

优特云

家居 4 寸网关接入 虚拟机 Java 层 技术使用手册

商用虚拟机+公共函数库

文件审批：

	部门/职位	签名	日期
编制			
编制			
会签			
会签			
审核			
批准			

文档修改记录：

版本	修改内容概要	修改人	批准人	生效日期
V0.1	1. 新建	刘春生		2023/05/05
V0.2	1. 修改 2. 2. 2com. utyun. bizvm. jni. AbsAbilityDriver 内容，删除 getVarKeyEnhance、 getVarIdEnhance、 getFuncKeyEnhance、 getFuncIdEnhance4 个抽象 方法，由 sdk 提供默认实现， 应用层如有需要再进行重写 2. 修改 1. 3 家居 4 寸网关接入 虚拟机 java 层概述的流程 图，去掉驱动外部能力的 4 个抽象方法(同上) 3. 修改 3. 2. 1 继承抽象类内容， 去掉 AbilityDriver 类对 4 个抽象方法的实现	刘春生		2023/06/27

目录

第一章 概述	4
1.1. 商用虚拟机 Java 层概述	4
1.2. 公共函数库 Java 层概述	4
1.3. 家居 4 寸网关接入虚拟机 java 层概述	5
第二章 SDK 核心数据结构及类说明	7
2.1. 基础数据结构定义	7
2.2. 外部能力及公共函数库抽象类	10
2.3. sdk 已提供的部分外部能力实现	14
第三章 SDK 接口	15
3.1. 接口定义	15
3.2. 接口使用集成样例	23

