

## 编程题

请同学们根据下面的业务描述和要求，使用第一代Spring Cloud核心组件完成项目构建、编码及测试。

### 业务描述

以注册、登录为主线，串联起**验证码生成及校验、邮件发送、IP防暴刷、用户统一认证**等功能。实现需基于Spring Cloud 微服务架构，技术涉及Nginx、Eureka、Feign（Ribbon、Hystrix）、Gateway、Config+Bus等。

### 注册

- 1) 用户访问到登录页面，在登录页面中有注册新账号功能
- 2) 点击“注册新账号”，跳转到注册页面
- 3) 在注册页面，需要用户输入邮箱地址、密码、确认密码，然后点击“获取验证码”，系统会生成验证码并向所输入的邮箱地址发送该验证码，用户拿到邮箱中的验证码输入后完成注册

规则如下

A：一分钟内只允许获取一次验证码（前端js控制即可），验证码为随机生成的6位数字，10分钟内有效，**验证码存储到mysql数据库中（因为大家未系统学习Redis，所以此处令牌存入数据库即可）**；

B：存储到mysql数据库之后，使用发邮件功能，将该验证码发送到所输入的邮箱地址中

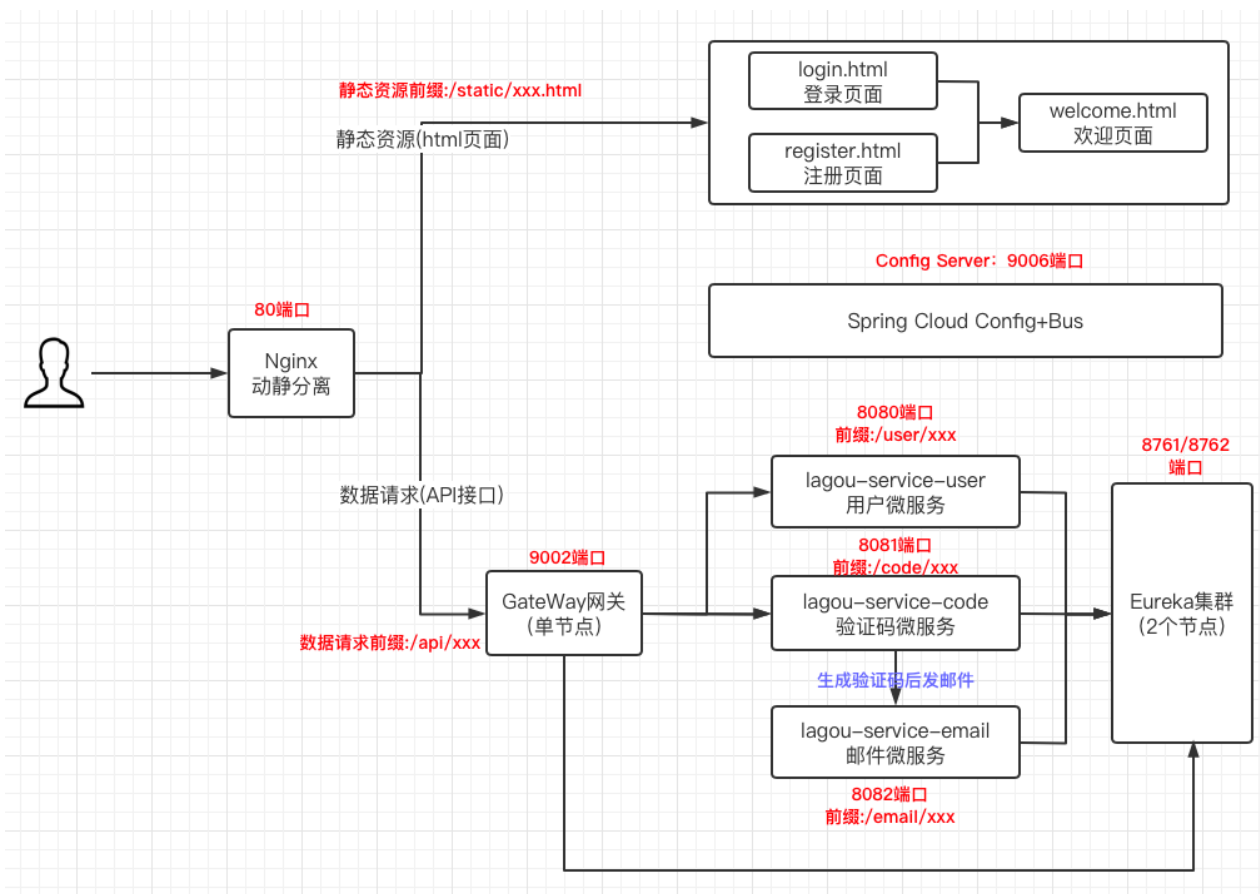
C：用户从邮箱中拿到验证码，点击注册时，需要进行校验，因为验证码已经存入数据库，此时只需要查询数据库中该邮箱地址对应的最近一次的验证码记录，校验验证码是否正确，是否超时，若有问题，准确提示给用户

