

03_开发用户注册_登录_在线状态机制【AskyEdu.com】

《Asky极简教程：零基础1小时学编程》开源教程

从零开始全程演示，如何开发一个大型互联网系统，开源教程 + 开源代码 + 开源解决方案

本教程从零基础入门，1小时学完前3节后，可学会基本编程思路，后面的教程是在这基础上的深入进阶，每个小节学习时间30分钟左右，由浅入深，循序渐进，从完全不懂编程到逐渐掌握编程技能

AskyEdu.com 首届dnc开源峰会 dncNew.com 开源社区QQ群 618093978

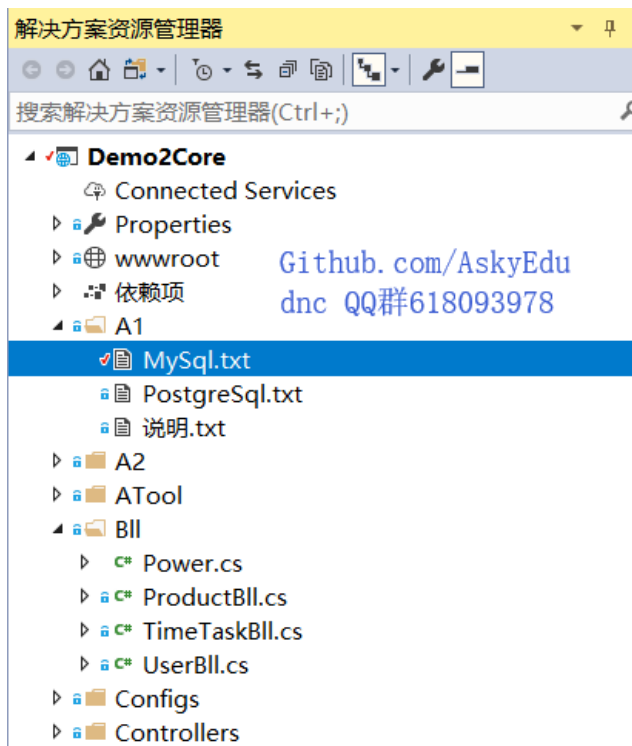
- 01、安装开发工具、开源数据库
- 02、零基础30分钟入门demo示例dnc + vue
- 03、开发用户注册、登录、在线状态模块
- 04、本机vbox虚拟机安装Linux CentOS系统
- 05、XShell连接Linux基本操作
- 06、Linux离线安装dnc运行环境
- 07、VS发布生成dnc部署包
- 08、部署dnc到Linux + 守护进程systemd
- 09、nginx负载均衡 + 多台Linux Web服务器
- 10、Linux搭建Redis Cluster集群
- 11、dnc + Redis 零基础30分钟上手
- 12、C#编程语言基本语法
- 13、Linux基本操作命令
- 14、vim基本操作
- 15、Asky开源工具类库Nuget
- 16、dnc + 开源数据库Tidb
- 17、dnc + 开源数据库PostgreSQL
- 18、dnc + 极简分片分库 + MySql/PostgreSQL
- 19、dnc + RabbitMQ 消息队列
- 20、dnc + kafka 消息队列
- 21、dnc + ElasticSearch 搜索引擎
- 22、dnc + Docker 容器
- 23、微服务架构 dnc on Linux/Docker

03、开发用户注册、登录、在线状态模块

1、请先学习《极速入门dnc+vue前后端分离》这一节教程，在它的基础上，下载本节示例代码

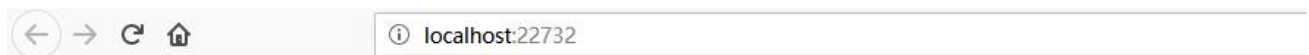
https://github.com/AskyEdu/Asky/raw/master/02_demoCode/02_Demo2Core.zip

