

# 个人安全中心设计文档

---

## 个人安全中心设计文档

需求

总体设计

项目架构

项目框架

包名定义

名词定义

表单流的安全控制

nonce 生成规则

nonce 的作用

功能设计

通用功能

登录

获取配置

帐号信息

基本信息

联系信息

帐号信息

安全设置

获取安全设置

修改登录密码

设置安全手机

设置安全邮箱

联合登录设置

获取联合登录设置

绑定支付宝

修改密码

## 需求

---

参考：

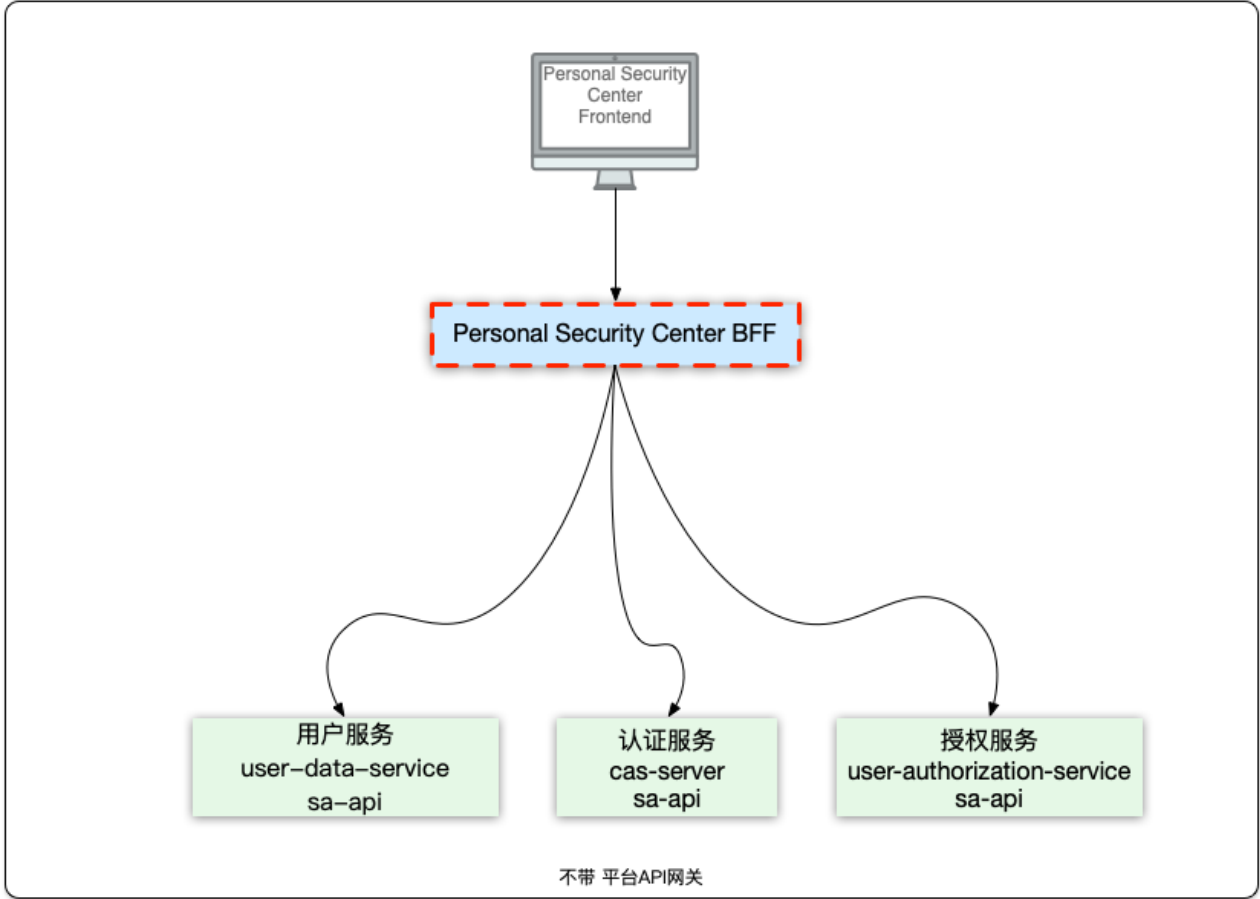
- 需求文档：统一身份认证需求v009-2019.04.22 <https://supwisdom.coding.net/p/cas-server-doc/attachment/369676/preview/431930> 文档中，个人安全中心 部分
- 交互原型：智慧校园-认证服务交互原型&文档 V1\_5.6(输出) <https://supwisdom.coding.net/p/cas-server-doc/attachment/443038/preview/474149> 原型中，个人安全中心 部分

## 总体设计

---

### 项目架构

整体架构，及当前项目的定位



Personal Security Center BFF，是面向 personal-security-center 个人安全中心 的后端项目 根据实际的业务需求，将后端分散在不同微服务中的API聚合，提供统一的访问入口 同时，可以进行 访问控制、应用缓存

# 项目框架



## 包名定义

com.supwisdom.institute.personal.security.center.bff

configuration 项目配置类

security 认证、访问控制

listener 监听器

utils 工具类（如，获取当前登录用户对象）

base 基类（如，entity 基类、vo 基类等）

entity 实体类

vo 值对象（如，请求、响应对象）

remote 远程服务API 调用类（基于 FeignClient 实现，返回 com.alibaba.fastjson.JSONObject 对象）

service 业务类（@Service 注解，调用 remote 获取到 JSONObject 对象后，转换成 POJO 对象，然后由 Controller 使用）

controller 控制层（@RestController 注解，调用 service 获取到 POJO 对象后，转换成 vo，返回响应）

## 名词定义

---

- 会话 token，确保当前用户为合法用户，每次与后端的请求交互，都必须带上该 token
- 表单 token，一次性token，确保用户在使用表单（特别是 表单流 中）提交数据时，处于安全、有效状态

- 表单流，一系列表单提交操作，且表单之间存在先后依赖。

由于，在前后端分离的模式下，每个表单的请求都认为是独立的，很容易造成用户绕过某个步骤直接进入其他步骤进行提交

故，采用 表单 token 的方式进行控制，即用户在提交当前表单的数据时，必须携带上 上一步骤的操作结果返回的 表单 token，当前请求须验证该 表单 token 是否有效

