

Mini表情服务

修改历史

Version	Contributor	Date	Change Log
v1.0	彭钉	17/12/26	新增Mini表情服务接口说明
v1.1	彭钉	18/02/02	1、表情过渡功能支持可配; 2、新增PC表情仿真接口

概述

Mini服务是ROSE中一个服务应用，安装表情服务相应要安装ROSE中Master应用，master应用的安装参看《[master用户指导](#)》。目前，Mini的眼部是两个LCD屏，通过在屏上渲染眼睛动画，Mini可显示设计给出的预定义表情，本文档简要说明这些接口及sdk集成。

SDK

- protobuf 结构

```
enum ExpressFormat {
    GIF = 0;
    PNG = 1;
}
```

//表情信息

```
message ExpressInfo {  
    int32 id = 1; //id  
    string name = 2; //名称  
    int32 duration = 3; //时长  
    int32 frames = 5; //帧数  
    ExpressFormat format = 4;  
    int32 intrinsicWidth = 6; //图片宽度  
    bool customize = 7; //客制化  
}
```

//请求参数

```
message DoExpressReq {  
    string name = 1;  
    float speed = 2;  
    int32 repeat = 3;  
    bool tweenable = 4; 表情过渡  
    bool customize = 5; //客制化  
}
```

//请求参数

```
message StopExpressReq {  
    string name = 1;  
    int32 pid = 2;  
    bool customize = 3; //客制化  
}
```

```
message setFrameReq {  
    string name = 1; //名字  
    int32 frame = 2; //帧  
    bool customize = 3; //自定义  
}
```

- java 接口

```
/**
 * 获取机器人上支持的表情
 *
 * @return list of Express.ExpressInfo
 */
public @Nullable List<Express.ExpressInfo> getExpressList() ;

/**
 * 获取机器人上/sdcard/customize/sdcard的表情动画
 *
 * @return list of {@link Express.ExpressInfo}
 */
public List<Express.ExpressInfo> getCustomizeExpressList() ;

/**
 * do一个表情动作
 *
 * @param name 表情名称
 */
public void doExpress(@NonNull String name) ;

/**
 * do一个表情动作
 *
 * @param name 表情名称
 * @param loopCount 运行次数
 */
public void doExpress(String name, @IntRange(from = 0, to = Character.MAX_VALUE) int loopCount) ;

/**
 * do一个表情动作
 *
 * @param name 表情名称
 * @param priority 优先级
 * @param loopCount 运行次数
 */
```

```
public void doExpress(String name, @IntRange(from = 0, to = Character.MAX_VALUE) int loopCount,
    Priority priority) ;

/**
 * do一个表情动作
 *
 * @param name 表情名称
 * @param loopCount 运行次数
 * @param tweenable 是否使能表情过渡
 * @param priority 优先级 {@link Priority}
 */
public void doExpress(String name, @IntRange(from = 0, to = Character.MAX_VALUE) int loopCount,
    boolean tweenable, Priority priority) ;

/**
 * do表情动画，并监听表情动画状态
 *
 * @param name 表情名称
 * @param loopCount 运行次数
 * @param priority 优先级 {@link Priority}
 * @param listener 动画监听
 */
public void doExpress(String name,
    @IntRange(from = 0, to = Character.MAX_VALUE) final int loopCount, Priority priority,
    @Nullable final AnimationListener listener) ;

/**
 * do表情动画，并监听表情动画状态
 *
 * @param name 表情名称
 * @param loopCount 运行次数
 * @param tweenable 是否使能表情过渡
 * @param priority 优先级 {@link Priority}
 * @param listener 动画监听
 */
public void doExpress(final String name,
    @IntRange(from = 0, to = Character.MAX_VALUE) final int loopCount, final boolean tweenable,
```

```
final Priority priority, @Nullable final AnimationListener listener) ;

/**
 * do一个内置到机器人sdcard/customize/expresss卡中自定义的表情动画
 *
 * @param name 表情名称
 * @param priority priority
 * @param listener 表情动画回调 {@link AnimationListener}
 */
public void doCustomizeExpress(final String name, Priority priority,
    @Nullable final AnimationListener listener) ;

/**
 * do一个内置到机器人sdcard卡中自定义的表情动画
 *
 * @param name 表情名称
 * @param priority priority
 * @param loopCount 运行次数 大于等于1
 * @param listener 表情动画回调 {@link AnimationListener}
 */
public void doCustomizeExpress(final String name, final int loopCount, Priority priority,
    @Nullable final AnimationListener listener) ;

/**
 * do一个内置到机器人sdcard/customize/expresss中自定义的表情动画
 *
 * @param name 表情名称
 * @param priority priority
 * @param tweenable tweenable
 * @param loopCount 运行次数 大于等于1
 * @param listener 表情动画回调 {@link AnimationListener}
 */
public void doCustomizeExpress(final String name, final int loopCount, final boolean tweenable,
    Priority priority, @Nullable final AnimationListener listener) ;

/**
 * do表情动画，并监听表情动画状态
```

```
*  
* @param name 表情名称  
* @param listener 动画监听  
*/  
public void doExpress(String name, @Nullable final AnimationListener listener) ;  
  
/**  
* PC仿真调用，显示表情的指定帧  
*  
* @param name 表情名  
* @param frame 帧  
*/  
public void setFrame(String name, @IntRange(from = 0) int frame) ;  
  
/**  
* PC仿真调用，显示表情的指定帧  
*  
* @param name 表情名  
* @param frame 帧  
* @param customize customize  
*/  
public void setFrame(String name, @IntRange(from = 0) int frame, boolean customize) ;  
  
/**  
* 圆形进度条动画表情  
*  
* @param status 0 非充电 , 1 充电  
* @param progress 当前进度  
* @param priority 优先级, {@link Priority}  
* @param animated 是否动画  
* @param listener listener  
*/  
public void doProgressExpress(@IntRange(from = 0, to = 1) final int status,  
    final @IntRange(from = 0, to = 100) int progress, final boolean animated, Priority priority,  
    @Nullable final AnimationListener listener);  
  
/**
```

```
* 停止指定表情名的表情, 如果该表情动画结束, 则什么都不做
*
* @param express express
*/
public void stopExpress(@NonNull String express) ;
```

```
public interface AnimationListener {
    /**
     * Notifies the start of the animation.
     */
    void onAnimationStart();

    /**
     * Notifies the end of the animation.
     */
    void onAnimationEnd(int code);

    /**
     * @param loopNumber loopNumber
     */
    void onAnimationRepeat(int loopNumber);
}
```