第一章 概述

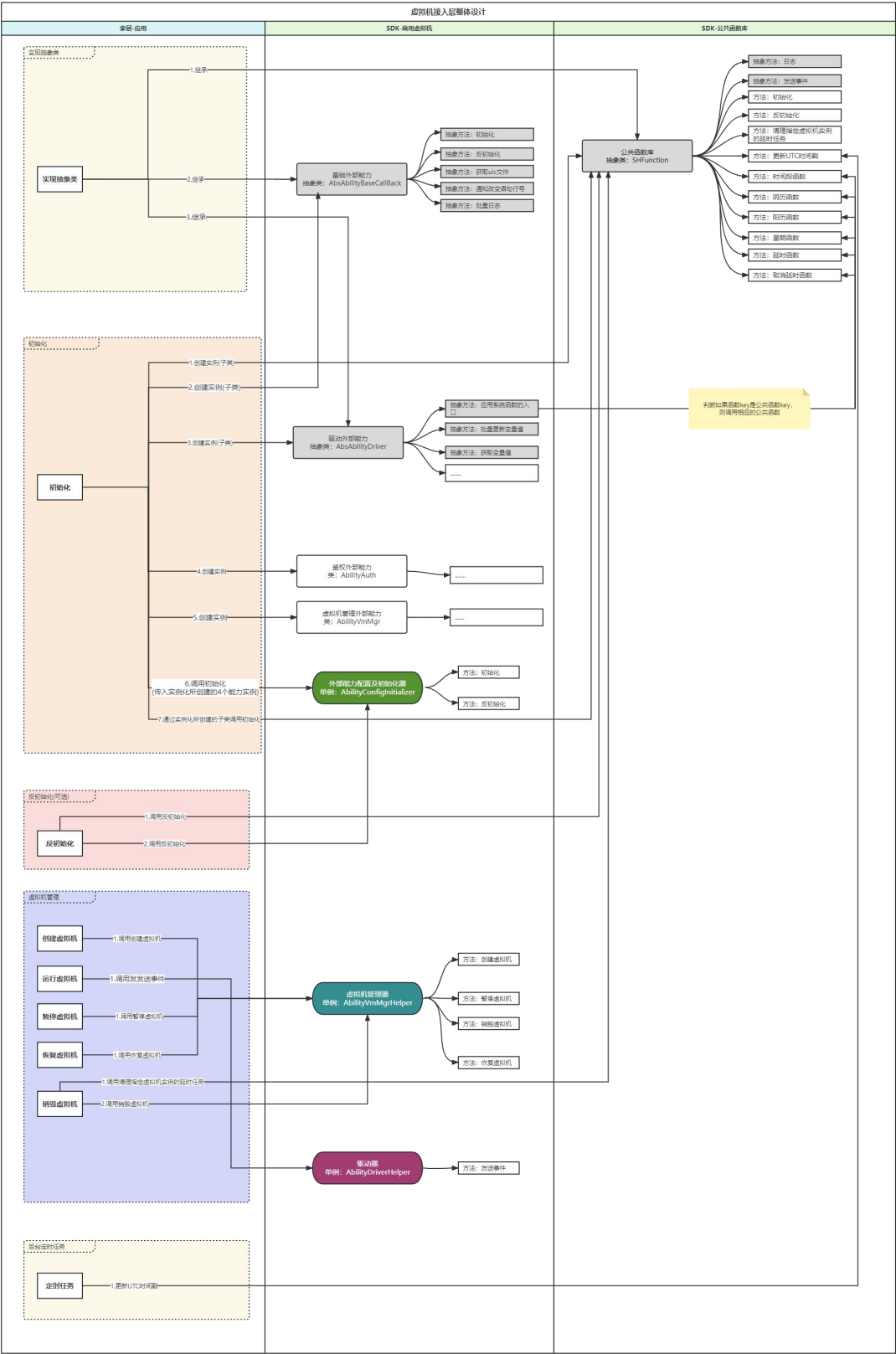
1.1. 商用虚拟机 Java 层概述

商用虚拟机 Java 层对原商用虚拟机 SDK 进行了封装,提供了 java 语言版本。主要包括商用虚拟机配置初始化及管理(创建/运行/暂停/恢复/销毁)等功能,以 jar 包的形式提供。

1.2. 公共函数库 Java 层概述

公共函数库 Java 层对原公共函数库 SDK 进行了封装,提供了 java 语言版本。主要包括初始化及公共函数调用等功能,以源码的形式提供。

1.3. 家居 4 寸网关接入虚拟机 java 层概述



上图最左侧浅蓝色列为家居应用侧，右侧浅绿色列为商用虚拟机 SDK 与公

共函数库 SDK；下方各颜色方框代表主要的功能；灰色方框代表应用侧接入需实现的内容。

第二章 SDK 核心数据结构及类说明

2.1. 基础数据结构定义

2.1.1. com.utyun.bizvm.bean.external.UResult

```
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class UResult<T> {
    private boolean ok; // 是否成功
    private T data;
}
```

2.1.2. com.utyun.bizvm.bean.external.UResultExBase

```
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class UResultExBase {
    private boolean ok; // 是否成功
    private int errorCode; // 错误码
    private String msg; // 错误码提示信息
}
```

2.1.3. com.utyun.bizvm.bean.external.UResultEx

```
import lombok.Data;
import lombok.EqualsAndHashCode;
import lombok.ToString;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@ToString(callSuper = true)
@EqualsAndHashCode(callSuper = true)
@NoArgsConstructor
@AllArgsConstructor
public class UResultEx<T> extends UResultExBase {
    private T data;
}
```

```
}
```

2.1.4. com.utyun.bizvm.bean.external.ULog

```
import com.utyun.bizvm.enums.external.ULogLevelEnum;
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class ULog {
    private ULogLevelEnum level;
    private String tag;
    private String log;
}
```

2.1.5. com.utyun.bizvm.bean.external.UObject

```
import com.utyun.bizvm.enums.external.UDataTypeEnum;
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class UObject {
    protected Object data;
    protected UDataTypeEnum dataType;
}
```

2.1.6. com.utyun.bizvm.bean.external.UVar

```
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class UVar {
    private int id;
    private UObject value = null;
}
```


2.1.7. com.utyun.bizvm.bean.external.UVarEx

```
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class UVarEx {
    private String varKey;
    private UObject value = null;
}
```

2.1.8. com.utyun.bizvm.enums.external.UDataTypeEnum

```
import lombok.Getter;

public enum UDataTypeEnum {
    VOID(0x20, "void", null),
    INT8(0x21, "int8", 0),
    INT16(0x22, "int16", 0),
    INT32(0x23, "int32", 0),
    INT64(0x24, "int64", 0),
    FLOAT(0x25, "float", 0f),
    DOUBLE(0x26, "double", 0.0),
    BOOL(0x27, "bool", false),
    STRING(0x28, "string", ""),

    NIL(0xFF, "nil", null);

    @Getter
    private final int code;
    @Getter
    private final String desc;
    @Getter
    private final Object defaultValue;

    UDataTypeEnum(int code, String desc, Object defaultValue) {
        this.code = code;
        this.desc = desc;
        this.defaultValue = defaultValue;
    }

    public static UDataTypeEnum getByCode(int code) {
        for (UDataTypeEnum datatype : UDataTypeEnum.values()) {
            if (datatype.getCode() == code) {
                return datatype;
            }
        }
        return NIL;
    }

    public static UDataTypeEnum getByDesc(String desc) {
        for (UDataTypeEnum datatype : UDataTypeEnum.values()) {
```

```

        if (datatype.getDesc().equals(desc)) {
            return datatype;
        }
    }
    return NIL;
}
}