而且，该 token 使用后立即失效，若未使用，则只保留5分钟有效期

## 表单流的安全控制

---

为了保证前端提交数据到后端接口的安全性（防止暴力攻击、防止CSRF等）

前端在展示表单页面或进入下一步操作时，需要换取 一次性票据 nonce，而在提交请求时，须带上该票据

1. 首次展示表单页面，通过请求一个 init 的接口获取 nonce
2. 进入下一步页面，需要由上一步页面中的提交成功时，返回的结果中获取 nonce

请求失败时，返回错误信息的同时，还须返回一个 新的nonce

请求失败次数大于3次时，不再返回 新的nonce，即无法继续提交，须重新打开表单页面并获取 nonce

## nonce 生成规则

当前登录用户 + 时间戳 + 密钥

## nonce 的作用

防止 CSRF 攻击

# 功能设计

## 通用功能

### 登录

### 获取配置

获取管理后台的配置，联合登录配置、安全设置配置、安全设置流程配置、激活方式配置等

- 根据安全设置配置，控制 帐号信息下 安全手机、安全邮箱及相关操作 是否显示
- 根据安全设置流程配置，控制 修改密码、绑定安全手机、绑定安全邮箱 的操作步骤
- 根据联合登录配置，控制 联合登录设置下 支付宝、微信、QQ及相关操作 是否显示

## 帐号信息

### 基本信息

#### 1. 后端接口设计

获取基本信息 GET /users/info

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "name": "张某某",
    "sex": "男",
    "nation": "汉族",
    "identityType": "身份证",
    "identityNo": "310*****90",
    "imageUrl": "xxxxx",
    "mobile": "",
    "emailAddress": "",
    "accounts": [
      {
        "id": "1",
        "username": "帐号",
        "identity": "身份(教师、本科生、研究生)",
        "organizationName": "所属机构",
        "expireDate": "2020-12-31",
        "isActivate": "是否激活(是、否)",
        "status": "状态(正常、冻结)"
      }
    ]
  }
}
```

```
    },
    {.....}
  ]
}
}
```

上传头像 POST /users/info/portrait

请求 (form) :

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "imageUrl": "xxxxx"
  }
}
```

修改头像 PUT /users/info/portrait

请求体 (application/json) :

```
{
  "imageUrl": "xxxxx"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "message": "保存成功"
  }
}
```

