### 手机验证短信设计与代码实现

# 手机验证短信设计与代码实现 >1. 时效限制: [5-10min] >2. 使用次数限制: 1次 >3. IP次数限制: 防止恶意刷手机验证码短信 >4. 手机号限制: 防止短信轰炸 >5. 跨域请求限制: 进一步限制恶意刷短信 >6. 验证码验证: 进一步限制恶意刷短信

**手机验证短信设计与代码实现**

**总体思路**

1. 时效限制: [5-10min]
2. 使用次数限制: 1次
3. IP次数限制: 防止恶意刷手机验证码短信
4. 手机号限制: 防止短信轰炸
5. 跨域请求限制: 进一步限制恶意刷短信
6. 验证码验证: 进一步限制恶意刷短信

**数据库设计 (CheckCode)**

ID mobile(手机号)   
checkCode(验证码)   
ip(IP地址)   
createAt(创建时间)   
expireAt(时效时间)   
isUse(是否使用)   
usingAt(使用时间)

**具体代码实现(mongoose + nodejs 代码不完全正确，主要see思路吧)**

1. 查询90s内是否发送过，如果存在，需要等待 90-(已发送时间)s

[?](http://www.open-open.com/lib/view/open1411819249531.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | var mobile = '155\*\*\*\*\*111',      ip = '127.0.0.1',      now = Date.now();        CheckCode.findOne({mobile: mobile, expireAt:{$gt: now}}, function(err, checkCode) {          if (err) {              //错误了          } else if(checkCode) {              var now = Date.now(),                  diffSeconds = parseInt((now - checkCode.createAt.getTime())/1000, 10);              if (diffSeconds < 90) {                  //时间间隔太小，老弟你刷短信纳是吧，果断拒绝你              } else {                  checkCode.setIsUsed();  //设置为已经使用过                  newCheckCodeByMobile(mobile, {ip:ip}, done);              }          } else {              newCheckCodeByMobile(mobile, {ip:ip}, done);          }      } |

1. 查询手机号码接收次数，如果太多明显要轰炸别人，让我们背黑锅呀

[?](http://www.open-open.com/lib/view/open1411819249531.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | var end = now,      begin = now - 24 \* 60 \* 60 \* 1000;      CheckCode.count({mobile: mobile, createAt: {$gt: begin, $lt: end}}, function(err, count){          if (err) {              //错误          } else if(count >3) {              //老大，都给你手机号发3次了还收不到，你是要用短信轰炸别人呢还是真收不到，果断舍弃你这用户把          } else {              newCheckCodeByMobile()  //          }      }); |

1. 查询这个Ip发送了多少次了， 如果太多明显是来浪费我们财产来了，短信是要钱的呀老大

[?](http://www.open-open.com/lib/view/open1411819249531.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | CheckCode.count({ip:ip, createAt: {$gt: begin, $lt: end}}, function(err, count){          if (err) {              //err          } else if (count >6) {              //老大，你这个Ip都浪费了我5毛钱了，你还不甘心呀，算了，放弃你了          }  else {              newCheckCodeByMobile() //          }      });        //生成手机验证码      function newCheckCodeByMobile(mobile, options, callback) {          if (arguments.length === 2 ) {              callback = options;              options = {};          }          var ip = options.ip;          var now = Date.now(),          expireAt = now + 60 \* 1000 \* 5; //5分钟后时效          CheckCode.save({mobile: mobile, ip:ip, checkCode:"123456", isUse:false}, callback);      } |

1. 限制跨域提交

[?](http://www.open-open.com/lib/view/open1411819249531.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | //渲染页面时   var oldCrsf = "12345679";   req.session.\_csrf = oldCrsf;     //接受提交   var \_crsf = req.body.\_crsf;   if (\_crsf !== req.session.\_csrf) {      res.send(302);   } else {      // ok   } |

1. 验证码限制 (同跨域提交思路)

来自：ht