

一、定时器

1.1 添加定时器

```
/**
 * @method timer_add
 * 增加一个timer。
 * @annotation ["scriptable:custom", "static"]
 * @param {timer_func_t} on_timer timer回调函数。
 * @param {void*} ctx timer回调函数的上下文。
 * @param {uint32_t} duration_ms 时间。
 *
 * @return {uint32_t} 返回timer的ID, 0表示失败。
 */
uint32_t timer_add(timer_func_t on_timer, void* ctx, uint32_t duration_ms);
```

示例：

```
static ret_t on_timer(const timer_info_t* timer) {
    return update_progress_bar(WIDGET(timer->ctx));
}

...

widget_t* progress_bar = progress_bar_create(win, 10, 80, 168, 20);
widget_set_value(progress_bar, 40);
timer_add(on_timer, progress_bar, 500);
...
```

1.2 删除定时器

```
/**
 * @method timer_remove
 * 删除指定的timer。
 * @annotation ["scriptable", "static"]
 * @param {uint32_t} timer_id timerID。
 *
 * @return {ret_t} 返回RET_OK表示成功, 否则表示失败。
 */
ret_t timer_remove(uint32_t timer_id);
```