```

2.1.9. com.utyun.bizvm.enums.external.ULogLevelEnum

```

import lombok.Getter;

public enum ULogLevelEnum {
    DEBUG(0),
    INFO(1),
    WARNING(2),
    ERROR(3);

    @Getter
    private final int code;

    ULogLevelEnum(int code) {
        this.code = code;
    }

    public static ULogLevelEnum getByCode(int code) {
        for (ULogLevelEnum priority : ULogLevelEnum.values()) {
            if (priority.getCode() == code) {
                return priority;
            }
        }
        return null;
    }
}

```

2.2. 外部能力及公共函数库抽象类

2.2.1. com.utyun.bizvm.jni.AbsAbilityBaseCallBack（基础外部能力抽象类）

```

import com.utyun.bizvm.bean.external.UResult;
import com.utyun.bizvm.bean.external.UResultExBase;

public abstract class AbsAbilityBaseCallBack {

    /**
     * 初始化
     */
    public abstract UResultExBase initEnhance();

    /**
     * 反初始化
     */
}

```

```

public abstract void deinit();

/**
 * 获取 ulc 文件内容
 *
 * @param ctxId 上下文 id, 调用创建虚拟机接口时传的 ctxId
 * @return ulc 文件字节
 */
public abstract UResult<byte[]> getUlcEnhance(String ctxId);

/**
 * 通知改变语句行号
 *
 * @param vmId 虚拟机 id
 * @param lineNo 当前行号
 */
public abstract void onLineNumberChange(long vmId, long lineNo);

/**
 * 批量日志输出
 *
 * @param batchLogJson 日志列表 json 串, 格式为: List<ULog>
 */
public abstract void batchLogout(String batchLogJson);
}

```

2.2.2. com.utyun.bizvm.jni.AbsAbilityDriver（驱动外部能力抽象类）

```

import com.utyun.bizvm.bean.external.UObject;
import com.utyun.bizvm.bean.external.UResult;
import com.utyun.bizvm.bean.external.UVarEx;

import java.util.List;

public abstract class AbsAbilityDriver extends AbsAbilityDriverCallBack
{
    /**
     * 应用系统函数的入口
     *
     * @param vmId 虚拟机 id
     * @param funKey 函数 key
     * @param params 参数列表, 无参数时为 null
     * @return 函数返回值 (无返回值是 data=null)
     */
    public abstract UResult<UObject> appFuncEnhance(long vmId, String funKey, List<UObject>
params);

    /**
     * 批量更新变量值
     *
     * @param vmId 虚拟机 id
     * @param varList 变量列表
     * @return 是否成功
     */
}

```

```

public abstract boolean batchSetVarEnhance(long vmId, List<UVarEx> varList);

/**
 * 获取变量值
 *
 * @param vmId 虚拟机 id
 * @param varKey 变量 key
 * @return 变量值
 */
public abstract UResult<UObject> getVarEnhance(long vmId, String varKey);

// 应用层无需关心其它方法实现，因此进行了省略
}

```

2.2.3. com.utyun.shfunction.SHFunction（公共函数库抽象类）

```

import com.utyun.bizvm.bean.external.UObject;
import com.utyun.bizvm.bean.external.UResult;
import com.utyun.bizvm.bean.external.UVarEx;
import com.utyun.bizvm.enums.external.ULogLevelEnum;

import java.util.List;

public abstract class SHFunction {

    /**
     * 日志
     *
     * @param level 日志级别
     * @param msg 日志
     */
    public abstract void logout(ULogLevelEnum level, String msg);

    /**
     * 发送事件
     *
     * @param vmId 虚拟机 id
     * @param eventList 事件列表
     * @return 是否成功
     */
    public abstract boolean postEvent(long vmId, List<UVarEx> eventList);

    /**
     * 初始化
     *
     * @param delayKey 延时变量
     * @param timeValKey 时间点变量 key
     * @return 是否成功
     */
    public boolean init(String delayKey, String timeValKey) {
        // 应用层无需关心该方法内部代码，因此进行了省略
        return false;
    }

    /**
     * 反初始化
     *

