Core类型项目对接ID操作文档

环境：Core项目

目的：对接ID项目登录，使在一个多系统共存的环境下，用户在一处登录后，就不用在其他系统中登录

**前提：**

应用对接ID项目登录，需ID登录的账户已对该应用授权

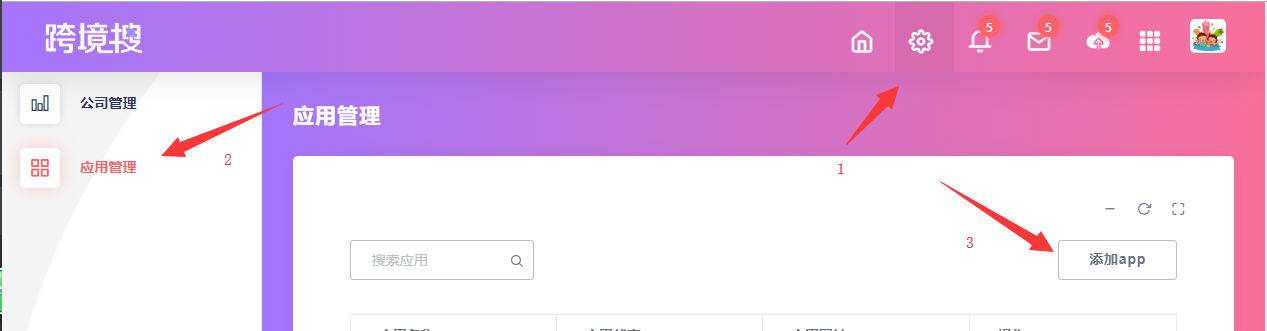
