# 1安装 **Zkweb.system.drawing 包**

# 2 实现VierificationCodeServices类

public class VierificationCodeServices

{

/// <summary>

/// 该方法用于生成指定位数的随机数

/// </summary>

/// <param name="VcodeNum">参数是随机数的位数</param>

/// <returns>返回一个随机数字符串</returns>

private string RndNum(int VcodeNum)

{

//验证码可以显示的字符集合

string Vchar = "0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f,g,h,i,j,k,l,m,n,p" +

",q,r,s,t,u,v,w,x,y,z,A,B,C,D,E,F,G,H,I,J,K,L,M,N,P,P,Q" +

",R,S,T,U,V,W,X,Y,Z";

string[] VcArray = Vchar.Split(new Char[] { ',' });//拆分成数组

string code = "";//产生的随机数

int temp = -1;//记录上次随机数值，尽量避避免生产几个一样的随机数

Random rand = new Random();

//采用一个简单的算法以保证生成随机数的不同

for (int i = 1; i < VcodeNum + 1; i++)

{

if (temp != -1)

{

rand = new Random(i \* temp \* unchecked((int)DateTime.Now.Ticks));//初始化随机类

}

int t = rand.Next(61);//获取随机数

if (temp != -1 && temp == t)

{

return RndNum(VcodeNum);//如果获取的随机数重复，则递归调用

}

temp = t;//把本次产生的随机数记录起来

code += VcArray[t];//随机数的位数加一

}

return code;

}

/// <summary>

/// 该方法是将生成的随机数写入图像文件

/// </summary>

/// <param name="code">code是一个随机数</param>

/// <param name="numbers">生成位数（默认4位）</param>

public MemoryStream Create(out string code, int numbers = 4)

{

code = RndNum(numbers);

Bitmap Img = null;

Graphics g = null;

MemoryStream ms = null;

Random random = new Random();

//验证码颜色集合

Color[] c = { Color.Black, Color.Red, Color.DarkBlue, Color.Green, Color.Orange, Color.Brown, Color.DarkCyan, Color.Purple };

//验证码字体集合

string[] fonts = { "Verdana", "Microsoft Sans Serif", "Comic Sans MS", "Arial", "宋体" };

//定义图像的大小，生成图像的实例

Img = new Bitmap((int)code.Length \* 18, 32);

g = Graphics.FromImage(Img);//从Img对象生成新的Graphics对象

g.Clear(Color.White);//背景设为白色

//在随机位置画背景点

for (int i = 0; i < 100; i++)

{

int x = random.Next(Img.Width);

int y = random.Next(Img.Height);

g.DrawRectangle(new Pen(Color.LightGray, 0), x, y, 1, 1);

}

//验证码绘制在g中

for (int i = 0; i < code.Length; i++)

{

int cindex = random.Next(7);//随机颜色索引值

int findex = random.Next(5);//随机字体索引值

Font f = new Font(fonts[findex], 15, FontStyle.Bold);//字体

Brush b = new SolidBrush(c[cindex]);//颜色

int ii = 4;

if ((i + 1) % 2 == 0)//控制验证码不在同一高度

{

ii = 2;

}

g.DrawString(code.Substring(i, 1), f, b, 3 + (i \* 12), ii);//绘制一个验证字符

}

ms = new MemoryStream();//生成内存流对象

Img.Save(ms, ImageFormat.Jpeg);//将此图像以Png图像文件的格式保存到流中

//回收资源

g.Dispose();

Img.Dispose();

return ms;

}

}

# 3 在Controller中新建一个 IActionResult 用于输出验证码：

/// <summary>

/// 图形验证码

/// </summary>

/// <returns></returns>

public IActionResult ValidateCode([FromServices]VierificationCodeServices \_vierificationCodeServices)

{

string code = "";

System.IO.MemoryStream ms = \_vierificationCodeServices.Create(out code);

HttpContext.Session.SetString("LoginValidateCode", code);

Response.Body.Dispose();

return File(ms.ToArray(), @"image/png");

}