解压后复制替换到之前的项目中：



2、复制A1/MySQL.txt中的创建user表的sql语句，在HeidiSql工具中，点击【查询】，复制到窗口里面，按F9执行，创建user表。关于HeidiSql工具的详细教程请看第1节、第2节课程

3、Ctrl+F5运行，自动在浏览器中打开



.NET Core 2.x demo1值

【0】A.GetRootPath()当前项目：D:\A7\VstsGit108\AskylDemo2Core\

【1】a.config配置节点Config.LogPath：D:\Log\

【2】读写cookie值：ck1cookieValue

【2】读写cookie值，中文编码解码：ck2cookieValue中文

【3】读写本地缓存值：k1缓存值

【4】api接口/home/demo1?id=1&name=测试

【5】dnc+vue示例/vue3.htm

【6】数据库缓存表配置节点Config.RootUrl：http://askyedu.com/

【7】dnc+vue示例，新用户注册/userAdd.htm

【8】dnc+vue示例，用户登录/userLogin.htm

1) 复制/userAdd.htm拼接到网址后面，输入邮箱、密码，注册一个新用户



邮箱

密码

确认密码

2) 注册成功后会自动登录，并跳转到目标页/home/demo1，可看到已登录用户的信息



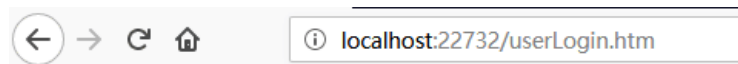
【已登录用户】

```
{ "Uid": 2, "Email": "a2@a.com", "Mobile": null, "NickName": null, "LoginToken": "8b19432a-676c-4737-b011-7f9e11391798" }
```

 【退出Url】 /user/logout?returnUrl=/home/demo1

[Github.com/AskYedu](https://github.com/AskYedu)
dnc QQ群618093978

3) 复制/userLogin.htm拼接到网址后面，输入邮箱、密码，测试手动登录功能



邮箱 请输入邮箱

密码 请输入密码

登录

[Github.com/AskYedu](https://github.com/AskYedu)
dnc QQ群618093978

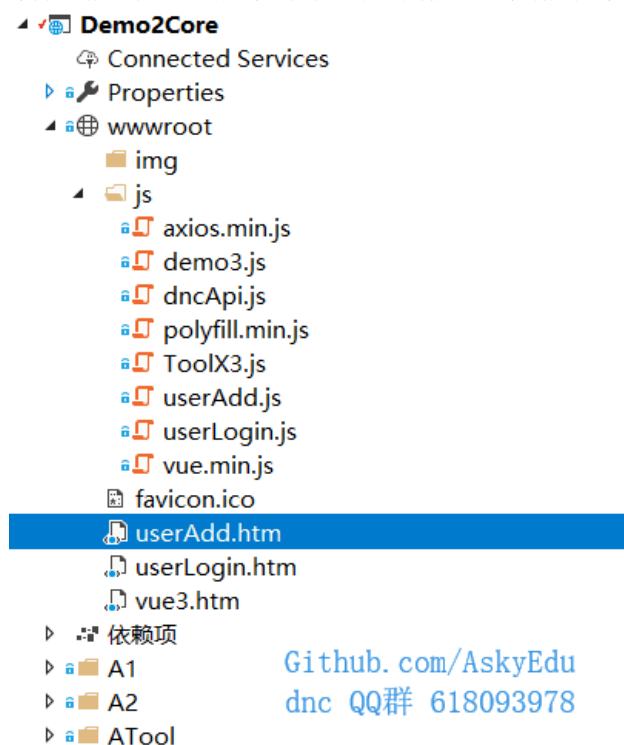
4) 复制 /user/logout?returnUrl=/home/demo1拼接到网址后面，测试用户退出登录状态，后面的returnUrl参数表示退出后跳转到指定的目标页/home/demo1