ID项目授权应用操作：

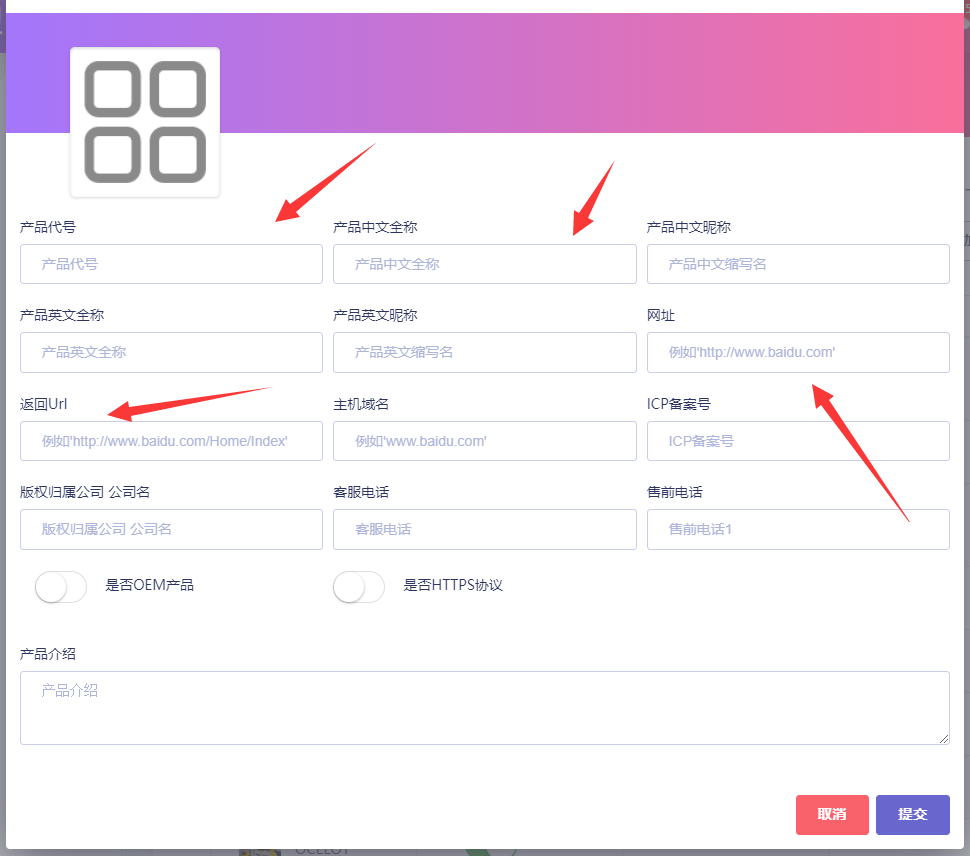
1. 登录ID项目

线上地址：<http://id.lsmaps.com>

本地开发测试：http://192.168.1.179:8087/

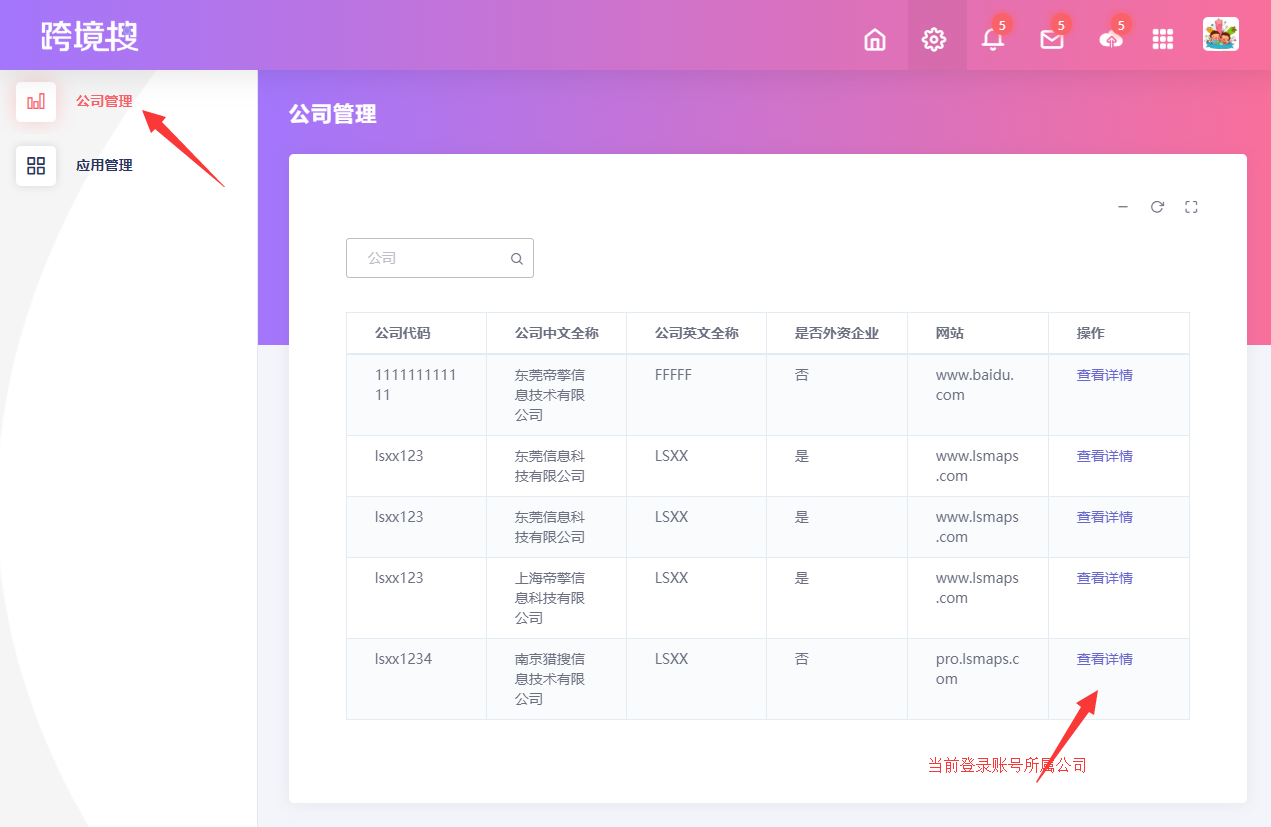
