

目录

1	大概思路.....	1
2	Nginx 集群之.Net 打造 WebApp(支持 IOS 和安卓).....	1
3	安卓模拟器.....	1
4	MUI 框架.....	3
5	编写.NET WebApi 的服务端.....	4
6	在本机启动 WebApi.....	7
7	Nginx 集群配置搭建	8
8	运行结果.....	9
9	总结.....	16

1 大概思路

- Nginx 集群之.Net 打造 WebApp(支持 IOS 和安卓)
- 安卓模拟器
- MUI 框架
- 编写.NET WebApi 的服务端
- 在本机启动 WebApi
- Nginx 集群配置
- 运行结果
- 总结

2 Nginx 集群之.Net 打造 WebApp(支持 IOS 和安卓)

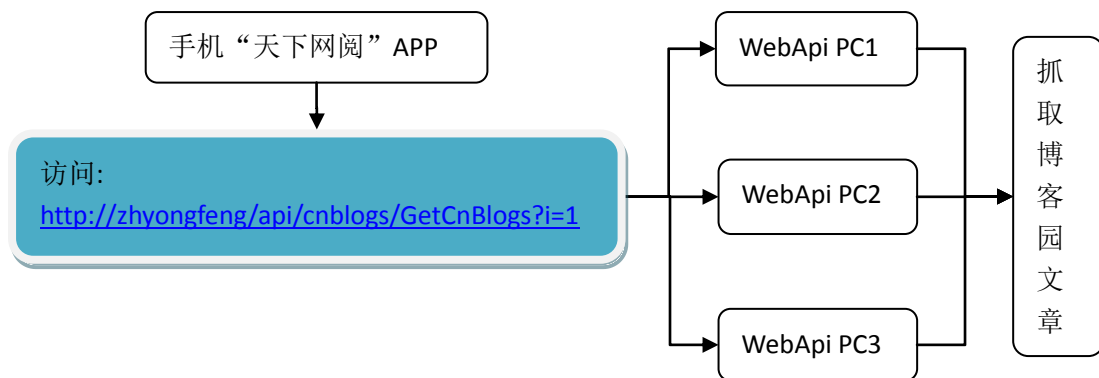
讨论了那么多 Nginx 集群的应用，本文就打造一个安卓 App 实例来展现给大家。

WebApi 好处就是无前端化，无论前端是 PC 机、笔记本、手机、平板电脑或者是 IOS 等，都可以通过集群，实现数据流的交互。如果要支持视频流这些，就要另当别论了。

MUI 框架在国内是个不错的框架，本次主要还是利用 MUI 框架实现一个简单的 APP，这个 APP 动态调用 WebApi，利用 XPATH 方式爬虫抓取博客园的文章，从而实现一个简单的“天下网阅”APP。

以下是本文讲述的主要结构图：

手机“天下网阅”APP，访问 Nginx 集群 WebApi，然后抓取博客园文章，如下所示：



3 安卓模拟器

安卓模拟器可以采用夜神模拟器（可以采用原生的 android 模拟器，或者其它类型的模拟器），输入以下 adb 命令：

Microsoft Windows [版本 6.1.7601]

