调用[云编译3.0RestAPI](#) (./language-1blg6goo5mrmh)：

2.1 curl

```
curl -X POST \  
https://www.utcook.com/lancp-compiler-core/public/v3/compile/ga-synchronous \  
-H 'Content-Type: application/json' \  
-d '${compileSrcContent}'
```

2.2 java-第三方业务模块

```
public static Response compile(String compileSrcContent) throws Exception {  
    OkHttpClient httpClient = new OkHttpClient();  
    Request request = new Request.Builder()  
        .post(RequestBody.create(MediaType.parse("application/json;charset=UTF-8"), compileSrcContent))  
        .url("https://www.utcook.com/lancp-compiler-core/public/v3/compile/ga-synchronous")  
        .build();  
  
    return httpClient.newCall(request).execute();  
}
```

2.3 java-优特云SaaS模块

```
@FeignClient(value = "lancp-compiler-core", url = "${platform.service-url.core}")  
public interface FeignCompile {  
  
    @PostMapping(value = "/public/v3/compile/ga-synchronous")  
    CoreCompiledInfo gaCompile(@RequestBody String compileSrcContent);  
  
}
```

其中platform.service-url.core取值见[《基础服务接入说明》](https://docs.utcook.com/docs/framework/framework-1bavg0emodcej)
(<https://docs.utcook.com/docs/framework/framework-1bavg0emodcej>)-框架调整方案-3. Feign远程调用-① 配置文件改造

上传

- 修订

更新内容	时间	编写人	版本
上传	2019-12-12	谢一铭	v1.0

一、上传用户语言执行文件

业务系统通过调用[云编译3.0RestAPI](#) (/language-1blg6goo5mrmh)获得用户语言执行文件后，需要将文件上传到云端，才能使用[云执行3.0](#) (/language-1blfn4ctgmr1o)。

二、上传示例

请求URL

- URL:
<https://www.utcook.com/binfile/v3/bin-file> (<https://www.utcook.com/binfile/v3/bin-file>)

请求方式

- POST

请求头

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型：text/plain
Authorization	是	string	OAuth2格式的鉴权Token

请求参数

参数名	是否必须	类型	说明
base64BinContent	是	string	基于BASE64编码的用户语言执行文件字节流

返回示例

d2ae70ab-2e2d-4d21-b4a6-1c74aa21cce6

用户语言执行文件ID

三、调用方式

业务系统可通过多种方式调用上述RestAPI：

3.1 curl

```
curl -X POST \  
  https://www.utcook.com/binfile/v3/bin-file \  
  -H 'Authorization: Bearer ${accessToken}' \  
  -H 'Content-Type: text/plain' \  
  -d 8wgAAA...
```

3.2 java-第三方业务模块

```
public static Response uploadBinFile(String base64BinContent, String accessToken) throws Exception {  
    OkHttpClient httpClient = new OkHttpClient();  
    Request request = new Request.Builder()  
        .header("Authorization", "bearer " + accessToken)  
        .post(RequestBody.create(MediaType.parse("text/plain; charset=UTF-8"), base64BinContent))  
        .url("https://www.utcook.com/binfile/v3/bin-file")  
        .build();  
  
    return httpClient.newCall(request).execute();  
}
```

3.3 java-优特云SaaS模块

```
@FeignClient(value = "lang-binfile-service", url = "${platform.service-url.binfile}")  
)  
public interface FeignBinFile {  
  
    @PostMapping(value = "/v3/bin-file")  
    String uploadBinFile(@RequestBody String base64BinContent);  
  
}
```

其中platform.service-url.binfile取值见[《基础服务接入说明》](#)
(<https://docs.utcook.com/docs/framework/framework-1bavg0emodcej>)-框架调整方案-3. Feign远程调用-① 配置文件改造