4) 注册成功后，根据 <用户邮箱+密码> 生成签发token令牌（此处生成一个UUID模拟即可），**该token令牌存入数据库（因为大家未系统学习Redis，所以此处令牌存入数据库即可）**，并写入cookie中（以后的每次请求都会在cookie中携带该token，网关过滤器通过验证token的合法性来确定用户请求是否合法，如果token合法，根据token取出用户信息---->邮箱），最后重定向到欢迎页面（显示邮箱地址）

### 登录

- 1) 用户访问登录页面，在登录页面输入邮箱地址+密码
- 2) 点击登录，后台对用户名和密码进行验证，然后根据<用户邮箱+密码> 生成签发token令牌（此处生成一个UUID模拟即可），**该token令牌存入数据库（因为大家未系统学习Redis，所以此处令牌存入数据库即可）**，并写入cookie中（以后的每次请求都会在cookie中携带该token，网关过滤器通过验证token的合法性来确定用户请求是否合法，如果token合法，根据token取出用户信息---->邮箱），最后重定向到欢迎页面（显示邮箱地址）

### 架构描述



- Nginx

占用端口：80

实现动静分离。将静态资源 html 页面存放至本地磁盘，数据请求统一经过 GateWay 网关路由到下游微服务。

- 静态资源（html页面）

访问前缀：/static/xxx.html

包括登录页面 login.html、注册页面 register.html、以及成功登录之后的欢迎页面 welcome.html，各个页面细节元素后面有描述

- GateWay 网关

占用端口：9002端口 数据请求前缀:/api/xxx

完成统一路由、IP防暴刷（限制单个客户端IP在最近X分钟内请求注册接口不能超过Y次）、统一认证（登录时验证用户名密码是否合法，合法调用用户微服务生成token，写入cookie，并且携带邮箱地址重定向到欢迎页面；后续请求再到来时，验证客户端请求cookie中携带的token是否合法，合法则放行，此处不考虑token更新问题）等功能

