**API接口安全机制设计**

* API网关接口实现回顾
* 接口安全的业务需求
* 基于API网关实现安全机制

### **API网关接口实现回顾**

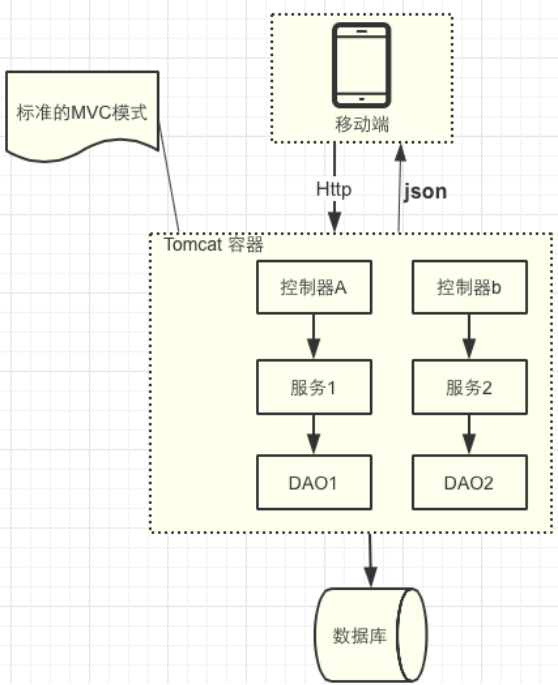
1. 什么是API网关？

API网关是一个轻量的java http 接口组件，可无缝将普通的 Serive 方法转换成 http 接口。并从已下几点来达到提高开发效率与接口质量的目的。

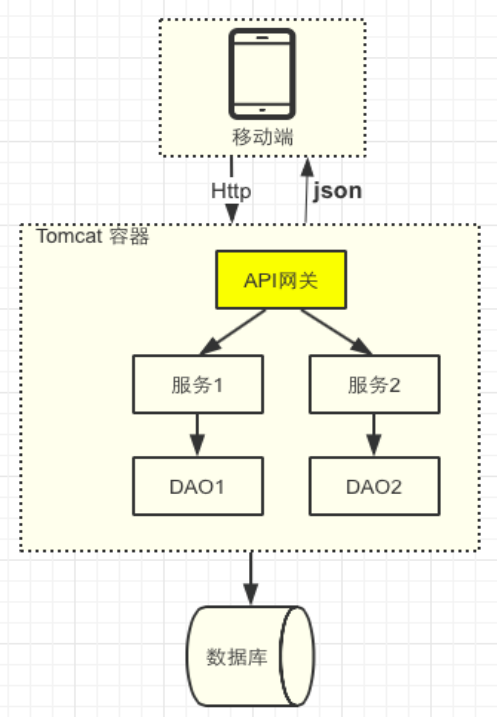
1. 去掉mvc控制器，将http请求直接无缝接入JAVA服务接口
2. 统一出入参格式
3. 统一异常规范
4. 自动检测服务接口规范

2、API网关 与普通Http接口实现流程对比

**普通Http 接口实现**

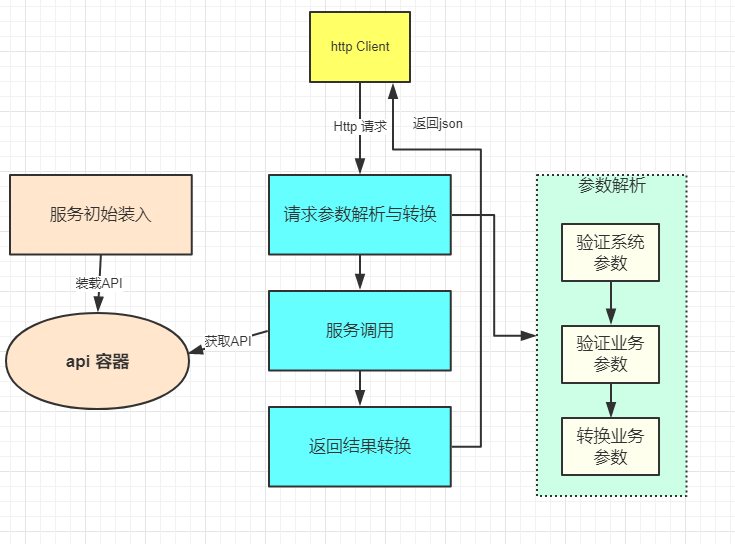


**API网关接口实现**



3、网关实现流程与技术

详细流程图



**请求参数说明：**

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| method | string | 方法名称 |
| paramter | json | 业务参数 |
| timestamp | long | 请求时间搓 |