启动

- 修订

更新内容	时间	编写人	版本
启动	2019-12-12	谢一铭	v1.0

一、启动用户语言执行文件

业务系统将文件上传到云端后，可通过调用[云执行3.0RestAPI \(/language-1blfn4ctgmr1o\)](#)启动用户语言执行文件。

二、启动方式

2.1 以静态实例方式启动

用户语言执行文件以静态实例方式启动后，每次执行的数据将相互隔离，上次执行的执行状态不会带入下次执行中。

以静态实例方式启动需要调用[云执行3.0RestAPI \(/language-1blfn4ctgmr1o\)](#)的[服务上架 \(/language-1blfn4ctgmr1o#29obmz\)](#)接口。

接口调用前，业务系统需在[用户中心 \(/services/services-1bl3k19p7dfog\)](#)注册应用；调用时，应为当前执行文件设置启动别名，配合注册应用获得的应用appKey，使用静态启动分类，并传入文件[上传 \(/language/upload bin file to bin file service\)](#)获得的用户语言执行文件ID，一并作为启动参数进行启动。

2.2 以动态实例方式启动

用户语言执行文件以动态实例方式启动后，同一实例下的执行数据将共享，上次执行的执行状态将带入下次执行中。

以动态实例方式启动同样需要调用[云执行3.0RestAPI \(/language-1blfn4ctgmr1o\)](#)的[服务上架 \(/language-1blfn4ctgmr1o#29obmz\)](#)接口。

启动参数与静态方式启动所需参数基本一致，不同的是启动分类需使用动态。

三、云执行启动调用方式

业务系统可通过多种方式调用[云执行3.0RestAPI \(/language-1blfn4ctgmr1o\)](#)的[服务上架 \(/language-1blfn4ctgmr1o#29obmz\)](#)接口：

3.1 静态实例方式启动

3.1.1 curl

```
curl -X POST \
  https://www.utcook.com/lvmp/serviceDeveloper/publish \
  -H 'Authorization: Bearer ${accessToken}' \
  -H 'Content-Type: application/json' \
  -d '{
    "appKey": "${应用appKey}",
    "alias": "${启动别名}",
    "category": 0,
    "binId": "${用户语言执行文件ID}"
  }'
```

3.1.2 java-第三方业务模块

```
public static Response launchAsStatic(String publishVOInJsonStr, String accessToken)
throws Exception {
    OkHttpClient httpClient = new OkHttpClient();
    Request request = new Request.Builder()
        .header("Authorization", "bearer " + accessToken)
        .post(RequestBody.create(MediaType.parse("application/json;charset=UTF-8"), publishVOInJsonStr))
        .url("https://www.utcook.com/lvmp/serviceDeveloper/publish")
        .build();

    return httpClient.newCall(request).execute();
}
```

3.1.3 java-优特云SaaS模块

```
@FeignClient(value = "lang-vm-service", url = "${platform.service-url.vm}")
public interface FeignLvmp {

    @PostMapping(value = "/serviceDeveloper/publish")
    void publish(@RequestBody PublishVO publishVO);

}
```

其中platform.service-url.vm取值见[《基础服务接入说明》](#)
<https://docs.utcook.com/docs/framework/framework-1bavg0emodcej>)-框架调整方案-3. Feign远程调用-① 配置文件改造；PublishVO结构见[服务上架](#)(./language-1blfn4ctgmr1o#29obmz)。

3.2 动态实例方式启动

3.2.1 curl

```
curl -X POST \
  https://www.utcook.com/lvmp/serviceDeveloper/publish \
  -H 'Authorization: Bearer ${accessToken}' \
  -H 'Content-Type: application/json' \
  -d '{
    "appKey": "${应用appKey}",
    "alias": "${启动别名}",
    "category": 1,
    "binId": "${用户语言执行文件ID}"
  }'
```