1. 添加应用

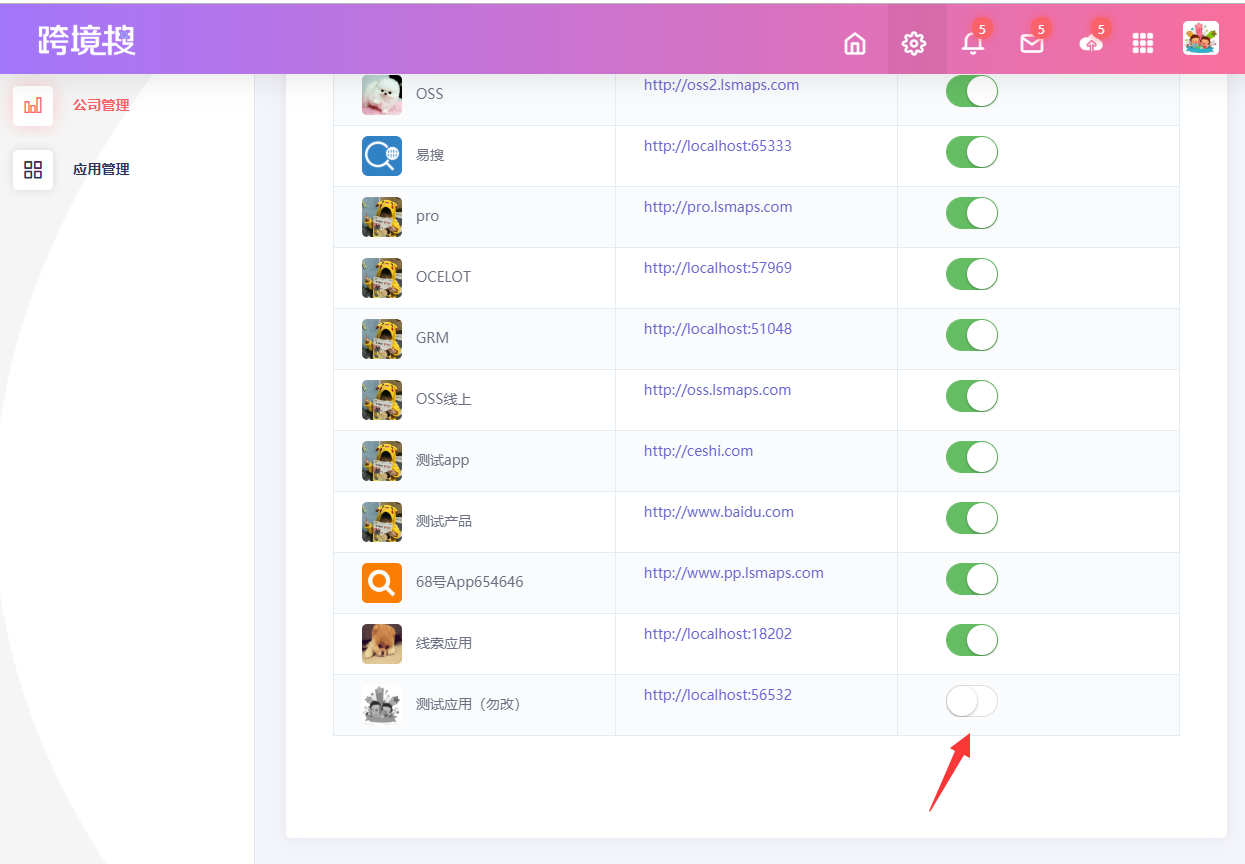




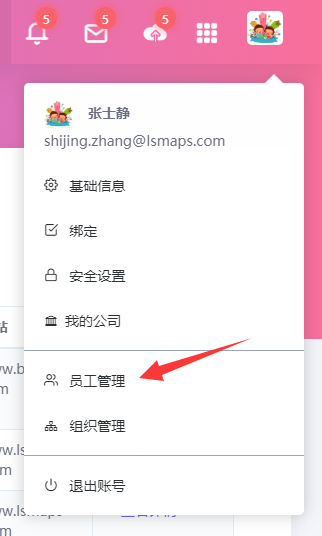
Ps: 1、应用ReturnUrl为后面需要在该应用配置的验证ticket方法路由地址