未登陆，currentUser为null，登录页/userLogin.htm

4、上面已经演示了用户注册、登录、退出、在线状态机制，现在开始分析实现这些功能的代码

1) wwwroot/userAdd.htm 新用户注册页，数据交互流程：从页面输入框邮箱、密码 -> vue调用dnc后台接口 -> 查询数据库邮箱是否已注册，如果未注册，则插入一条新用户数据



```
<div id="a1">
  <!-- dnc + vue 新用户注册 -->
  <div>
    邮箱 <input type="text" v-model="email" @blur="blurEmail()" style="width:80px" />
    <span v-if="isErrorEmail" style="color:red;">{{tipErrorEmail}}</span>
  </div>
  <div>
    密码 <input type="password" v-model="pwd" @blur="blurPwd()" style="width:80px" />
    <span v-if="isErrorPwd" style="color:red;">{{tipErrorPwd}}</span>
  </div>
  <div>
    确认密码 <input type="password" v-model="pwd2" @blur="blurPwd2()" style="width:80px" />
    <span v-if="isErrorPwd2" style="color:red;">{{tipErrorPwd2}}</span>
  </div>
  <div><button @click="addUserByEmail">注册</button></div>
```

wwwroot/userAdd.js 这是vue接收页面输入框的值

```
methods: {
  //新用户注册
  addUserByEmail: function () {
    var vm = this; //必须这样，后面的字段赋值才能在页面显示{{result}}
    vm.blurEmail(); vm.blurPwd(); vm.blurPwd2(); //验证输入框
    if (vm.isErrorEmail == true || vm.isErrorPwd == true || vm.isErrorPwd2 == true)
      return;
    apiAddUserByEmail(vm.email, vm.pwd, '/home/demo1');//vue调用dnc后台接口
  },
  //验证邮箱输入框
  blurEmail: function () {
    var vm = this; vm.isErrorEmail = true;
    if (vm.email == "") { vm.tipErrorEmail = "请输入邮箱"; }
    else if (IsEmail(vm.email) == false) { vm.tipErrorEmail = "邮箱格式不正确"; }
    else { vm.isErrorEmail = false; }
  },
  //验证密码输入框
  blurPwd: function () {
    var vm = this; vm.isErrorPwd = true;
    if (vm.pwd == "") { vm.tipErrorPwd = "请输入密码"; }
    else if (vm.pwd.length < 6) { vm.tipErrorPwd = "密码必须6个字符以上"; }
    else { vm.isErrorPwd = false; }
  },
}
```

wwwroot/dncApi.js 这是vue调用dnc后台接口代码

//vue调用dnc后台接口, 新用户注册, 方法名加api前缀

```
function apiAddUserByEmail(email, pwd, returnUrl) {  
  if (email == "" || pwd == "") { alert('email、密码不能为空'); return; }  
  var params = new URLSearchParams();  
  params.append('email', email);  
  params.append('pwd', pwd);  
  axios.post('/User/AddUserByEmail', params) //axios是一个js工具类库, 用于调用后台接口  
    .then(function (response) {  
      var result = response.data; //返回值  
      if (result != null) {  
        if (result.status == 1) //alert("注册成功");  
          apiLoginByEmail(email, pwd, returnUrl); //自动登录并跳转到目标页  
        else  
          alert(result.msg);  
      }  
    }).catch(function (error) { console.log("【操作失败】" + error); });  
}
```

dnc后台接口代码 /Controllers/UserController.cs 鼠标点击方法名 UserBll.AddUserByEmail 按F12进入查看代码

```
public class UserController : Controller
```

```
{  
  //新用户注册  
  // /user/AddUserByEmail?email=a1@a.com&pwd=123  
  0 个引用 | mike, 7 小时前 | 1 名作者, 1 项更改 | 0 个请求 | 0 异常  
  public async Task<string> AddUserByEmail(string email, string pwd)  
  {  
    return await UserBll.AddUserByEmail(email, pwd);  
  }  
}
```