3.2.2 java-第三方业务模块

见3.1.2

3.2.3 java-优特云SaaS模块

见3.1.3

3.3 动态启动后创建实例

3.3.1 curl

```
curl -X POST \
  https://www.utcook.com/lvmp/serviceUser/instance/create \
  -H 'Authorization: Bearer ${accessToken}' \
  -H 'Content-Type: text/plain' \
  -d ${启动别名}
```

3.3.2 java-第三方业务模块

```
public static Response instantiateAsDynamic(String alias, String accessToken) throws
Exception {
    MultiValueMap<String, String> params = new LinkedMultiValueMap<>();
    param.put("alias", Collections.singletonList(alias));
    String encodedUrl = "https://www.utcook.com/lvmp/serviceUser/instance/creat
e" + encodeGetQuery(params);

    OkHttpClient httpClient = new OkHttpClient();
    Request request = new Request.Builder()
        .url(encodedUrl)
        .header("Authorization", "bearer " + accessToken)
        .build();

    return httpClient.newCall(request).execute();
}
```

3.3.3 java-优特云SaaS模块

```
@FeignClient(value = "lang-vm-service", url = "${platform.service-url.vm}")
public interface FeignLvmp {

    @PostMapping(value = "/serviceUser/instance/create")
    String createDynamicInstance(@RequestParam(value = "alias") String alias);

}
```

其中platform.service-url.vm取值见[《基础服务接入说明》](#)
(<https://docs.utcook.com/docs/framework/framework-1bavg0emodcej>)-框架调整方案-3. Feign远程调用-① 配置文件改造；PublishVO结构见[服务上架](#) ([./language-1blfn4ctgmr1o#29obmz](#))。

执行

- 修订

更新内容	时间	编写人	版本
执行	2019-12-12	谢一铭	v1.0

一、云执行

业务系统通过[云执行3.0RestAPI \(/language-1blfn4ctgmr1o\)](#)启动用户语言执行文件后，可通过调用调用[云执行3.0RestAPI \(/language-1blfn4ctgmr1o\)](#)触发文件执行。

二、执行触发方式

云执行目前支持4种执行触发方式：

- 应用触发
- 设备消息触发
- Kafka消息触发
- 定时触发

并且，以不同方式启动的用户语言执行文件，其执行接口也不同。

2.1 应用触发(主要)

业务系统可调用应用触发接口，通过更新应用变量的方式来触发执行。应用触发接口在[云执行 \(/language-1blfn4ctgmr1o\)](#)中称为[运行静态定制流 \(/language-1blfn4ctgmr1o#4bkaeh\)](#)/[运行动态定制流 \(/language-1blfn4ctgmr1o#f6uh7q\)](#)，分别与不同的启动方式对应。

2.2 设备消息触发

当业务系统在[编译源文件 \(/language-1bno00sg83psj\)](#)的[逻辑语句 \(/language-1bnpk42dorslf\)](#)中使用设备类[优模型实例 \(/basic_conception_on_cloud\)](#)作为条件时，设备消息即可触发用户语言执行文件的执行。

2.3 Kafka消息触发

类似的，当业务系统在[编译源文件 \(/language-1bno00sg83psj\)](#)的[逻辑语句 \(/language-1bnpk42dorslf\)](#)中使用用户语言执行文件类[优模型实例 \(/basic_conception_on_cloud\)](#)作为条件时，其他用户语言执行文件执行过程中生成的Kafka消息即可触发当前启动的文件的执行。

2.4 定时触发

当业务系统在[编译源文件 \(/language-1bno00sg83psj\)](#)的[逻辑语句 \(/language-1bnpk42dorslf\)](#)中使用云执行提供的定时特性作为条件时，用户语言执行文件即可被定时触发执行。

三、云执行应用触发调用方式：

业务系统可通过多种方式调用[云执行3.0RestAPI \(/language-1blfn4ctgmr1o\)](#)的[运行静态定制流 \(/language-1blfn4ctgmr1o#4bkaeh\)](#)/[运行动态定制流 \(/language-1blfn4ctgmr1o#f6uh7q\)](#)接口：

