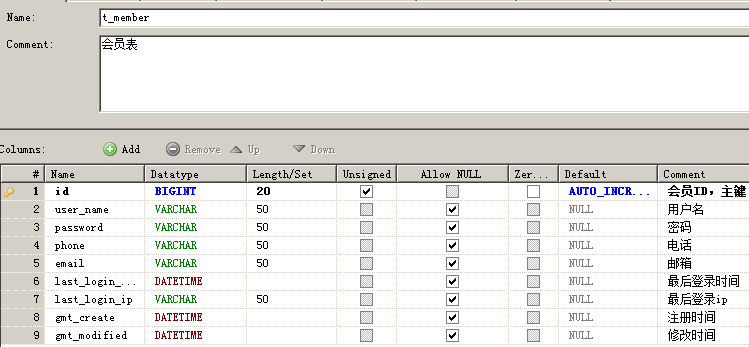
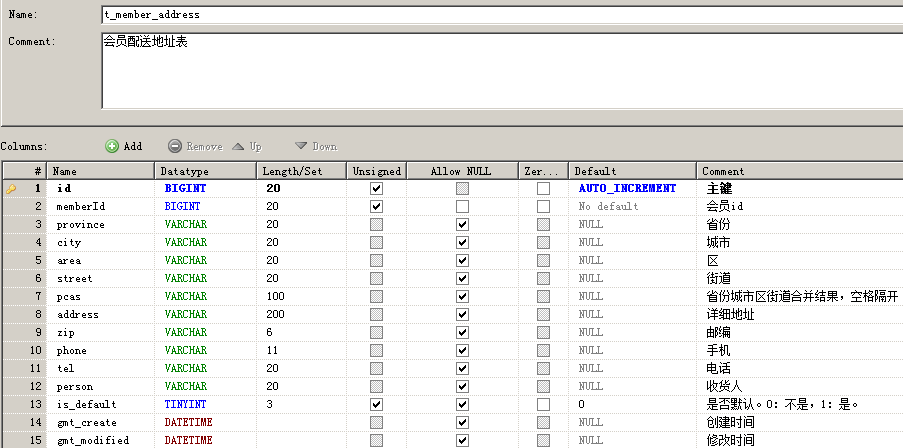
# 今日任务

1. 会员系统
2. 系统会员登录问题
3. SSO

# 会员系统





# 登录分析

在我们的实际开发中，更多的是采用集群系统。那么问题来了，对于集群系统的登录问题，我们如何解决呢？

负载均衡

登录

服务器

session

session

session

你会用tomcat的一个session拷贝方式来做同步。但只是局域网。但是使用不多，现在。

2）多系统如何登录？？

    如果说我们在每一个系统中都要进行一次登录，那么用户体验度也就差的没法用了。以京东商城为例，如果用户在登录京东商城的时候需要登录一次，在查询商品的时候还有在登录一次，加入购物车是还要重新登录，...(注意，每跳过一个页面都是进入了一个新的系统，请看他的url 地址的变化) 那么京东商城的成绩 也就不会那么优异了。

    如果说用session 进行解决的话，可是session的作用域只限于当前系统，如果打开其他系统，当前session信息是不存在于其他系统的。

同一系统或者同一系统的不同子系统使用，多次登录？？？体验差，效率低！！！

商城系统

聊天系统

一次登录

二次登录

无法同步

**如何做到一次登录，一个系统或者多个子系统免登录？**

# Sso

SSO英文全称Single Sign On，**单点登录**。SSO是在多个应用系统中，用户只需要登录一次就可以访问所有相互信任的应用系统。它包括可以将这次主要的登录映射到其他应用中用于同一个用户的登录的机制。它是目前比较流行的**企业业务整合**的解决方案之一。

实现方式：

1. 基于cookie
2. *基于CAS*
3. 基于TOKEN
4. 其他

# 项目实战

## Sso架构

Sso服务端

mall

token

没有token

没有

Token

 返回

登录页面

认证成功后，将服务端token返回给客户端

有就验证

tokencache

## 接口

**注册数据校验**

URL：/member/check

参数：主要是数据唯一性校验，有三个数据，分别是，userName, tel, email，为了区分校验，添加额外辅助参数 type：1 ，2， 3 分别代表上述三个字段。