## 联系信息

### 1. 后端接口设计

获取联系信息

见，基本信息 - 获取基本信息接口。

手机、邮箱、地区住址；同时，返回 安全设置配置（前端可根据配置，控制 安全手机、安全邮箱 的操作的显示）

修改地区住址

所在地区、街道地址

修改安全手机

见，安全设置 - 设置安全手机

修改安全邮箱

见，安全设置 - 设置安全邮箱

## 帐号信息

一个用户下，存在多个帐号的情况，故显示帐号列表。

## 安全设置

### 获取安全设置

获取当前用户的 安全等级、手机绑定状态、邮箱绑定状态

同时，返回 安全设置配置（前端可根据配置，控制 安全手机、安全邮箱 及相关操作的显示）

#### 1. 后端接口设计

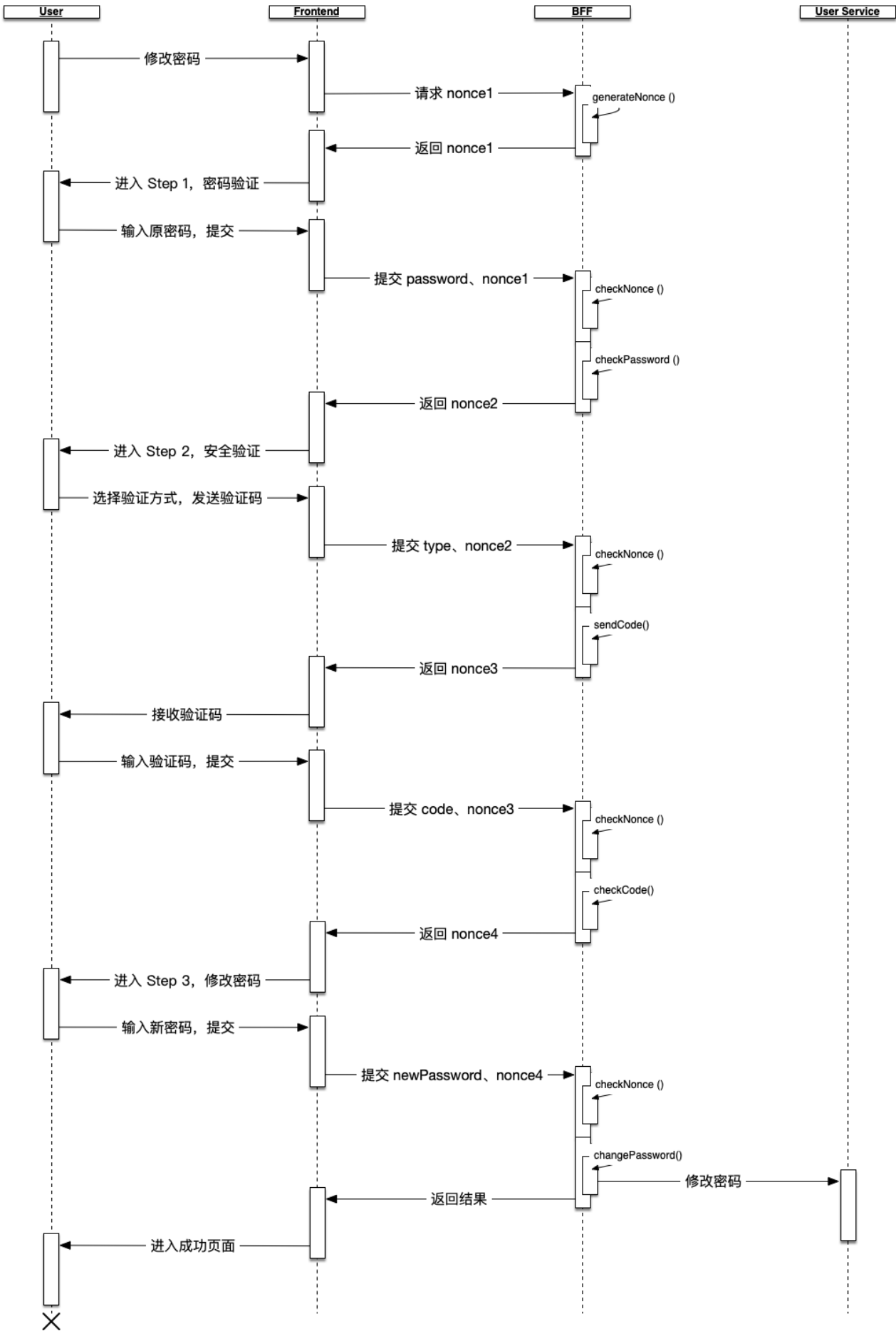
获取安全设置 GET /users/security

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "securitySettingConfigs": [

    ],
    "securityLevel": 3, // 1 低, 2 中, 3 高
    "mobileBinded": true,
    "emailAddressBinded": false
  }
}
```

修改登录密码





## 1. 功能流程设计：

### STEP 1

请求后端，/users/security/password/initPassword，获取 安全设置流程配置，根据 当前用户 + 时间戳 + 步骤号 (1) + 密钥，生成 nonce

前端展示页面

用户输入原密码，提交 (原密码，步骤号 (1) , nonce)

请求后端，/users/security/password/checkPassword，校验 nonce，校验 原密码；校验正确，根据安全设置流程配置 确定步骤号 (2 或 3) ，根据 当前用户 + 时间戳 + 步骤号 (2 或 3) + 密钥，生成 nonce；否则，返回错误信息

前端根据结果，提示错误 或 根据步骤号 导航到 STEP 2 或 3

### STEP 2

前端展示页面

用户选择验证方式，点击发送验证码 (控制点击频率 60s) ，提交 (步骤号 (2) , nonce)

请求后端，/users/security/password/sendCode，校验 nonce；校验正确，发送验证码到当前用户的手机或邮箱；否则，返回错误信息

前端提示发送结果

用户输入验证码，提交 (验证码，步骤号 (2) , nonce)

请求后端，/users/password/checkCode，校验 nonce，校验 验证码；校验正确，根据 当前用户 + 时间戳 + 步骤号 (3) + 密钥，生成 nonce；否则，返回错误信息

前端根据结果，提示错误 或 根据步骤号 导航到 STEP 3

### STEP 3

前端展示页面

用户输入 新密码，确认密码，提交 (新密码、确认密码、步骤号 (3) , nonce)

请求后端，/users/security/password/changePassword，校验 nonce，校验密码是否符合密码策略；校验正确，保存新密码，返回成功信息；否则，返回错误信息

前端根据结果，提示信息

## 2. 后端接口设计

初始化 GET /users/security/password/initPassword

获取初始信息 (如: nonce)

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "nonce": "aaa",
    "step": 1,
    "message": "初始化成功"
  }
}
```

验证登录密码 POST /users/security/password/checkPassword

提交登录密码，进行验证，并根据安全设置流程配置，确定步骤（step：2 或 3）

请求体（application/json）：

```
{
  "nonce": "aaa",
  "password": "pppooo"
}
```

响应（application/json）：

```
{
  "acknowledged": true,
  "data": {
    "nonce": "bbb",
    "step": step,
    "message": "验证成功"
  }
}
```

发送验证码 POST /users/security/password/sendCode

根据验证方式（checkType：mobile、emailAddress），将验证码发送到用户的原手机或原邮箱

请求体（application/json）：

```
{
  "nonce": "bbb",
  "checkType": "mobile"
}
```

响应（application/json）：

```
{
  "acknowledged": true,
  "data": {
    "nonce": "ccc",
    "step": 2,
    "message": "发送成功"
  }
}
```

验证验证码 POST /users/security/password/checkCode

提交验证码

请求体 (application/json) :

```
{
  "nonce": "ccc",
  "code": "1234"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "nonce": "xxx",
    "step": 3,
    "message": "验证成功"
  }
}
```

修改登录密码 POST /users/security/password/changePassword

修改登录密码

请求体 (application/json) :

```
{
  "nonce": "xxx",
  "newPassword": "qqqwww",
  "confirmPassword": "qqqwww"
}
```