按F12进入查看代码

dnc后台Bll业务层代码, 执行数据库操作

```

public static async Task<string> AddUserByEmail(string email, string pwd)
{
    if (email.IsNullOrEmpty()) return Code.FailJson("email不能为空");
    if (pwd.IsNullOrEmpty()) return Code.FailJson("密码不能为空");
    email = email.Trim().ToLower(); //转小写
    pwd = pwd.Trim(); //去掉首尾空格

    try
    {
        using (var conn = Db.Conn())
        {
            var sql = "select uid from user where email=@email" + Tool.SqlPage(); //取出第一条数据
            var uid = (await conn.QueryAsync<int>(sql, new { email = email })).FirstOrDefault();
            if (uid > 0)
                return Code.FailJson(email + "邮箱已注册，请直接登录");

            var pwdHash = GetPwdHash(pwd); //密码+guid混淆后md5加密
            sql = "insert into user (email, pwd) values (@email,@pwd)"; //插入数据库一条新用户数据
            var amount = await conn.ExecuteAsync(sql, new { email = email, pwd = pwdHash });
            return (amount > 0) ? Code.SuccessJson(amount) : Code.FailJson("注册用户失败");
        }
    }
    catch (Exception ex) { LogA.AppError("AddUserByEmail" + ex.ToString()); return Code.FailJson(ex.Message); }
}

```

5、用户登录、退出模块，代码类似，不再截图，请下载示例代码查看

1) 用户登录

wwwroot/userLogin.htm

wwwroot/userLogin.js

wwwroot/dncApi.js

Controllers/UserController.cs 方法名 LoginByEmail(string email, string pwd)

Bll/UserBll.cs 方法名 LoginByEmail(string email, string pwd)

2) 用户退出登录状态的链接 /user/logout?returnUrl=/home/demo1

Controllers/UserController.cs 方法名 Logout(string returnUrl)

Bll/UserBll.cs 方法名 Logout(string cookieValue)

6、当前登录用户CurrentUser用 cookie + 缓存 实现在线状态机制

1) 用户输入邮箱、密码 -> 取出这个邮箱的对应用户 -> 验证密码 -> 如果密码正确，则更新登录时间、登录次数 -> 设置当前登录用户的Cookie、当前登录缓存

Bll/UserBll.cs 方法名 LoginByEmail(string email, string pwd)


```

using (var conn = Db.Conn())
{
    //Tool.SqlPage() 取出第一条数据
    var sql = "select pwd,uid,email,mobile,nickname,logintoken from user where email=@email" + Tool.SqlPage();
    var vm = (await conn.QueryAsync<AUser>(sql, new { email = email })).FirstOrDefault();
    if (vm == null) return Code.Fail("账号或密码错误"); //找不到邮箱, 也提示账号或密码错误, 防止被恶意尝试破解密码

    //验证密码正确后, 清除上次登录缓存, 更新loginToken、登录时间、登录次数, 设置cookie、缓存
    var pwdHash = GetPwdHash(pwd); //密码+guid混淆后md5加密
    if (vm.Pwd != pwdHash) return Code.Fail("账号或密码错误");
    LogA.Debug("LoginByEmail VM为" + vm.ToJson()); //调试日志

    if (!vm.LoginToken.IsNullOrEmpty()) CacheA.Remove(vm.LoginToken); //清除上次登录缓存
    var logintoken = Guid.NewGuid().ToString(); //每次登录, 更新loginToken
    var now = DateTime.Now;
    sql = "update user set logintoken=@logintoken, loginamount=loginamount+1, lastlogintime=@now where email=@email";
    var amount = await conn.ExecuteAsync(sql, new { email = email, logintoken = logintoken, now = now });
    if (amount > 0)
    {
        ToolA.SetCookie(Config.UserToken, logintoken); //设置当前登录Cookie
        vm.LoginToken = logintoken; //新loginToken
        var userModel = GetUserModelByAUser(vm); //vm转换
        CacheA.Set(logintoken, userModel); //设置当前登录缓存
        return Code.Success("登录成功");
    }
    return Code.Fail("账号或密码错误");
}