**实现技术：**

1. java servlet
2. spring Ioc
3. Json 转换工具的使用

### **接口安全的业务需求**

**1、接口安全级别分组**

1. 白名单组
2. 黑名单组
3. 黑白名单组

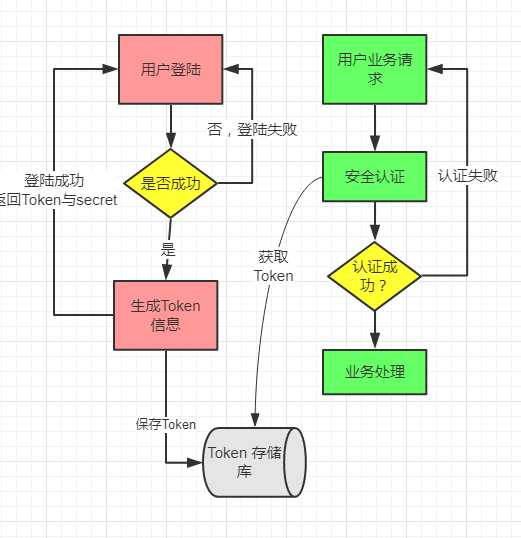
**2、基于Token安全机制认证需求**

1. 登录鉴权
2. 防止业务参数串改
3. 保护用户敏感信息
4. 防签名伪造

3、T**oken 认证机制整体架构**

整体架构分为Token生成与认证两部分：

1. Token生成指在登陆成功之后生成 Token 和密钥，并其与用户隐私信息、客户端信息一起存储至Token表,同时返回Token 与Secret 至客户端。
2. Token认证指客户端请求黑名单接口时，认证中心基于Token生成签名



**Token表结构说明:**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类别 | 说明 | 约束 |
| id | number | id主键 | 主键，自增长 |
| memberId | number | 会员ID |  |
| accessToken | varchar(50) | Token | 索引 |
| secret | varchar(50) | 密钥 |  |
| createdTime | datetime | 创建时间 |  |
| expiresTime | datetime | 有效期至 |  |
| clientIp | varchar(50) | 客户端IP |  |
| clientType | varchar(50) | 客户端类别 |  |
| eCode | varchar(50) | 设备标识 |  |
| uCode | varchar(50) | 设备用户标识 |  |

**业务请求具体参数：**

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| method | string | 方法名称 |
| param | json | 业务参数 |
| token | string | token值 |
| sign | string | 签名规则:md5(secret+method+param+token+secret+timestamp) |
| timestamp | long | 请求时间搓，允许与服务端10分钟误差 |

**签名规则：**

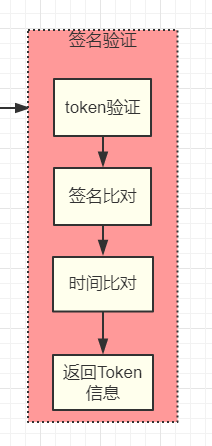
1.已指定顺序拼接字符串 secret+method+param+token+timestamp+secret

2.使用MD5进行加密，在转化成大写

**签名的目的：**

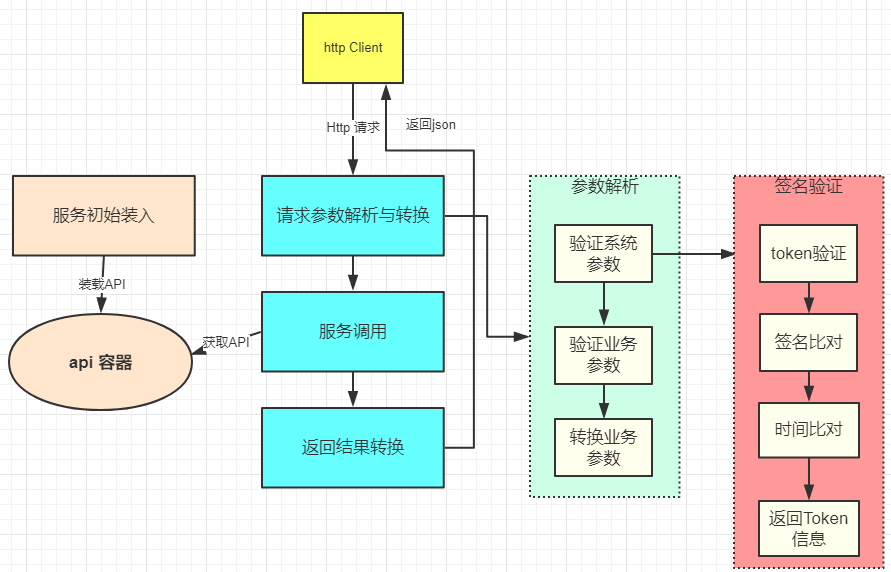
1. 防串改
2. 防伪造
3. 防重复使用签名

**服务端签名验证的具体流程：**



### **基于API网关实现安全机制**

**签名认证与API网关的整体认证流程**



**两个流程**

**Token生成**

登录成功后插，生成token与secret 保存至数据库。具体实现勿略

**Token 认证相关解决方案：**

1. 接口如何标识黑白名单？
2. 签名具体验证流程?
3. 用户ID等信息如何传递给业务实现接口？