路径路由规则：

/api/user/\*\* 路由到用户微服务

/api/code/\*\* 路由到验证码微服务

/api/email/\*\* 路由到邮件微服务

- lagou-service-user 用户微服务

占用端口：8080 数据请求前缀:/api/user/\*\*

提供注册接口、用户是否已注册接口、登录接口（生成token并入库，token写入cookie中）、查询用户登录邮箱接口等

- lagou-service-code 验证码微服务

占用端口：8081 数据请求前缀:/api/code/\*\*

用于提供验证码生成、验证码校验等接口，同时调用邮件微服务发送验证码

- lagou-service-email 邮件微服务

占用端口：8082 数据请求前缀:/api/email/\*\*

提供邮件发送功能，用于将生成的验证码发送到注册邮箱

- Spring Cloud Config+Bus

占用端口：9006

共享的配置：数据库连接信息、邮件发送相关配置、IP防暴刷指标参数（X分钟的X，Y上限的Y）

**注意：**除去Eureka是2个实例的集群模式，其他保持单实例

### 接口描述

涉及到的微服务名称定义、接口定义、参数名称，可以和下文提到的保持一致。

微服务名称	API 接口	返回值	接口描述
lagou-service-user	/user/register/{email}/{password}/{code}	true/false	注册接口，true成功，false失败
	/user/isRegistered/{email}	true/false	是否已注册，根据邮箱判断,true代表已经注册过，false代表尚未注册
	/user/login/{email}/{password}	邮箱地址	登录接口，验证用户名密码合法性，根据用户名和密码生成token，token存入数据库，并写入cookie中，登录成功返回邮箱地址，重定向到欢迎页
	/user/info/{token}	邮箱地址	根据token查询用户登录邮箱接口
lagou-service-code	/code/create/{email}	true/false	生成验证码并发送到对应邮箱，成功true，失败false
	/code/validate/{email}/{code}	0/1/2	校验验证码是否正确，0正确1错误2超时
lagou-service-email	/email/{email}/{code}	true/false	发送验证码到邮箱，true成功，false失败

### 页面描述

涉及到界面的，界面样式不做要求，界面元素完备即可。

- 登录界面 login.html



A login form diagram enclosed in a black rectangular border. It features two input fields: the top one is labeled '邮箱' (Email) and the bottom one is labeled '密码' (Password). Below these fields is a solid green button with the white text '登录' (Login). Underneath the button is the green text '注册新账号' (Register new account). The entire form is set against a light gray grid background.