```

```

    * @return 是否成功
    */
    public boolean deinit() {
        // 应用层无需关心该方法内部代码，因此进行了省略
        return false;
    }

    /**
     * 清理指定虚拟机实例的延时任务，（销毁虚拟机实例前需要调用）
     *
     * @param iVmId 虚拟机 id
     * @return 是否成功
     */
    public boolean cleanUpDelayTask(long iVmId) {
        // 应用层无需关心该方法内部代码，因此进行了省略
        return false;
    }

    /**
     * 更新 UTC 时间戳
     *
     * @param timestamp UTC 时间戳（以 1970-01-01 00:00:00 开始计算），单位为毫秒
     * @return 是否成功
     */
    public boolean timeUpdate(long timestamp) {
        // 应用层无需关心该方法内部代码，因此进行了省略
        return false;
    }

    /**
     * 时间段函数
     *
     * @param vmId 虚拟机 id
     * @param params 参数列表
     * @return 函数返回值
     */
    public UResult<UObject> funcTimeRang(long vmId, List<UObject> params) {
        // 应用层无需关心该方法内部代码，因此进行了省略
        return null;
    }

    /**
     * 阴历函数
     *
     * @param vmId 虚拟机 id
     * @param params 参数列表
     * @return 函数返回值
     */
    public UResult<UObject> funcLunarTime(long vmId, List<UObject> params) {
        // 应用层无需关心该方法内部代码，因此进行了省略
        return null;
    }

    /**
     * 阳历函数
     *
     * @param vmId 虚拟机 id
     * @param params 参数列表

```

```

    * @return 函数返回值
    */
    public UResult<UObject> funcCalendar(long vmId, List<UObject> params) {
        // 应用层无需关心该方法内部代码，因此进行了省略
        return null;
    }

    /**
     * 星期函数
     *
     * @param vmId 虚拟机 id
     * @param params 参数列表
     * @return 函数返回值
     */
    public UResult<UObject> funcWeekDay(long vmId, List<UObject> params) {
        // 应用层无需关心该方法内部代码，因此进行了省略
        return null;
    }

    /**
     * 延时函数
     *
     * @param vmId 虚拟机 id
     * @param params 参数列表
     * @return 函数返回值
     */
    public UResult<UObject> funcDelay(long vmId, List<UObject> params) {
        // 应用层无需关心该方法内部代码，因此进行了省略
        return null;
    }

    /**
     * 取消延时函数
     *
     * @param vmId 虚拟机 id
     * @param params 参数列表
     * @return 函数返回值
     */
    public UResult<UObject> funcCancelDelay(long vmId, List<UObject> params) {
        // 应用层无需关心该方法内部代码，因此进行了省略
        return null;
    }
}

```

2.3. sdk 已提供的部分外部能力实现

2.3.1. com.utyun.bizvm.jni.AbilityAuth

```

import lombok.NoArgsConstructor;

@NoArgsConstructor
public class AbilityAuth extends AbsAbilityAuthCallBack {
    // 应用层无需关心该类内部代码，因此进行了省略
    .....
}

```

2.3.2. com.utyun.bizvm.jni.AbilityVmMgr

```
import lombok.NoArgsConstructor;

@NoArgsConstructor
public class AbilityVmMgr extends AbsAbilityVmMgrCallBack {
    // 应用层无需关心该类内部代码，因此进行了省略
    .....
}
```

第三章 SDK 接口

3.1. 接口定义

3.1.1. 【虚拟机】初始化（外部能力配置及初始化）

类：

com.utyun.bizvm.sdk.AbilityConfigInitializer

方法：

public synchronized UResultExBase initAbility(AbsAbilityBaseCallBack baseCB,
AbsAbilityVmMgrCallBack vmMgrCB, AbsAbilityAuthCallBack authCB, AbsAbilityDriverCallBack driverCB)