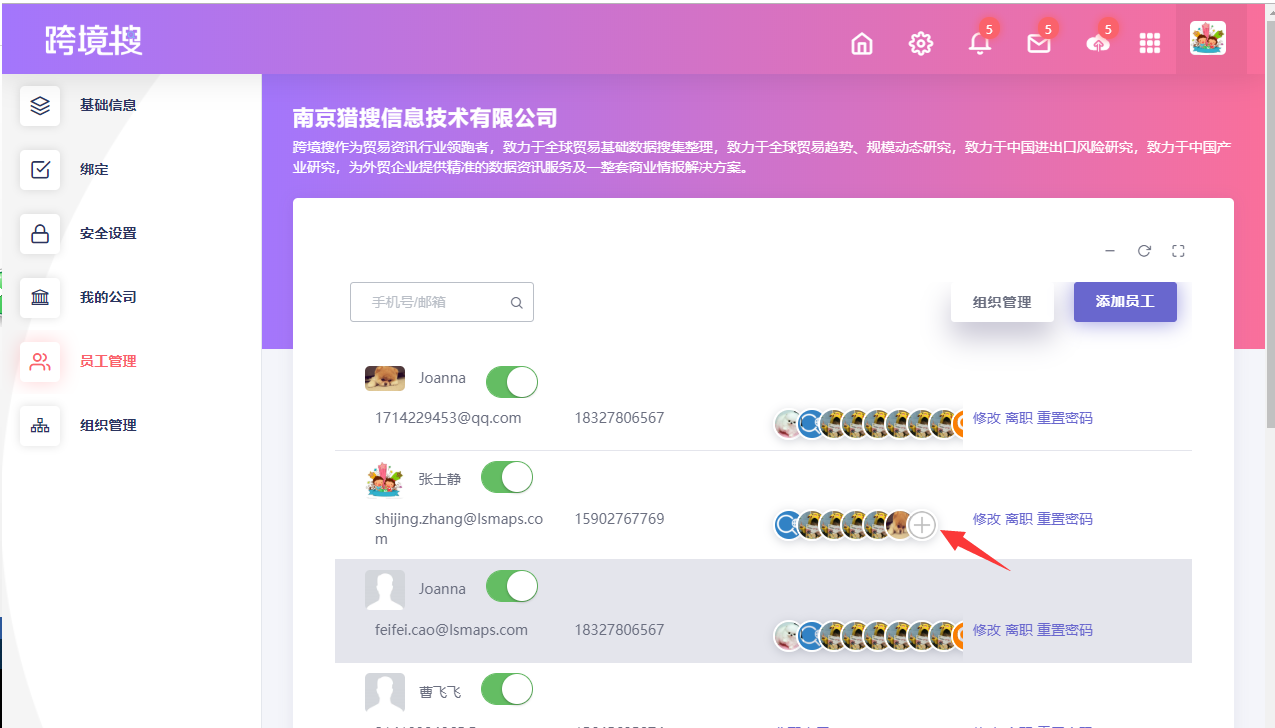
1. 本地测试http://localhost:000网址添加会提示格式不正确，可以https://www.baidu.com/为网址或ReturnUrl添加,添加成功后在数据库更改正确地址
2. 为当前登录账户所属公司授权该应用





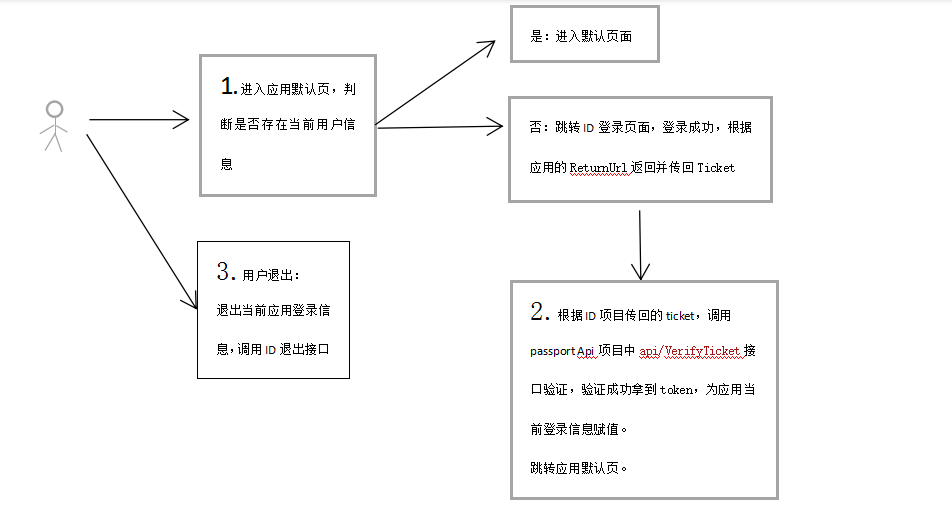
1. 为当前登录账户授权该应用





**应用配置：**

流程图：



1. 登录信息存储配置（采用Cookie）

Startup文件配置（配置方法内，注入或使用顺序不可变）

public void ConfigureServices(IServiceCollection services)

{

services.Configure<CookiePolicyOptions>(options =>

{

options.CheckConsentNeeded = context => true;

options.MinimumSameSitePolicy = SameSiteMode.None;

});

services.AddMvc(option =>

{

//添加Token验证

option.Filters.Add(typeof(AuthenticationAttribute));

}).SetCompatibilityVersion(CompatibilityVersion.Version\_2\_2);