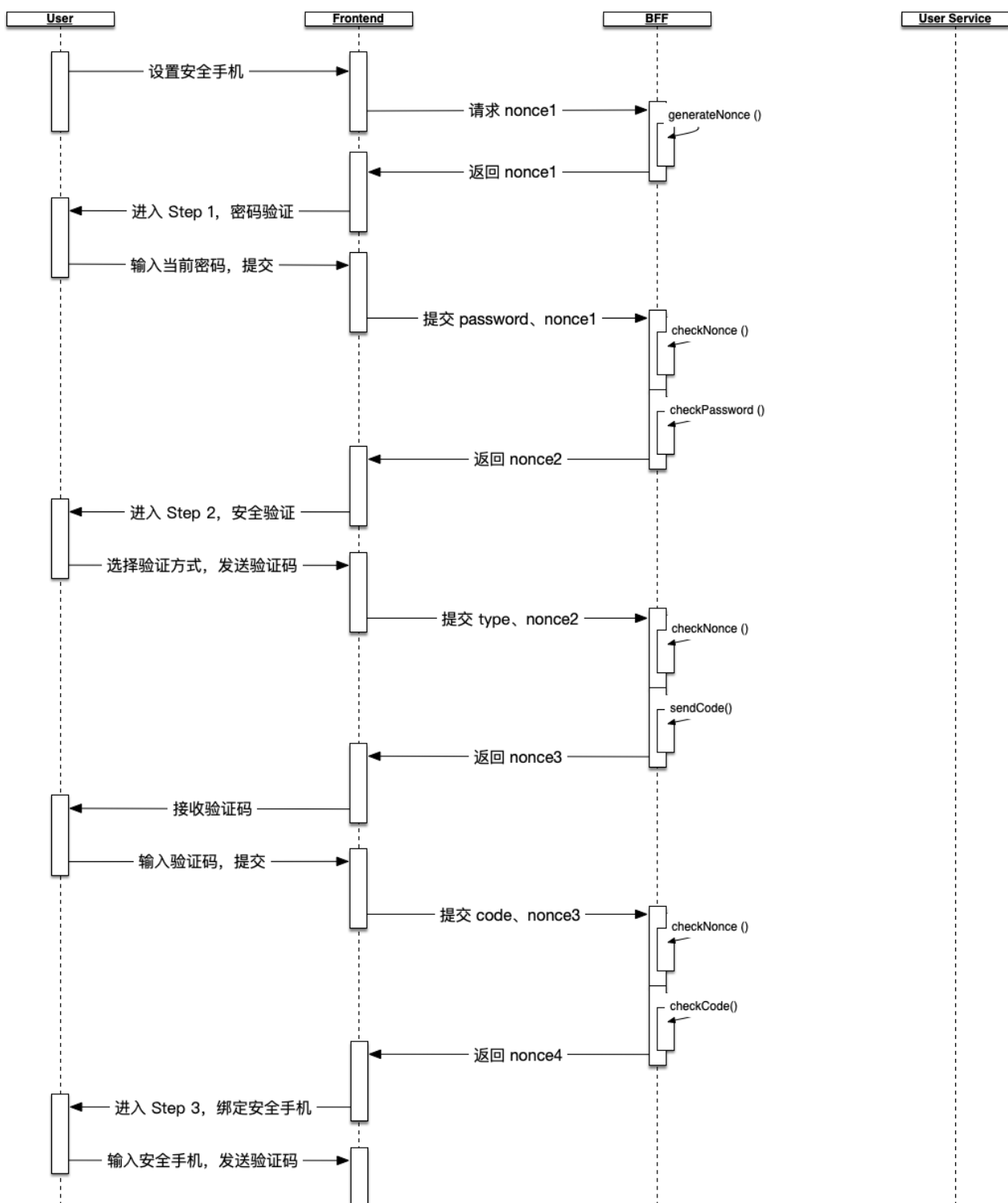
响应 (application/json) :

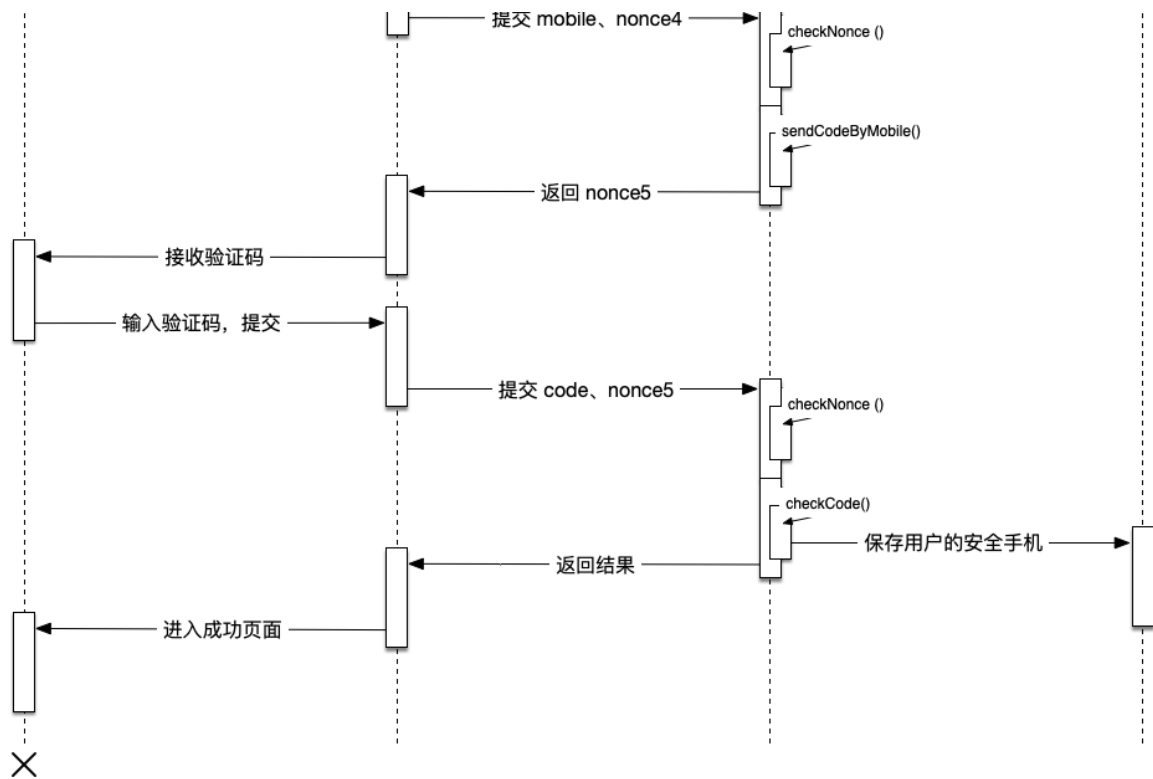
```

{
  "acknowledged": true,
  "data": {
    "message": "修改成功"
  }
}

```

## 设置安全手机





## 1. 功能流程设计

### STEP 1

请求后端，/users/security/mobile/initMobile，获取 安全设置流程配置，根据 当前用户 + 时间戳 + 步骤号（1）+ 密钥，生成 nonce

前端展示页面

用户输入原密码，提交（原密码，步骤号（1），nonce）

请求后端，/users/security/mobile/checkPassword，校验 nonce，校验 原密码；校验正确，根据安全设置流程配置 确定步骤号（2 或 3），根据 当前用户 + 时间戳 + 步骤号（2 或 3）+ 密钥，生成 nonce；否则，返回错误信息

前端根据结果，提示错误 或 根据步骤号 导航到 STEP 2 或 3

### STEP 2

前端展示页面

用户选择验证方式，点击发送验证码（控制点击频率 60s），提交（步骤号（2），nonce）

请求后端，/users/security/mobile/sendCode，校验 nonce；校验正确，发送验证码到当前用户的手机或邮箱；否则，返回错误信息

前端提示发送结果

用户输入验证码，提交（验证码，步骤号（2），nonce）

请求后端，/users/security/mobile/checkCode，校验 nonce，校验 验证码；校验正确，根据 当前用户 + 时间戳 + 步骤号（3）+ 密钥，生成 nonce；否则，返回错误信息