```

2) CurrentUser表示当前登录用户, 如果CurrentUser为null表示用户未登录

每次访问页面时, CurrentUser根据cookie值读取当前登录缓存 -> 如果缓存不存在, 则从数据库读取一次 -> 如果数据库也没数据, 则自动清除cookie退出登录状态

Bll/UserBll.cs

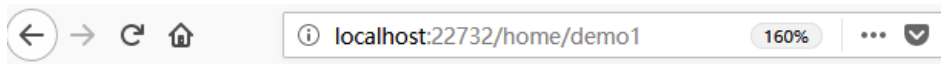
```

public static UserModel CurrentUser
{
    get
    {
        var cookieValue = ToolA.GetCookie(Config.UserToken); //读取当前登录cookie
        if (cookieValue.IsNullOrEmpty())
            return null;

        var userCache = CacheA.Get(cookieValue) as UserModel; //读取当前登录缓存
        if (userCache == null)
        {
            //cookie已存在, 未登录时根据cookie查数据库loginToken如果不存在, 自动移除cookie退出登录
            //loginToken数据库存在时, 保存到缓存
            var vm = GetUserFromDb(cookieValue); //从数据库读取一次
            if (vm == null)
                ToolA.RemoveCookie(cookieValue); //移除cookie
            else
                CacheA.Set(cookieValue, vm, Config.LoginTokenCacheMinute); //设置当前登录缓存
            return vm;
        }
        return userCache;
    }
}

```

3) 复制 /home/demo1 拼接到网址后面, 查看用户是否已登录



未登陆，currentUser为null，登录页/userLogin.htm



【已登陆用户】

```
{"Uid":2,"Email":"a2@a.com","Mobile":null,"NickName":null,"LoginToken":"8b19432a-676c-4737-b011-7f9e11391798"} 【退出Url】 /user/logout?returnUrl=/home/demo1
```

[Github.com/AskYedu](https://github.com/AskYedu)
dnc QQ群618093978

7、恭喜！到这一步，你已经神奇地学会了如何开发用户注册、登录、退出、当前在线状态模块，建议用VS开发工具，再查看一遍这些代码，加深理解

dnc = .NET Core、dotnet Core 简写

dnc是微软新一代主力编程平台，开源、免费、跨平台、轻量级、高性能，可部署到Linux、Docker、k8s等环境，适合开发微服务、云原生、大型互联网应用、全开源解决方案

dnc国内公司案例

微软、腾讯、网易、同程旅游、龙珠直播、ThoughtWorks、新东方教育科技、中通快递、申通快递、青客白领公寓、途虎养车、博客园、视高盛景、如鹏网、行云创新、大连医卫、盛派网络、切尔思科技、斯诺物联、山海致远、neo.org开源区块链、aelf.io开源区块链……等公司

dnc各公司招聘职位: <http://cnblogs.com/dncNew/p/dnc.job.html>

本系列开源教程的后续章节，正在准备中，敬请期待~

- 1、如何实现SSO单点登录？如何设计复杂的电商平台？
- 2、电商平台上线运行后，订单越来越多，访问量越来越大，系统快扛不住了，怎么办？
- 3、研发团队500人以下的公司是否只能照搬BAT的复杂架构，招聘大量工程师才能开发出大型系统？
- 4、有没有性价比更高的技术方案？作为老板，你的利润是否快要赶不上不断上涨的研发成本？
- 5、如何开发一个高性能、水平扩展、十亿级到百亿级数据量、百万级并发访问的系统？

重要声明：这是从零开始的开源教程+开源代码，与任何培训机构无关，也不用于盈利目的

【版权声明】

本教程+代码 基于MIT开源协议，欢迎转载，但必须保留来源链接，否则追究法律责任

来源: <http://AskYedu.com> 首届dnc开源峰会 <http://dncNew.com> 开源社区QQ群 618093978