

SMSServiceProvider

•
•
•
•
•
•

""=>

SMSServiceProvider

```
package com.fr.stable.fun;

import com.fr.json.JSONArray;
import com.fr.json.JSONObject;
import com.fr.stable.StringUtils;
import com.fr.stable.fun.mark.Selectable;

import java.util.List;
import java.util.Map;

/**
 * sms
 *
 * @author Lanlan
 */
public interface SMSServiceProvider extends Selectable {

    int CURRENT_LEVEL = 1;

    String XML_TAG = "SMSServiceProvider";

    /**
     *
     *
     * @return
     */
    Map<String, String> mapping();

    /**
     *
     *
     * @param mobile
     * @return
     */
    Response sendTest(String mobile);

    /**
     * FR
     *
     * @param template para
     * @param mobile
     * @param para
     * @param receiver
     * @return
     * @throws Exception
     */
}
```

```

Response send(String template, String mobile, JSONObject para, String receiver) throws Exception;

/**
 * FR
 *
 * @param template para
 * @param mobiles
 * @param params JSON
 * @param receivers /
 * @return
 * @throws Exception
 */
Response batchSendSMS(String template, List<String> mobiles, JSONArray params, List<String> receivers)
throws Exception;

/**
 * sms
 */
class Response {

    public final static String RES_STATUS_SUCCESS = "success";
    public final static String RES_STATUS_FAILED = "failed";

    private String status;
    private String msg;
    private JSONObject content;

    public static Response create(String status, String msg, JSONObject content) {
        if (StringUtils.isEmpty(status) || StringUtils.isEmpty(msg)) {
            return null;
        }
        if (content == null) {
            content = JSONObject.create();
        }
        return new Response(status, msg, content);
    }

    private Response(String status, String msg, JSONObject content) {
        this.status = status;
        this.msg = msg;
        this.content = content;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public String getMsg() {
        return msg;
    }

    public void setMsg(String msg) {
        this.msg = msg;
    }

    public JSONObject getContent() {
        if (!content.has("status")) {
            content.put("status", status);
        }
        if (!content.has("msg")) {
            content.put("msg", msg);
        }
        return content;
    }

    public void setContent(JSONObject content) {
        this.content = content;
    }
}

```

```

    }
}

/**
 * sms
 *
 * @author Lanlan
 * @date 2019/1/29
 */
interface Listener {

    /**
     *
     *
     * @param text      para
     * @param mobiles
     * @param params    JSON
     * @param receivers
     */
    void beforeSend(String text, List<String> mobiles, JSONArray params, List<String> receivers);

    /**
     *
     *
     * @param text      para
     * @param mobiles
     * @param params    JSON
     * @param receivers
     * @param response
     */
    void afterSend(String text, List<String> mobiles, JSONArray params, List<String> receivers, Response
response);
}
}

```

```

<extra-core>
  <SMSServiceProvider class="com.fr.plugin.third.sms.SRGT"/>
</extra-core>

```

<https://code.fanruan.com/fanruan/demo-third-sms>

	10	#taskname##time#
	11	#name##order_number#
	13	#webname##m##n#
	14	#taskname##time#
	15	#taskname##time#
	16	#Verifiecode#10

17	#webname##m##n#
18	
19	#taskname##time#
20	#verifiecode#10
49	#webname##logsize#M
50	#webname##clustername##m##n#
51	#task##time#
53	#task##time#
54	#name##task#
63	#node_id#
64	#node_id#
89	#nodename##node1name#jar
90	#nodename#
125	#praname#
127	#praname##startpeople##starttime#
134	Redis#ip_port#
135	
136	#webname##clustername#--
239	#warningname##templatePath#
264	#job_name##year##month##day##hr_min##c_hr##c_min##c_sec# #success_basetable##total_basetable# #success_dataset# /total_dataset# #success_relation##total_relation#
265	
266	
269	
276	[#taskname#]#time#
367	#node_name#
368	#node_name#WEB-INF#directory#
998	#nodename1##nodename2#[TCP:7800,7810, 7820,7830, 7840, 7850,7860,7870][UDP:4558865536]
999	#node_name##type#