前端根据结果，提示错误 或 根据步骤号 导航到 STEP 3

### STEP 3

#### 前端展示页面

用户输入 新手机，点击发送验证码（控制点击频率 60s），提交（新手机、步骤号（3），nonce）

请求后端，/users/security/mobile/sendCodeByMobile，校验 nonce；校验正确，发送验证码到新手机；否则，返回错误信息

#### 前端提示发送结果

用户在收到验证码后，输入验证码，提交（新手机、验证码、步骤号（3），nonce）

请求后端，/users/security/mobile/checkCodeBindMobile，校验 nonce，校验 验证码，校验手机号是否与之前发送验证码的手机号一致；校验正确，保存新手机，返回成功信息；否则，返回错误信息

前端根据结果，提示信息

## 2. 后端接口设计

初始化 GET /users/security/mobile/initMobile

获取初始信息（如：nonce）

响应（application/json）：

```
{
  "acknowledged": true,
  "data": {
    "nonce": "aaa",
    "step": 1,
    "message": "初始化成功"
  }
}
```

验证登录密码 POST /users/security/mobile/checkPassword

提交登录密码，进行验证，并根据安全设置流程配置，确定步骤（step：2 或 3）

请求体（application/json）：

```
{
  "nonce": "aaa",
  "password": "pppooo"
}
```

响应（application/json）：

```
{
  "acknowledged": true,
  "data": {
    "nonce": "bbb",
    "step": step,
    "message": "验证成功"
  }
}
```

发送验证码 POST /users/security/mobile/sendCode

根据验证方式 (checkType: mobile、emailAddress) , 将验证码发送到用户的原手机或原邮箱  
请求体 (application/json) :

```
{
  "nonce": "bbb",
  "checkType": "mobile"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "nonce": "ccc",
    "step": 2,
    "message": "发送成功"
  }
}
```

验证验证码 POST /users/security/mobile/checkCode

提交验证码

请求体 (application/json) :

```
{
  "nonce": "ccc",
  "code": "1234"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "nonce": "xxx",
    "step": 3,
    "message": "验证成功"
  }
}
```

发送验证码到新手机 POST /users/security/mobile/sendCodeByMobile

用户输入新手机后，发送验证码到新手机

请求体 (application/json) :

```
{
  "nonce": "xxx",
  "mobile": "13788880000"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "nonce": "yyy",
    "step": 3,
    "message": "发送成功"
  }
}
```

验证验证码并绑定新手机 POST /users/security/mobile/checkCodeBindMobile

用户输入新手机的验证码，提交进行新手机绑定

请求体 (application/json) :

```
{
  "nonce": "xxx",
  "mobile": "13788880000",
  "code": "0987"
}
```

响应 (application/json) :