返回值：ResponseMessage对象

**注册**

URL：/member/register

参数：member对象。

返回值：ResponseMessage对象

**登录**

URL：/member/login

参数：用户名和密码

返回值：如果有redirectUrl，就返回该URL，否则返回商城首页

# Sso

## Pom

|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>com.go9</groupId>  <artifactId>go9</artifactId>  <version>1.0.0-SNAPSHOT</version>  </parent>  <artifactId>go9-sso-service</artifactId>  <dependencies>  <dependency>  <groupId>com.go9</groupId>  <artifactId>go9-common</artifactId>  <version>1.0.0-SNAPSHOT</version>  </dependency>  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>  </dependency>  <dependency>  <groupId>org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid-spring-boot-starter</artifactId>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <scope>runtime</scope>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-thymeleaf</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-test</artifactId>  <scope>test</scope>  </dependency>  </dependencies>  </project> |

## Yml

|  |
| --- |
| eureka:  client:  serviceUrl:  defaultZone: http://localhost:666/eureka/  server:  port: 3030  spring:  application:  name: go9-service-sso  thymeleaf:  cache: **false**  datasource:  type: com.alibaba.druid.pool.DruidDataSource  url: jdbc:mysql://localhost:3306/go9?characterEncoding=utf-8&autoReconnect=true&failOverReadOnly=false  username: root  password: 111111  driver-class-name: com.mysql.cj.jdbc.Driver  logging:  level:  root: info  #指定某个具体包下的类输出等级  com.netflix.discovery.DiscoveryClient: error |

## MemberMapper



|  |
| --- |
| <select id=*"selectByUnique"* parameterType=*"com.go9.sso.pojo.Member"* resultMap=*"BaseResultMap"*>  select  <include refid=*"Base\_Column\_List"* />  from t\_member  <where>  <if test=*"userName != null"*>  user\_name = #{userName,jdbcType=VARCHAR}  </if>  <if test=*"phone != null"*>  or phone = #{phone,jdbcType=VARCHAR}  </if>  <if test=*"email != null"*>  or email = #{email,jdbcType=VARCHAR}  </if>  </where>  </select>  <select id=*"selectByLogin"* parameterType=*"com.go9.sso.pojo.Member"* resultMap=*"BaseResultMap"*>  select  <include refid=*"Base\_Column\_List"* />  from t\_member  <where>  <if test=*"userName != null"*>  user\_name = #{userName,jdbcType=VARCHAR}  </if>  <if test=*"phone != null"*>  and phone = #{phone,jdbcType=VARCHAR}  </if>  <if test=*"email != null"*>  and email = #{email,jdbcType=VARCHAR}  </if>  and password = #{password,jdbcType=VARCHAR}  </where>  </select> |

## MemberService

|  |
| --- |
| **package** com.go9.sso.token;  **import** java.util.HashMap;  **import** java.util.Map;  **public** **class** TokenCache {  **private** Map<String,Object> tokens;    **public** TokenCache() {  **this**.tokens = **new** HashMap<>();  }  **public** Map<String, Object> getTokens() {  **return** tokens;  }    } |

|  |
| --- |
| package com.go9.sso.service;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.go9.common.util.UUIDUtils;  import com.go9.sso.mapper.MemberMapper;  import com.go9.sso.pojo.Member;  import com.go9.sso.token.TokenCache;  @Service  public class MemberServiceImpl implements MemberService {  @Autowired  private MemberMapper memberMapper;  @Autowired  private TokenCache tokenCache;    @Override  public String login(Member member) {  Member mem = memberMapper.selectByLogin(member);  if(mem!=null) {  //generate token  String uuid = UUIDUtils.getUUID();  tokenCache.getTokens().put(uuid, mem);  return uuid;  }else {  return null;  }  }  @Override  public void logout(String token) {  tokenCache.getTokens().remove(token);  }  @Override  public Member getMemberInfo(String token) {  Object object = tokenCache.getTokens().get(token);  if(object!=null)  return (Member)object;  else {  return null;  }  }  @Override  public void register(Member member) {  memberMapper.insertSelective(member);  }  @Override  public Member validateUnique(Member member) {  return memberMapper.selectByUnique(member);  }  } |