//添加Cookie注入

services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)

.AddCookie(options =>

{

options.CookieName = "\_oceloctTestCookie";

options.LoginPath = "/Home/Index";

options.LogoutPath = "/Home/ExitAsync";

options.ExpireTimeSpan = new TimeSpan(2, 0, 0);

});

#region 接口注入

services.AddSingleton<IHttpContextAccessor, HttpContextAccessor>();

#endregion

}

public void Configure(IApplicationBuilder app, IHostingEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseCookiePolicy();

//使用身份验证

app.UseAuthentication();

app.UseMvc(routes =>

{

routes.MapRoute(

name: "default",

template: "{controller=Home}/{action=Index}/{id?}");

});

//使用Cookie

var cookiePolicyOptions = new CookiePolicyOptions

{

MinimumSameSitePolicy = SameSiteMode.Strict,

};

app.UseCookiePolicy(cookiePolicyOptions);

}

1. 判断登录信息跳转ID登录页

///跳转链接<http://id.lsmaps.com/Login?ReturnUrl=http://ocelot.lsmaps.com:80>

/// <summary>

/// 判断当前登陆

/// </summary>

/// <param name="ReturnUrl"></param>

/// <returns></returns>

public IActionResult Index(string ReturnUrl = "")

{

var user = \_accessor.HttpContext.User;

if (user == null || !user.Identity.IsAuthenticated)

{

var loginUrl = idUrl + $"Login?ReturnUrl=http://{\_accessor.HttpContext.Request.Host.Host}:{HttpContext.Request.Host.Port}{ReturnUrl}";

Response.Redirect(loginUrl);

}

return View();

}

1. 验证ticket方法

/// <summary>

/// 验证登录

/// </summary>

/// <param name="ticket"></param>

/// <param name="returnUrl"></param>

public IActionResult VerifyLoginAsync(string ticket, string returnUrl)

{

try

{

if (string.IsNullOrEmpty(ticket) && string.IsNullOrEmpty(returnUrl))

{

return Ok("验证Ticket失败");

}

//验证Ticket

var verifyTicketUrl = string.Format(passportApiUrl + "api/VerifyTicket?ticket={0}", ticket);

var web = new WebClient { Encoding = Encoding.UTF8 };

var data = web.DownloadString(verifyTicketUrl);

var result = JsonConvert.DeserializeObject<ResParameter>(data);

if (result.code != "200")

{

return Ok(result);

}

//设置当前登录状态为 true

var claims = new List<Claim>

{

new Claim(ClaimTypes.Name, result.data.ToString()),

new Claim("FullName", "Joanna.Zhang")

};

var claimsIdentity = new ClaimsIdentity(claims, CookieAuthenticationDefaults.AuthenticationScheme);

var authProperties = new AuthenticationProperties

{

AllowRefresh = true,

ExpiresUtc = DateTimeOffset.UtcNow.AddHours(2),

IsPersistent = true,

IssuedUtc = DateTimeOffset.Now,

RedirectUri = "../Home/Index",

};

Task.Run(async () =>

{

await HttpContext.SignInAsync(

CookieAuthenticationDefaults.AuthenticationScheme,

new ClaimsPrincipal(claimsIdentity), authProperties);

}).Wait();

return RedirectToAction("Index", "Home");

}

catch (Exception e)

{

return Ok(e.Message.ToString());

}

}

PS：这个方法地址就是在ID项目添加应用时填写的返回Url

1. 退出

///退出地址链接[http://id.lsmaps.com/Login/Exit?ReturnUrl=http://ocelot.lsmaps.com:80](http://id.lsmaps.com/Login?ReturnUrl=http://ocelot.lsmaps.com:80)

/// <summary>

/// 退出

/// </summary>

/// <returns></returns>

public async Task ExitAsync()

{

await HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);

var exitUrl = idUrl + $"Login/Exit?ReturnUrl=http://{\_accessor.HttpContext.Request.Host.Host}:{HttpContext.Request.Host.Port}";

Response.Redirect(exitUrl);

}

1. Token验证

每次调用方法前都需判断当前token是否有效

添加AuthenticationAttribute.cs文件存放于Controllers文件夹内，注入代码已在startup文件配置

1. 权限验证

需权限验证控制器或方法添加

[Authorize]

1. 提示页面

触发ID项目退出方法

<iframe hidden="hidden" src="http://id.lsmaps.com/Exit/Logout"></iframe>