3.1 运行静态定制流

3.1.1 curl

```
curl -X POST \
  https://www.utcook.com/lvmp/serviceUser/flowService/run \
  -H 'Authorization: Bearer ${accessToken}' \
  -H 'Content-Type: application/json' \
  -d '{
    "alias": "${启动别名}",
    "param": {
      "varList": [
        {
          "object": {
            "data": {},
            "len": 0,
            "type": 0
          },
          "varPtr": {
            "variableKey": "${应用变量varKey}",
            "variableType": 0,
            "varid": 0
          }
        }
      ]
    }
  }.
```

3.1.2 java-第三方业务模块

```
public static Response runStatic(String flowServiceExecuteDT0InJsonStr, String accessToken) throws Exception {
    OkHttpClient httpClient = new OkHttpClient();
    Request request = new Request.Builder()
        .header("Authorization", "bearer " + accessToken)
        .post(RequestBody.create(MediaType.parse("application/json;charset=UTF-8"), flowServiceExecuteDT0InJsonStr))
        .url("https://www.utcook.com/lvmp/serviceUser/flowService/run")
        .build();

    return httpClient.newCall(request).execute();
}
```

3.1.3 java-优特云SaaS模块

```

@FeignClient(value = "lang-vm-service", url = "${platform.service-url.vm}")
public interface FeignLVMPServiceUser {

    @RequestMapping(value = "/serviceUser/flowService/run", method = RequestMethod.POST)
    String runFlowService(@RequestBody FlowServiceExecuteDTO flowServiceExecuteDTO);

}

```

其中platform.service-url.vm取值见[《基础服务接入说明》](#)(./framework/framework-1bavg0emodcej)-框架调整方案-3. Feign远程调用-① 配置文件改造；PublishVO结构见[服务上架](#)(./language-1blfn4ctgmr1o#29obmz)。

3.2 运行动态定制流

3.1.1 curl

```

curl -X POST \
  https://www.utcook.com/lvmp/serviceUser/dynamicFlow/run \
  -H 'Authorization: Bearer ${accessToken}' \
  -H 'Content-Type: application/json' \
  -d '{
    "executeInstanceId": "${动态实例ID}",
    "param": {
      "varList": [
        {
          "object": {
            "data": {},
            "len": 0,
            "type": 0
          },
          "varPtr": {
            "variableKey": "${应用变量varKey}",
            "variableType": 0,
            "varid": 0
          }
        }
      ]
    }
  }.'

```

3.1.2 java-第三方业务模块

```

public static Response runDynamic(String dynamicFlowExecuteDTOInJsonStr, String accessToken) throws Exception {
    OkHttpClient httpClient = new OkHttpClient();
    Request request = new Request.Builder()
        .header("Authorization", "bearer " + accessToken)
        .post(RequestBody.create(MediaType.parse("application/json;charset=UTF-8"), dynamicFlowExecuteDTOInJsonStr))
        .url("https://www.utcook.com/lvmp/serviceUser/dynamicFlow/run")
        .build();

    return httpClient.newCall(request).execute();
}

```

3.1.3 java-优特云SaaS模块

```

@FeignClient(value = "lang-vm-service", url = "${platform.service-url.vm}")
public interface FeignLVMPServiceUser {

    @RequestMapping(value = "/serviceUser/dynamicFlow/run", method = RequestMethod.POST)
    String runDynamicFlow(@RequestBody DynamicFlowExecuteDTO dynamicFlowExecuteDTO);

}

```

其中platform.service-url.vm取值见[《基础服务接入说明》](#)
<https://docs.utcook.com/docs/framework/framework-1bavg0emodcej>)-框架调整方案-3. Feign远程调用-① 配置文件改造；PublishVO结构见[服务上架 \(/language-1blfn4ctgmr1o#29obmz\)](#)。