## MemberController

|  |
| --- |
| package com.go9.sso.controller;  import java.util.HashMap;  import java.util.Map;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Controller;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.ResponseBody;  import com.go9.common.ServiceMessage;  import com.go9.sso.pojo.Member;  import com.go9.sso.service.MemberService;  @Controller  public class MemberController {  @Autowired  private MemberService memberService;  @PostMapping("/register")  public @ResponseBody Object register(Member member){  try {  memberService.register(member);  return ServiceMessage.ok();  } catch (Exception e) {  e.printStackTrace();  return ServiceMessage.error();  }  }    @PostMapping("/login")  public @ResponseBody Object login(Member member){  try {  String token = memberService.login(member);  if(token == null) {  return ServiceMessage.info("登录名或者密码错误");  }  Map<String,Object> data = new HashMap<>();  data.put("token", token);  return ServiceMessage.ok(data);  } catch (Exception e) {  e.printStackTrace();  return ServiceMessage.error();  }  }    @GetMapping("/info")  public @ResponseBody Object getUserInfo(String token){  try {  Member member = memberService.getMemberInfo(token);  return ServiceMessage.ok(member);  } catch (Exception e) {  e.printStackTrace();  return ServiceMessage.error();  }  }    @GetMapping("/check")  public @ResponseBody Object checkUnique(Member member){  try {  Member mem = memberService.validateUnique(member);  return ServiceMessage.ok(mem);  } catch (Exception e) {  e.printStackTrace();  return ServiceMessage.error();  }  }    @PostMapping("/logout")  public @ResponseBody Object logout(String token){  try {  memberService.logout(token);  return ServiceMessage.ok();  } catch (Exception e) {  e.printStackTrace();  return ServiceMessage.error();  }  }  } |

## MemberAddressMapper

|  |
| --- |
| <select id=*"selectDefaultAddress"* resultMap=*"BaseResultMap"*>  select  <include refid=*"Base\_Column\_List"* />  from t\_member\_address  where is\_default = 1  </select> |

## MemberAddressService

|  |
| --- |
| package com.go9.sso.service;  import java.util.Date;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.go9.sso.mapper.MemberAddressMapper;  import com.go9.sso.pojo.MemberAddress;  @Service  public class MemberAddressServiceImpl implements MemberAddressService {  @Autowired  private MemberAddressMapper memberAddressMapper;    @Override  public List<MemberAddress> getAddresses(long memberId) {  return memberAddressMapper.selectByMemberId(memberId);  }  @Override  public void add(MemberAddress address) {  if(address.getYesDefault()) {  MemberAddress address2 = memberAddressMapper.selectDefaultAddress();  if(address2!=null) {  address2.setYesDefault(false);  memberAddressMapper.updateByPrimaryKey(address2);  }  }  Date now = new Date();  address.setGmtCreate(now);  address.setGmtModified(now);  memberAddressMapper.insertSelective(address);  }  } |

## MemberAddressController

|  |
| --- |
| package com.go9.sso.controller;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Controller;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.ResponseBody;  import com.go9.common.ServiceMessage;  import com.go9.sso.pojo.MemberAddress;  import com.go9.sso.service.MemberAddressService;  @Controller  public class MemberAddressController {  @Autowired  private MemberAddressService memberAddressService;    @GetMapping("/address")  public @ResponseBody Object getAddresses(long memberId){  try {  List<MemberAddress> addresses = memberAddressService.getAddresses(memberId);  return ServiceMessage.ok(addresses);  } catch (Exception e) {  e.printStackTrace();  return ServiceMessage.error();  }  }    @PostMapping("/address/add")  public @ResponseBody Object add(MemberAddress memberAddress){  try {  memberAddressService.add(memberAddress);  return ServiceMessage.ok(memberAddress);  } catch (Exception e) {  e.printStackTrace();  return ServiceMessage.error();  }  }  } |