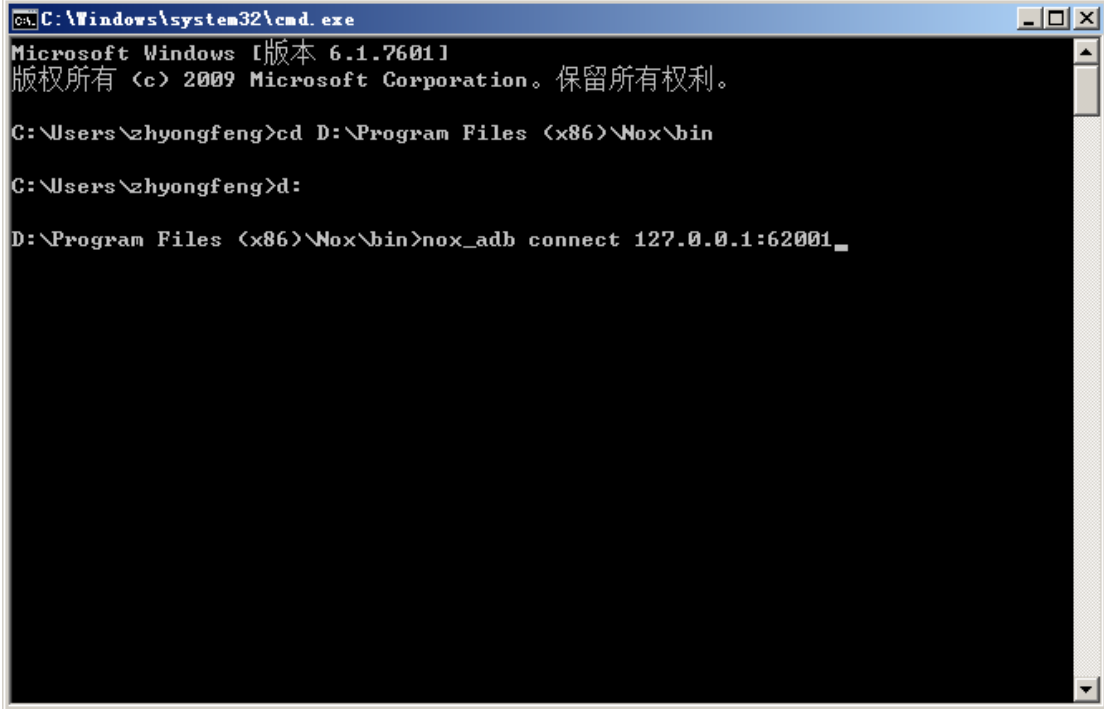
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

```
C:\Users\zhyongfeng>cd D:\Program Files (x86)\Nox\bin
```

```
C:\Users\zhyongfeng>d:
```

```
D:\Program Files (x86)\Nox\bin>nox_adb connect 127.0.0.1:62001
```





```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\zhyongfeng>cd D:\Program Files (x86)\Nox\bin

C:\Users\zhyongfeng>d:

D:\Program Files (x86)\Nox\bin>nox_adb connect 127.0.0.1:62001_
```