参数：

名称	类型	备注
baseCB	com.utyun.bizvm.jni.AbsAbilityBaseCallBack	基础外部能力 1.应用侧需自行继承 com.utyun.bizvm.jni.AbsAbilityBaseCallBack，创建实例
vmMgrCB	com.utyun.bizvm.jni.AbsAbilityVmMgrCallBack	虚拟机管理外部能力 1.通过 com.utyun.bizvm.jni.AbilityVmMgr 的无参构造函数创建实例
authCB	com.utyun.bizvm.jni.AbsAbilityAuthCallBack	鉴权外部能力 1.通过 com.utyun.bizvm.jni.AbilityAuth 的无参构造函数创建实例
driverCB	com.utyun.bizvm.jni.AbsAbilityDriverCallBack	驱动外部能力 1.应用侧需自行继承 com.utyun.bizvm.jni.AbsAbilityDriver，创建实例

返回值：

名称	类型	备注
----	----	----

ok	boolean	是否成功
errorCode	int	错误码
msg	String	错误码提示信息

使用：

```
AbilityConfigInitializer.getInstance().initAbility(...)
```

3.1.2. 【虚拟机】反初始化

类：

com.utyun.bizvm.sdk.AbilityConfigInitializer

方法：

public synchronized void deInitAbility()

参数：

无

返回值：

无

使用：

```
AbilityConfigInitializer.getInstance().deInitAbility()
```

3.1.3. 【虚拟机】创建虚拟机

类：

com.utyun.bizvm.sdk.AbilityVmMgrHelper

方法：

public UResultEx createVm(String ctxId, String secret)

参数：

名称	类型	备注
ctxId	String	上下文 id，由应用侧自行传入,后续会通过该上下文 id 获取 ulc 文件
secret	String	鉴权密钥，由优特云提供

返回值：

名称	类型	备注
ok	boolean	是否成功
errorCode	int	错误码
data	Long	虚拟机 id
msg	String	错误码提示信息

使用：

```
AbilityVmMgrHelper.getInstance().createVm(...)
```


3.1.4. 【虚拟机】暂停虚拟机

类:

com.utyun.bizvm.sdk.AbilityVmMgrHelper

方法:

public UResultExBase pauseVm(long vmId)

参数:

名称	类型	备注
vmId	long	虚拟机 id(创建虚拟机时返回的虚拟机 id)

返回值:

名称	类型	备注
ok	boolean	是否成功
errorCode	int	错误码
msg	String	错误码提示信息

使用:

```
AbilityVmMgrHelper.getInstance().pauseVm(...)
```

3.1.5. 【虚拟机】恢复虚拟机

类:

com.utyun.bizvm.sdk.AbilityVmMgrHelper

方法:

public UResultExBase resumeVm(long vmId)

参数:

名称	类型	备注
vmId	long	虚拟机 id(创建虚拟机时返回的虚拟机 id)

返回值:

名称	类型	备注
ok	boolean	是否成功
errorCode	int	错误码
msg	String	错误码提示信息

使用:

```
AbilityVmMgrHelper.getInstance().resumeVm(...)
```

3.1.6. 【虚拟机】销毁虚拟机

类:

com.utyun.bizvm.sdk.AbilityVmMgrHelper

方法:

public void freeVm(long vmId)

参数:

名称	类型	备注
vmId	long	虚拟机 id(创建虚拟机时返回的虚拟机 id)

返回值:

无

使用:

```
AbilityVmMgrHelper.getInstance().freeVm(...)
```

3.1.7. 【虚拟机】运行虚拟机（发送事件）

类:

com.utyun.bizvm.sdk.AbilityDriverHelper

方法:

```
public UResultExBase postEvent(long vmId, List<UVarEx> eventList)
```

参数:

名称	类型	备注
vmId	long	虚拟机 id(创建虚拟机时返回的虚拟机 id), 当 vmId 传 0 时, 向所有虚拟机发送事件
eventList	List<UVarEx>	事件列表

返回值:

名称	类型	备注
ok	boolean	是否成功
errorCode	int	错误码
msg	String	错误码提示信息

使用:

```
AbilityDriverHelper.getInstance().postEvent(...)
```

3.1.8. 【公共函数库】初始化

类:

com.utyun.shfunction.SHFunction

方法:

```
public boolean init(String delayKey, String timeValKey)
```

参数:

名称	类型	备注
delayKey	String	延时变量
timeValKey	String	时间点变量 key

返回值:

名称	类型	备注
ok	boolean	是否成功

3.1.9. 【公共函数库】反初始化

类:

com.utyun.shfunction.SHFunction

方法:

public boolean deinit()