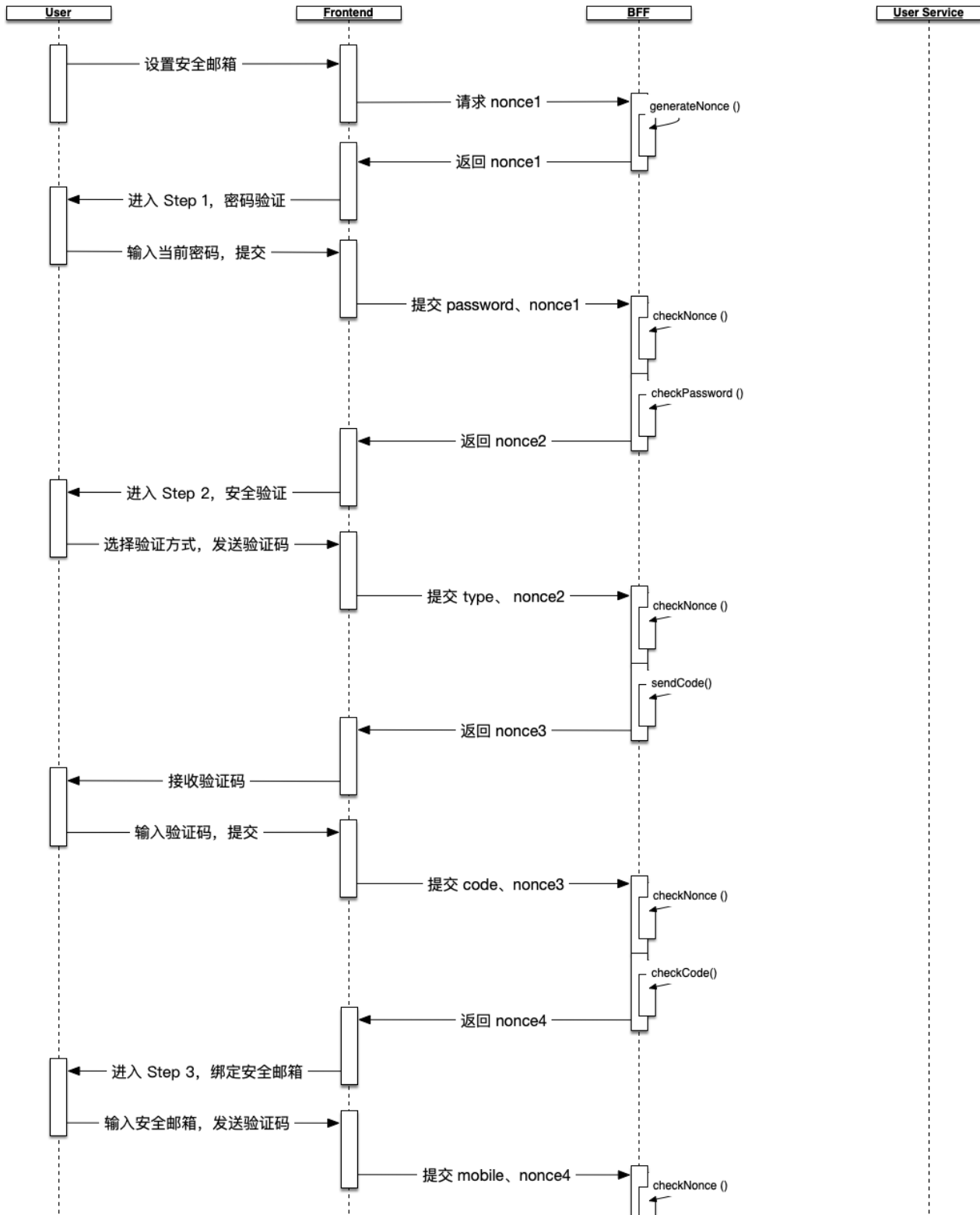


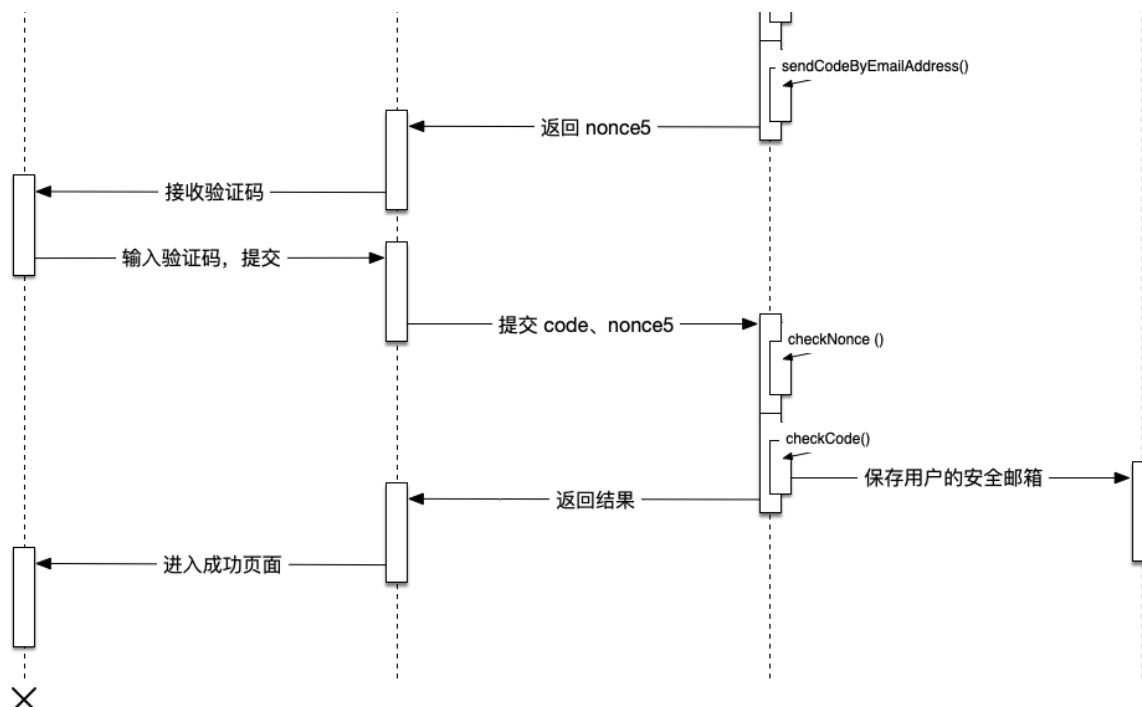
```

{
  "acknowledged": true,
  "data": {
    "message": "绑定成功"
  }
}

```

## 设置安全邮箱





## 1. 功能流程设计

### STEP 1

请求后端，/users/security/emailAddress/initEmailAddress，获取 安全设置流程配置，根据 当前用户 + 时间戳 + 步骤号（1）+ 密钥，生成 nonce

前端展示页面

用户输入原密码，提交（原密码，步骤号（1），nonce）

请求后端，/users/security/emailAddress/checkPassword，校验 nonce，校验 原密码；校验正确，根据安全设置流程配置 确定步骤号（2 或 3），根据 当前用户 + 时间戳 + 步骤号（2 或 3）+ 密钥，生成 nonce；否则，返回错误信息

前端根据结果，提示错误 或 根据步骤号 导航到 STEP 2 或 3

### STEP 2

前端展示页面

用户选择验证方式，点击发送验证码（控制点击频率 60s），提交（步骤号（2），nonce）

请求后端，/users/security/emailAddress/sendCode，校验 nonce；校验正确，发送验证码到当前用户的手机或邮箱；否则，返回错误信息

前端提示发送结果

用户输入验证码，提交（验证码，步骤号（2），nonce）

请求后端，/users/security/emailAddress/checkCode，校验 nonce，校验 验证码；校验正确，根据 当前用户 + 时间戳 + 步骤号（3）+ 密钥，生成 nonce；否则，返回错误信息

前端根据结果，提示错误 或 根据步骤号 导航到 STEP 3

### STEP 3

前端展示页面

用户输入 新邮箱，点击发送验证码（控制点击频率 60s），提交（新手机、步骤号（3），nonce）

请求后端，/users/security/emailAddress/sendCodeByEmailAddress，校验 nonce；校验正确，发送验证码到新邮箱；否则，返回错误信息

前端提示发送结果

用户在收到验证码后，输入验证码，提交（新邮箱、验证码、步骤号（3），nonce）

请求后端，/users/security/emailAddress/checkCodeBindEmailAddress，校验 nonce，校验验证码，校验邮箱是否与之前发送验证码的邮箱一致；校验正确，保存新邮箱，返回成功信息；否则，返回错误信息

前端根据结果，提示信息

## 2. 后端接口设计

初始化 GET /users/security/emailAddress/initEmailAddress

获取初始信息（如：nonce）

响应（application/json）：

```
{
  "acknowledged": true,
  "data": {
    "nonce": "aaa",
    "step": 1,
    "message": "初始化成功"
  }
}
```

验证登录密码 POST /users/security/emailAddress/checkPassword

提交登录密码，进行验证，并根据安全设置流程配置，确定步骤（step：2 或 3）

请求体（application/json）：

```
{
  "nonce": "aaa",
  "password": "pppooo"
}
```

响应（application/json）：

```
{
  "acknowledged": true,
  "data": {
    "nonce": "bbb",
    "step": step,
    "message": "验证成功"
  }
}
```