登录界面包含输入项：

邮箱：email

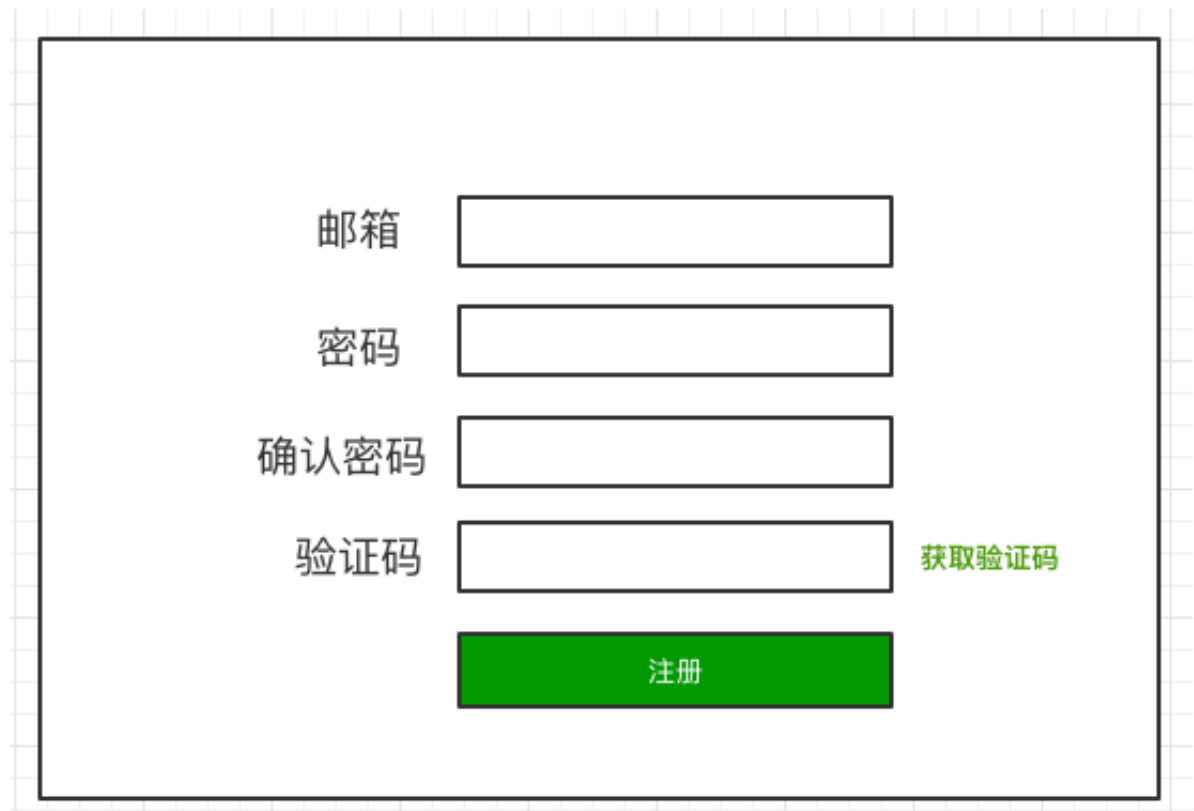
密码：password

点击“登录”调用登录接口，成功后重定向到欢迎页面welcome.html

ajax调用<http://localhost:9002/api/user/login/{email}/{password}>

点击“注册新账号”可跳转到注册页面register.html

- 注册界面 register.html



A registration form diagram enclosed in a black rectangular border. It contains four input fields stacked vertically, each with a label to its left: '邮箱' (Email), '密码' (Password), '确认密码' (Confirm Password), and '验证码' (Verification Code). To the right of the '验证码' field is the green text '获取验证码' (Get verification code). Below the input fields is a solid green button with the white text '注册' (Register). The form is set against a light gray grid background.

注册界面包含输入项：

邮箱：email

密码：password

确认密码：ConfirmPassword

验证码：code

点击“获取验证码”：调用获取验证码接口 <http://localhost:9002/code/create/{email}>

点击“注册”：调用注册接口 <http://localhost:9002/user/register/{email}/{password}/{code}>

- 欢迎页面 welcome.html

如下显示登录账户信息即可



## 关于网关和配置中心的说明

- GateWay网关
  - 统一路由，数据请求的统一出入口
  - 添加全局过滤器，进行IP注册接口的防暴刷控制，超过阈值直接返回错误码及错误信息（错误码：303，错误信息：您频繁进行注册，请求已被拒绝）
  - 添加全局过滤，进行token的验证，用户微服务和验证码微服务的请求不过滤（网关调用下游用户微服务的token验证接口）

- Config配置中心

将数据库配置、邮件发送相关配置、防暴刷参数配置，使用Config进行管理，对应微服务可以在自己的配置文件中直接使用\${xxx.yyy}的方式取出使用

## 关于数据表的说明

涉及到两个数据表：验证码存储表+令牌存储表，数据表参考如下

#### 验证码存储表

```
-- -----  
-- Table structure for lagou_auth_code  
-- -----  
  
DROP TABLE IF EXISTS `lagou_auth_code`;  
CREATE TABLE `lagou_auth_code` (  
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '自增主键',  
  `email` varchar(64) DEFAULT NULL COMMENT '邮箱地址',  
  `code` varchar(6) DEFAULT NULL COMMENT '验证码',  
  `createtime` datetime DEFAULT NULL COMMENT '创建时间',  
  `expiretime` datetime DEFAULT NULL COMMENT '过期时间',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
SET FOREIGN_KEY_CHECKS = 1;
```

#### 令牌存储表

```
-- -----  
-- Table structure for lagou_token  
-- -----  
  
DROP TABLE IF EXISTS `lagou_token`;  
CREATE TABLE `lagou_token` (  
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '自增主键',  
  `email` varchar(64) NOT NULL COMMENT '邮箱地址',  
  `token` varchar(255) NOT NULL COMMENT '令牌',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
SET FOREIGN_KEY_CHECKS = 1;
```

#### 关于测试

- 注册测试
- 登录测试
- 未登录状态下，清空cookie，直接访问后台的邮件服务，<http://www.test.com/api/email/{email}/{code}>，验证无token情况下是否被网关拦截

#### 关于跨域

前文在 Html 静态页面中有ajax 请求数据API统一接口9002的地方，会涉及跨域问题，可以考虑将静态资源和数据请求接口放在同一个域名下，根据url前缀在nginx层进行区分。

比如所有部署，包括nginx都在一台机器上

- 可以给机器设置一个域名 [www.test.com](http://www.test.com)

- 静态资源访问 [www.test.com/static/xxx.html](http://www.test.com/static/xxx.html)
  - 数据API接口请求 [www.test.com/api/xxx/yyy](http://www.test.com/api/xxx/yyy)
- 通过/static和/api在nginx层进行区分