4 MUI 框架

MUI 是最接近原生 App 体验的前端框架,可多端发布到 Appstore、Android 市场、浏览器、微信公众号、百度直达号及流应用。

MUI 相关的学习资源: <http://dev.dcloud.net.cn/mui/>

MUI 使用的工具: HBuilder

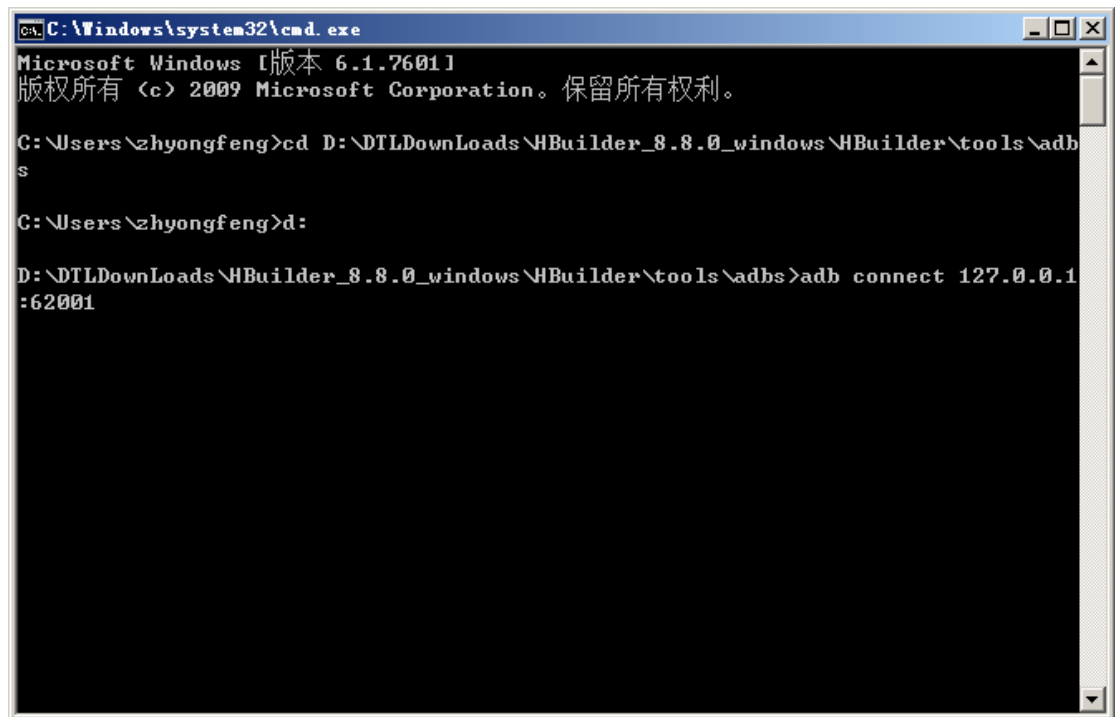
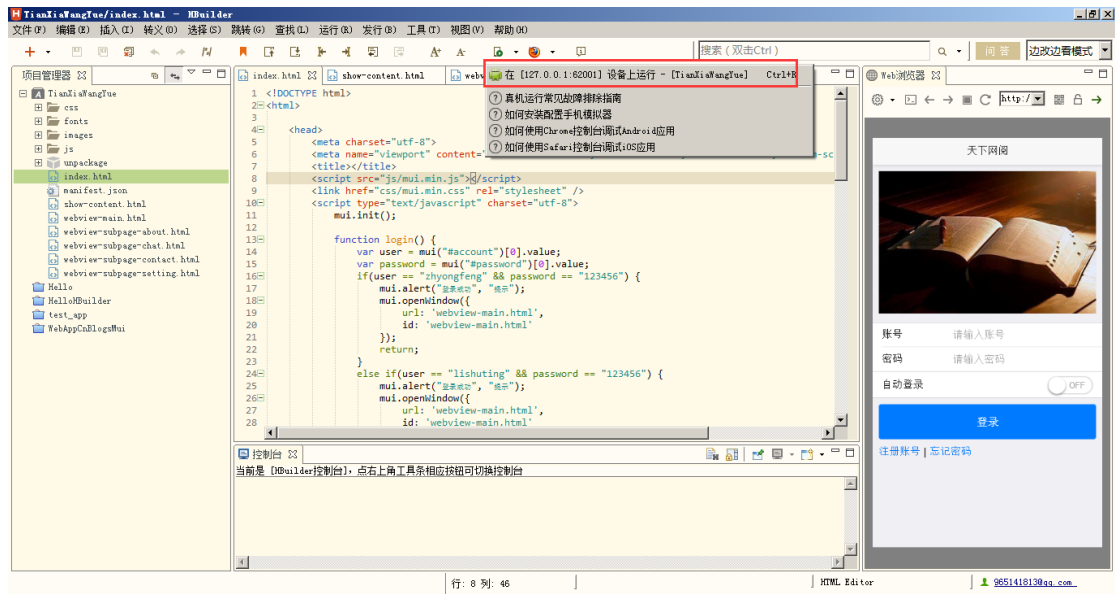
运行 CMD, 输入以下命令:

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\zhyongfeng>cd D:\DTLDownloads\HBuilder_8.8.0_windows\HBuilder\tools\adb
s

C:\Users\zhyongfeng>d:

D:\DTLDownloads\HBuilder_8.8.0_windows\HBuilder\tools\adbs>adb connect 127.0.0.1
:62001
```



5 编写.NET WebApi 的服务端

CnBlogsController.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Net;
using System.Text;
using System.Web.Http;
```

```
using Newtonsoft.Json;
using HtmlAgilityPack;
using WebAppCnBlogsApi.Models;

namespace WebAppCnBlogsApi.Controllers
{
    [RoutePrefix("api/CnBlogs")]
    public class CnBlogsController : ApiController
    {
        [Route("GetCnBlogs")]
        public string GetCnBlogs(int i)
        {
            string url = "https://www.cnblogs.com/mvc/AggSite/PostList.aspx";
            HttpRequest req = (HttpRequest)WebRequest.Create(url);
            req.Method = "POST";
            req.ContentType = "application/json; charset=utf-8";
            object o = new
            {
                CategoryType = "SiteHome",
                ParentCategoryId = 0,
                CategoryId = 808,
                PageIndex = i,
                TotalPostCount = 4000,
                ItemListActionName = "PostList"
            };
            //如果需要 POST 数据
            byte[] data = Encoding.UTF8.GetBytes(JsonConvert.SerializeObject(o));
            req.ContentLength = data.Length;
            using (Stream stream = req.GetRequestStream())
            {
                stream.Write(data, 0, data.Length);
            }

            HttpResponse response = (HttpResponse)req.GetResponse();
            Stream streamReceive = response.GetResponseStream();
            StreamReader streamReader = new StreamReader(streamReceive,
            Encoding.UTF8);
            string strResult = streamReader.ReadToEnd();
            streamReader.Close();
            streamReceive.Close();
            req.Abort();
            response.Close();
        }
    }
}
```

```

//解析 HTML
HtmlDocument doc = new HtmlDocument();
doc.LoadHtml(strResult);
HtmlNodeCollection cnBlogsNodeCollection =
doc.DocumentNode.SelectNodes("//div[@class='post_item_body']");
List<HtmlCnBlogs> listCnBlogs = new List<HtmlCnBlogs>();

int j = 0;
foreach(HtmlNode m in cnBlogsNodeCollection)
{
    j++;
    HtmlCnBlogs cn = new HtmlCnBlogs();
    HtmlNode author = m.SelectSingleNode("/div[" + j + "]/div[2]/div[1]/a[1]");
    cn.author = author.InnerText.Trim();
    cn.authorHref = author.Attributes["href"].Value;
    HtmlNode authorCreateTime = m.SelectSingleNode("/div[" + j +
"/div[2]/div[1]");
    cn.createtime = authorCreateTime.ChildNodes[2].InnerText.Trim();
    HtmlNode img = m.SelectSingleNode("/div[" + j + "]/div[2]/p[1]/a[1]/img[1]");
    if (img != null)
        cn.authorImg = string.Format("https:{0}", img.Attributes["src"].Value);

    HtmlNode h3 = m.SelectSingleNode("/div[" + j + "]/div[2]/h3[1]");
    cn.title = h3.InnerText;
    HtmlNode titleInk = m.SelectSingleNode("/div[" + j + "]/div[2]/h3[1]/a[1]");
    cn.titlelink = titleInk.Attributes["href"].Value;
    //简介
    HtmlNode content = m.SelectSingleNode("/div[" + j + "]/div[2]/p[1]");
    cn.content = content.InnerText.Trim();
    //评论、评论 URL
    HtmlNode comments = m.SelectSingleNode("/div[" + j +
"/div[2]/div[1]/span[1]/a[1]");
    string strComments = comments.InnerText.Trim();
    StringBuilder sbComments = new StringBuilder();
    foreach (char c in strComments)
    {
        if (Convert.ToInt32(c) >= 48 && Convert.ToInt32(c) <= 57)
        {
            sbComments.Append(c);
        }
    }
    cn.commentsHref = comments.Attributes["href"].Value;
    cn.comments = int.Parse(sbComments.ToString());
};