发送验证码 POST /users/security/emailAddress/sendCode

根据验证方式 (checkType: mobile、emailAddress) , 将验证码发送到用户的原手机或原邮箱  
请求体 (application/json) :

```
{
  "nonce": "bbb",
  "checkType": "mobile"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "nonce": "ccc",
    "step": 2,
    "message": "发送成功"
  }
}
```

验证验证码 POST /users/security/emailAddress/checkCode

提交验证码

请求体 (application/json) :

```
{
  "nonce": "ccc",
  "code": "1234"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "nonce": "xxx",
    "step": 3,
    "message": "验证成功"
  }
}
```

发送验证码到新邮箱 POST /users/security/emailAddress/sendCodeByEmailAddress

用户输入新邮箱后，发送验证码到新邮箱

请求体 (application/json) :

```
{
  "nonce": "xxx",
  "emailAddress": "bind@supwisdom.com"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "nonce": "yyy",
    "step": 3,
    "message": "发送成功"
  }
}
```

验证验证码并绑定新邮箱 POST /users/security/emailAddress/checkCodeBindEmailAddress

用户输入新邮箱的验证码，提交进行新邮箱绑定

请求体 (application/json) :

```
{
  "nonce": "xxx",
  "emailAddress": "bind@supwisdom.com",
  "code": "0987"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "message": "绑定成功"
  }
}
```

