|  |  |
| --- | --- |
| 文档编号 | V1.0 |
| 文档负责人 |  |
| 文档名称 | 用户注册&登录概要设计 |

用户注册&登录概要设计

目录

[目录 2](#_Toc417848221)

[第1章 文档记录 3](#_Toc417848222)

[第2章 引言 4](#_Toc417848223)

[2.1 目的 4](#_Toc417848224)

[2.2 目标读者 4](#_Toc417848225)

[2.3 名词解释 4](#_Toc417848226)

[2.4 文档约定 4](#_Toc417848227)

[第3章 业务流程 5](#_Toc417848228)

[3.1 下发短信验证码 5](#_Toc417848229)

[3.2 验证短信验证码 6](#_Toc417848230)

[第4章 数据库设计 7](#_Toc417848231)

[4.1 短信验证码表 7](#_Toc417848232)

[4.2 短信验证码流水表 8](#_Toc417848233)

[第5章 接口设计 11](#_Toc417848234)

[5.1 账户交易（mvcode\_service） 11](#_Toc417848235)

[5.1.1 下发短信验证码 11](#_Toc417848236)

[5.1.2 验证短信验证码 12](#_Toc417848237)

[第6章 安全设计 13](#_Toc417848238)

# 文档记录

# 引言

## 目的

## 目标读者

## 名词解释

## 文档约定

是否必填项说明

|  |  |  |
| --- | --- | --- |
| 选项 | 全称 | 说明 |
| R | Required | 必需 |
| C | Conditional | 有条件必填 |
| O | Optional | 可选项 |

# 业务流程

## 用户登录创建session



## session验证

# Session存储

## Cache选择

|  |  |  |
| --- | --- | --- |
| cache组件 | memcached | redis |
| 特点 | 纯内存型的缓存系统，读写速度快，性能好。 | 内存缓存，支持主从复制，支持持久化。  支持主从复制 |
| 选择 |  | 建议选择redis |

当前主流的缓存memcached和redis，考虑到缓存机器出现问题或服务被重启，session存在缓存中会丢失，或者内存不够缓存中的数据会被淘汰。建议选择redis做session存储。原因如下：

1. 对缓存机制的ha和是否对特殊数据需要进行持久化无需特殊处理，直接使用redis的自有功能即可。使用memcached，如果缓存crash了则所有用户的session全部丢失。
2. 除session外其他场景也可用redis做缓存。Redis的数据结构比memcached丰富，可以提高开发效率。

## session id生成

生成的规则需要有两点：

1. 不重复
2. 随机性，防止有人猜测规律进行攻击

生成方式：MD5(UUID+ unix timestamp)，unix timestamp为当前时间的秒数。

## Session内容

Session的内容以k-v形式保存，多个k-v以&连接。最终存储到redies时将session内容作为value，以“usr\_session\_”+session\_id作为key存储。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| version | string | 当前1.0。版本号，不同版本内容字段可能不同，方便后续升级兼容处理 |
| userid | string | 用户名 |
| client\_info | string | 用户登陆客户端信息 |
| client\_ip | string | 用户登录的IP |
| state | int | 用户的登录状态 |
| logintime | timestamp | 用户登陆记录session的时间 |
| channel\_id | string | 渠道号  1-PC  2-APP |
| sign | string | session内容签名  MD5(version+ userid+ state+ logintime+ client\_ip+session\_id+固定的盐) |

# 接口设计

## CMS模块（Cache Management Service）

缓存管理模块，以redis做为缓存，对外提供session服务、token缓存服务功能

### 创建session

接口说明

xxxx/ session.hessian的createSession方法

功能说明

创建用户session

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| userid | string | R | 用户名 |
| client\_info | string | C | 用户登陆客户端信息 |
| client\_ip | string | R | 用户登录的IP |
| channel\_id | int | R | 渠道号  1-PC  2-APP |
|  |  |  |  |

输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| session\_id | string | R | 创建的session的id值，后面验证时需传入id值 |

内部逻辑

1. 检查输入参数合法性
2. 获取当前时间
3. 生成session\_id和签名信息
4. 在配置中获取session过期时间，保存session内容
5. 返回session\_id

### 验证session

接口说明

xxxx/ session.hessian的checkSession方法

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| session\_id | string | R | session的ID |
| userid | string | R | 用户名 |
| channel\_id | int | C | 渠道号  1-PC  2-APP |
|  |  |  |  |

输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
|  |  |  |  |

内部逻辑

1. 检查输入参数合法性
2. 根据session\_id查询session内容
3. 如果获取内容成功，校验签名是否正确
4. 校验用户信息是否正确，验证session中登陆时间是否过期
5. 返回验证结果

# 安全设计