参数:

无

返回值:

名称	类型	备注
ok	boolean	是否成功

3.1.10. 【公共函数库】日志输出

类:

com.utyun.shfunction.SHFunction

方法:

public abstract void logout(LogLevel level, String msg)

参数:

名称	类型	备注
level	LogLevelEnum	日志级别, com.utyun.bizvm.enums.e xternal.LogLevelEnum
msg	String	日志

返回值:

无

3.1.11. 【公共函数库】发送事件

类:

com.utyun.shfunction.SHFunction

方法:

public abstract boolean postEvent(long vmId, List<UVarEx> eventList)

参数:

名称	类型	备注
vmId	long	虚拟机 id (创建虚拟机时 返回的虚拟机 id)
eventList	List<UVarEx>	事件列表

返回值:

名称	类型	备注
ok	boolean	是否成功

3.1.12. 【公共函数库】清理指定虚拟机实例的延时任务

类:

com.utyun.shfunction.SHFunction

方法:

public boolean cleanUpDelayTask(long iVmId)

参数:

名称	类型	备注
iVmId	long	销毁虚拟机的虚拟机 id

返回值:

名称	类型	备注
ok	boolean	是否成功

3.1.13. 【公共函数库】更新 UTC 时间戳

类:

com.utyun.shfunction.SHFunction

方法:

public boolean timeUpdate(long timestamp)

参数:

名称	类型	备注
timestamp	long	UTC 时 间 戳 （ 以 1970-01-01 00:00:00 开始计算），单位为毫秒

返回值:

名称	类型	备注
ok	boolean	是否成功

3.1.14. 【公共函数库】时间段函数

类:

com.utyun.shfunction.SHFunction

方法:

UResult<UObject> funcTimeRang(long vmId, List<UObject> params)

参数:

名称	类型	备注
vmId	long	虚拟机 id (创建虚拟机时返回的虚拟机 id)
params	List<UObject>	参数列表

返回值:

名称	类型	备注
ok	boolean	是否成功

data	UObject	函数返回值
------	---------	-------

3. 1. 15. 【公共函数库】阴历函数

类:

com.utyun.shfunction.SHFunction

方法:

public UResult<UObject> funcLunarTime(long vmId, List<UObject> params)

参数:

名称	类型	备注
vmId	long	虚拟机 id(创建虚拟机时返回的虚拟机 id)
params	List<UObject>	参数列表

返回值:

名称	类型	备注
ok	boolean	是否成功
data	UObject	函数返回值

3. 1. 16. 【公共函数库】阳历函数

类:

com.utyun.shfunction.SHFunction

方法:

UResult<UObject> funcCalendar(long vmId, List<UObject> params)

参数:

名称	类型	备注
vmId	long	虚拟机 id(创建虚拟机时返回的虚拟机 id)
params	List<UObject>	参数列表

返回值:

名称	类型	备注
ok	boolean	是否成功
data	UObject	函数返回值

3. 1. 17. 【公共函数库】星期函数

类:

com.utyun.shfunction.SHFunction

方法:

UResult<UObject> funcWeekDay(long vmId, List<UObject> params)

参数:

名称	类型	备注
vmId	long	虚拟机 id (创建虚拟机时返回的虚拟机 id)
params	List<UObject>	参数列表

返回值:

名称	类型	备注
ok	boolean	是否成功
data	UObject	函数返回值

3.1.18. 【公共函数库】延时函数

类:

com.utyun.shfunction.SHFunction

方法:

public UResult<UObject> funcDelay(long vmId, List<UObject> params)

参数:

名称	类型	备注
vmId	long	虚拟机 id (创建虚拟机时返回的虚拟机 id)
params	List<UObject>	参数列表

返回值:

名称	类型	备注
ok	boolean	是否成功
data	UObject	函数返回值

3.1.19. 【公共函数库】取消延时函数

类:

com.utyun.shfunction.SHFunction

方法:

public UResult<UObject> funcCancelDelay(long vmId, List<UObject> params)

参数:

名称	类型	备注
vmId	long	虚拟机 id (创建虚拟机时返回的虚拟机 id)
params	List<UObject>	参数列表

返回值:

名称	类型	备注
ok	boolean	是否成功
data	UObject	函数返回值

3.2. 接口使用集成样例

- 以下样例中，主要关注接口使用，相关参数或变量等为模拟构造的，请勿照抄
- 在项目中引入商用虚拟机 jar 包，并将 `com.utyun.shfunction.SHFunction` 类拷贝至项目中