## 联合登录设置

### 获取联合登录设置

获取当前用户的 支付宝绑定状态、微信绑定状态、QQ绑定状态

同时，返回 联合登录配置（前端可根据配置，控制 支付宝、微信、QQ 及相关操作的显示）

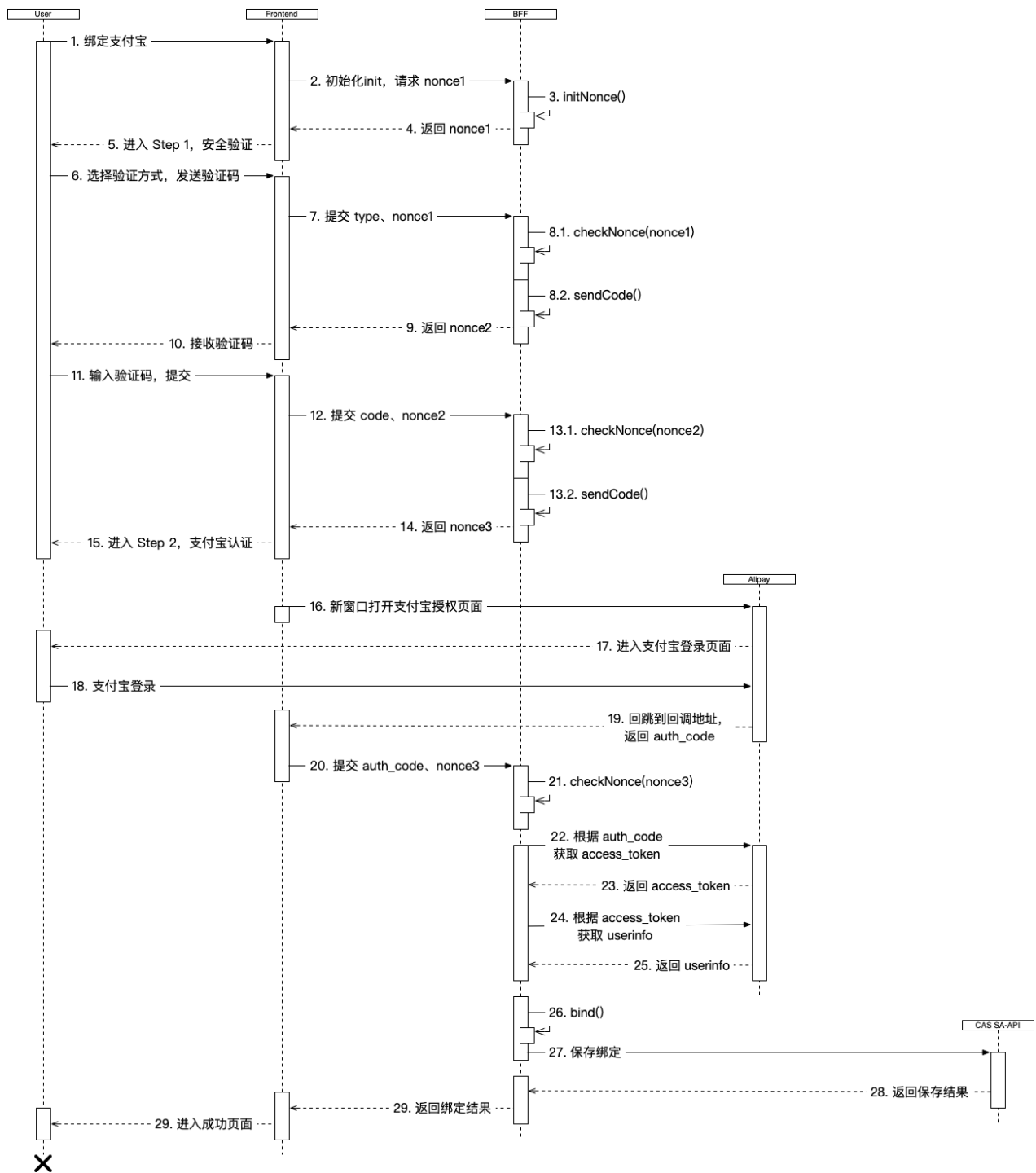
#### 1. 后端接口设计

获取安全设置 GET /users/federation

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "federationModeSetting": {
      "alipayEnabled": false,      // 支付宝
      "qqEnabled": false,         // QQ
      "openweixinEnabled": false, // 微信
      "workweixinEnabled": false  // 企业微信
    },
    "alipayBinded": true,
    "qqBinded": false,
    "openweixinBinded": false
  }
}
```

### 绑定支付宝



## 1. 功能流程设计

### STEP 1 安全验证

#### 点击绑定支付宝

请求后端, `/users/federation/alipay/initAlipay`, 根据 当前用户 + 时间戳 + 步骤号 (1) + 密钥, 生成 nonce1

#### 前端展示页面 (安全验证页面)

用户选择验证方式, 点击发送验证码 (控制点击频率 60s), 提交 (步骤号 (1), nonce1)

请求后端, `/users/federation/alipay/sendCode`, 校验 nonce; 校验正确, 发送验证码到当前用户的手机或邮箱, 生成 nonce2; 否则, 返回错误信息

前端提示发送结果

用户接收到验证码后，输入验证码，提交（验证码，步骤号（1），nonce2）

请求后端，/users/federation/alipay/checkCode，校验 nonce，校验 验证码；校验正确，生成 nonce3；否则，返回错误信息

前端根据结果，提示错误 或 导航到 STEP 2

## STEP 2 支付宝认证

前端展示页面（提示，即将跳转到支付宝认证页面，期间需要使用支付宝帐号登录 或 支付宝APP 扫码）

前端在新窗口中，**跳转到CAS联合登录接口**

（<https://cas.supwisdom.com/cas/federation/federated/alipay?redirectUri=https://example.com&state=1>），随后，浏览器自动跳转，打开支付宝的用户授权地址

用户登录 或 扫码后，CAS联合登录接口生成 federatedCode，并返回到个人安全中心的回调页面

前端获取到回调地址中的 federatedCode，将 federatedCode 提交到后端（federatedCode，nonce3）

请求后端，/users/federation/alipay/bindAlipay，校验 nonce；校验正确，根据 federatedCode 获取到CAS联合登录的信息（如：federatedId）；校验该支付宝帐号是否已和其他用户绑定；校验通过（未绑定），将 federatedId 和 当前用户进行绑定；否则，返回错误信息

前端根据结果，提示信息

## 2. 后端接口设计

初始化 GET /users/federation/alipay/initAlipay

获取初始信息（如：nonce）

响应（application/json）：

```
{
  "acknowledged": true,
  "data": {
    "nonce": "aaa",
    "step": 1,
    "message": "初始化成功"
  }
}
```

发送验证码 POST /users/federation/alipay/sendCode

根据验证方式（checkType: mobile、emailAddress），将验证码发送到用户的原手机或原邮箱

请求体（application/json）：



```
{
  "nonce": "bbb",
  "checkType": "mobile"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "nonce": "ccc",
    "step": 1,
    "message": "发送成功"
  }
}
```

验证验证码 POST /users/federation/alipay/checkCode

提交验证码

请求体 (application/json) :

```
{
  "nonce": "ccc",
  "code": "1234"
}
```

响应 (application/json) :

```
{
  "acknowledged": true,
  "data": {
    "nonce": "xxx",
    "step": 2,
    "message": "验证成功",
    "": ""
  }
}
```

绑定 POST /users/federation/alipay/bindAlipay

- 提交 auth\_code 并绑定支付宝
- 请求体 (application/json) :

```
{ "nonce": "xxx", "auth_code": "acacac" }
```

- 响应 (application/json) :

```
{ "acknowledged": true, "data": { "message": "绑定成功" } }
```

#### 解除绑定支付宝

## 1. 功能流程设计

### STEP 1 安全验证

- 点击绑定支付宝
- 请求后端, /users/federation/alipay/initUnbindAlipay, 根据 当前用户 + 时间戳 + 步骤号 (1) + 密钥, 生成 nonce1
- 前端展示页面 (安全验证页面)
- 用户选择验证方式, 点击发送验证码 (控制点击频率 60s), 提交 (步骤号 (1), nonce1)
- 请求后端, /users/federation/alipay/sendCodeUnbindAlipay, 校验 nonce; 校验正确, 发送验证码到当前用户的手机或邮箱, 生成 nonce2; 否则, 返回错误信息
- 前端提示发送结果
- 用户接收到验证码后, 输入验证码, 提交 (验证码, 步骤号 (1), nonce2)
- 请求后端, /users/federation/alipay/checkCodeUnbindAlipay, 校验 nonce, 校验 验证码; 校验正确, 解除绑定; 否则, 返回错误信息

- 前端根据结果，提示错误 或 提示解绑成功

## 2. 后端接口设计

初始化 GET /users/federation/alipay/initUnbindAlipay

- 获取初始信息（如：nonce）
- 响应（application/json）：

```
{ "acknowledged": true, "data": { "nonce": "aaa", "step": 1, "message": "初始化成功" } }
```

发送验证码 POST /users/federation/alipay/sendCodeUnbindAlipay

- 根据验证方式（checkType: mobile、emailAddress），将验证码发送到用户的原手机或原邮箱
- 请求体（application/json）：

```
{ "nonce": "bbb", "checkType": "mobile" }
```

- 响应（application/json）：

```
{ "acknowledged": true, "data": { "nonce": "ccc", "step": 1, "message": "发送成功" } }
```

验证验证码 POST /users/federation/alipay/checkCodeUnbindAlipay

- 提交验证码
- 请求体（application/json）：

```
{ "nonce": "ccc", "code": "1234" }
```

- 响应 (application/json) :

```
{ "acknowledged": true, "data": { "message": "解绑成功" } }
```

#### #### 绑定微信

参考支付宝，仅微信认证、及获取微信用户信息的实现逻辑不同

#### #### 解除绑定微信

参考支付宝

#### #### 绑定QQ

参考支付宝，仅QQ认证、及获取QQ用户信息的实现逻辑不同

#### #### 解除绑定QQ

参考支付宝

#### ### 登录历史

##### #### 获取登录历史记录（分页）

获取当前用户的 登录历史，

## 1. 后端接口设计

获取安全设置 GET /users/log/authenticationLog

- 请求参数:

pageIndex

pageSize

- 响应 (application/json) :

```
```json
{
  "acknowledged": true,
  "data": {
    "pageIndex": 0,
    "pageSize": 20,
    "items": [
      {
        "username": "帐号",
        "ip": "IP地址",
        "userAgent": "客户端信息",
        "authnType": "登录方式 (1 电脑端, 2 移动端, 3 其他)",
        "authnTime": "登录时间",
        "authnResult": "登录结果 (1 成功, 2 失败)",
        "authnFailReason": "登录失败原因"
      },
      {.....}
    ]
  }
}
```

## 修改密码

见, 安全设置 - 修改密码