3.2.1. 继承抽象类

```
/**
 * 继承 AbsAbilityBaseCallBack 抽象类
 */
public class AbilityBase extends AbsAbilityBaseCallBack {
    @Override
    public UResultExBase initEnhance() {
        // TODO 自行实现逻辑
        return null;
    }

    @Override
    public UResult<byte[]> getUlcEnhance(String ctxId) {
        // TODO 自行实现逻辑
        return null;
    }

    @Override
    public void onLineNumberChange(long vmId, long lineNo) {
        // TODO 自行实现逻辑
    }

    @Override
    public void batchLogout(String batchLogJson) {
        // TODO 自行实现逻辑
    }

    @Override
    public void deinit() {
        // TODO 自行实现逻辑
    }
}
```

```
/**
 * 继承 AbsAbilityDriver 抽象类
 */
public class AbilityDriver extends AbsAbilityDriver {
    @Override
    public UResult<UObject> appFuncEnhance(long vmId, String funKey, List<UObject> params) {
        // TODO 自行实现逻辑
        return null;
    }

    @Override
    public boolean batchSetVarEnhance(long vmId, List<UVarEx> varList) {
```

```

        // TODO 自行实现逻辑
        return false;
    }

    @Override
    public UResult<UObject> getVarEnhance(long vmId, String varKey) {
        // TODO 自行实现逻辑
        return null;
    }
}

```

```

/**
 * 继承公共函数库抽象类
 */
public class SHFunctionSub extends SHFunction {

    @Override
    public void logout(ULogLevelEnum level, String msg) {
        // TODO 自行实现逻辑
    }

    @Override
    public boolean postEvent(long vmId, List<UVarEx> eventList) {
        // TODO 自行实现逻辑
        return false;
    }
}

```

3.2.2. 初始化

```

// 1. 创建实例
AbilityBase abilityBase = new AbilityBase();
AbsAbilityDriver absAbilityDriver = new AbilityDriver();
AbilityAuth abilityAuth = new AbilityAuth();
AbilityVmMgr abilityVmMgr = new AbilityVmMgr();
SHFunctionSub shFunctionSub = new SHFunctionSub();
// 2. 初始化
AbilityConfigInitializer.getInstance().initAbility(abilityBase, abilityVmMgr, abilityAuth,
absAbilityDriver);
shFunctionSub.init("延时变量", "时间点变量 key");

```

3.2.3. 创建虚拟机

```

AbilityVmMgrHelper.getInstance().createVm("上下文 id", "密钥");

```

3.2.4. 运行虚拟机

```

AbilityDriverHelper.getInstance().postEvent(123, Collections.emptyList());

```

3.2.5. 暂停虚拟机

```

AbilityVmMgrHelper.getInstance().pauseVm(123);

```


3.2.6. 恢复虚拟机

```
AbilityVmMgrHelper.getInstance().resumeVm(123);
```

3.2.7. 销毁虚拟机

```
// 1. 清理虚拟机实例的延时任务  
shFunctionSub.cleanupDelayTask(123);  
// 2. 销毁虚拟机  
AbilityVmMgrHelper.getInstance().freeVm(123);
```

3.2.8. 公共函数库调用-日志输出

```
shFunctionSub.logout(ULogLevelEnum.INFO, "日志");
```

3.2.9. 公共函数库调用-更新 UTC 时间戳

```
shFunctionSub.timeUpdate(System.currentTimeMillis());
```

3.2.10. 公共函数库调用-时间段函数

```
shFunctionSub.funcTimeRang(123, Collections.emptyList());
```

3.2.11. 公共函数库调用-阴历函数

```
shFunctionSub.funcLunarTime(123, Collections.emptyList());
```

3.2.12. 公共函数库调用-阳历函数

```
shFunctionSub.funcCalendar(123, Collections.emptyList());
```

3.2.13. 公共函数库调用-星期函数

```
shFunctionSub.funcWeekDay(123, Collections.emptyList());
```

3.2.14. 公共函数库调用-延时函数

```
shFunctionSub.funcDelay(123, Collections.emptyList());
```

3.2.15. 公共函数库调用-取消延时函数

```
shFunctionSub.funcCancelDelay(123, Collections.emptyList());
```