```

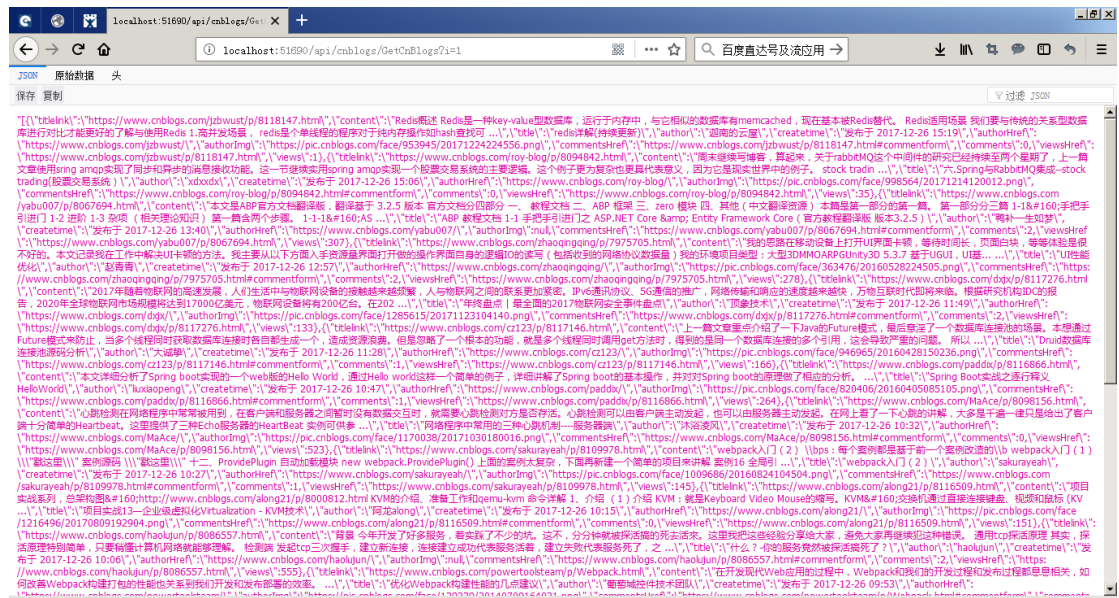
```

HtmlNode views
m.SelectSingleNode("/div["+j+"]/div[2]/div[1]/span[2]/a[1]");
string strViews = views.InnerText.Trim();
StringBuilder sbViews = new StringBuilder();
foreach (char c in strViews)
{
    if (Convert.ToInt32(c) >= 48 && Convert.ToInt32(c) <= 57)
    {
        sbViews.Append(c);
    }
}
cn.viewsHref= views.Attributes["href"].Value;
cn.views = int.Parse(sbViews.ToString());
listCnBlogs.Add(cn);
}
return JsonConvert.SerializeObject(listCnBlogs);
}
}

```

6 在本机启动 WebApi

将 WebApi 部署到以下 10.93.85.66（因网络限制，所以这里只在本机上执行启动）



7 Nginx 集群配置搭建

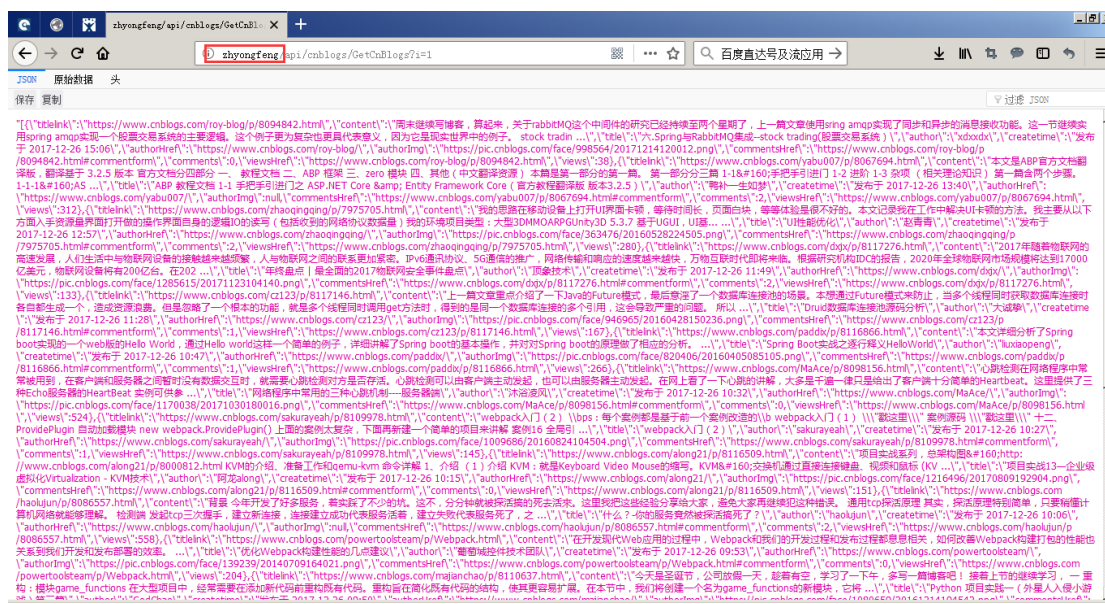
Nginx 的集群配置:

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    server {
        listen 80;
        server_name localhost zhyongfeng;
        location / {
            proxy_pass http://localhost:51690;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root html;
        }
    }
}
```

运行 CMD:

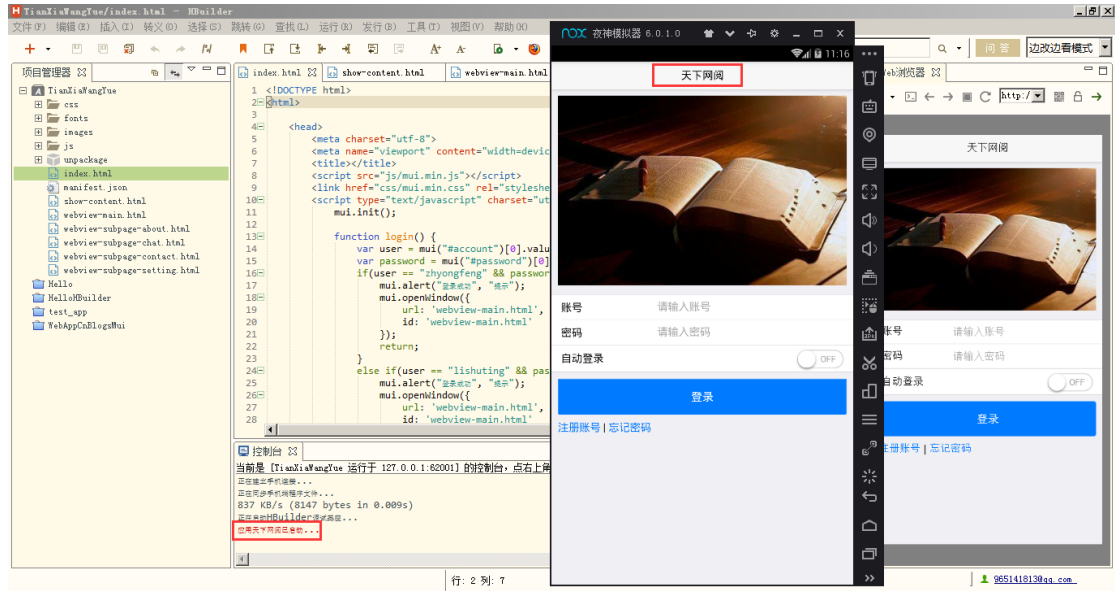
D:\DTLDownloads\nginx-1.10.2>start nginx

D:\DTLDownloads\nginx-1.10.2>nginx -s reload



8 运行结果

启动安装在手机端的 APP（或者启动 HBuilder）










11:00

天下网闻

Oracle自动备份数据

一、 Oracle自动备份单表一个月数据 方式一：三步处理（建批处理文件，写sql文件，设置任务计划） 1、 第一步：建立一个批处理文件 2、 第二步、写清除数据库并释放空间的sql文件 3、 第三步、使用windows自带的任务计划程序 A）打开任务计划程序，新建一个基本任务计划 B）填上名和 ...

zhufeiwu
发布于 2017-12-26 10:59
评论(0)
阅读(0)

Spring Boot实战之逐行释义HelloWorld



本文详细分析了Spring boot实现的一个web版的Hello World，通过Hello world这样一个简单的例子，详细讲解了Spring boot的基本操作，并对对Spring boot的原理做了相应的分析。 ...

liuxiaopeng
发布于 2017-12-26 10:47
评论(1)
阅读(29)

网络程序中常用的三种心跳机制--服务器端



心跳检测在网络程序中常常被用到，在客户端和服务端之间暂时没有数据交互时，就需要心跳检测对方是否存活。心跳检测可以由客户端主动发起，也可以由服务器主动发起。在网上看了一下心跳的讲解，大多是千篇一律只是给出了客户端十分简单的Heartbeat。这里提供了三种Echo服务器的HeartBeat 实例可供参 ...

沐浴凌风
发布于 2017-12-26 10:32
评论(0)
阅读(118)

webpack入门 (2)



webpack入门 (2) ps：每个案例都是基于前一个案例改造的 webpack入门 (1) "戳这里" 案例源码 "戳这里" 十二、 ProvidePlugin 自动加载模块 new webpack.ProvidePlugin() 上面的案例太复杂，下面再新建一个简单的项目来讲解 案例16 全局引 ...

sakuraveah
发布于 2017-12-26 10:27
评论(0)
阅读(39)



天下网闻



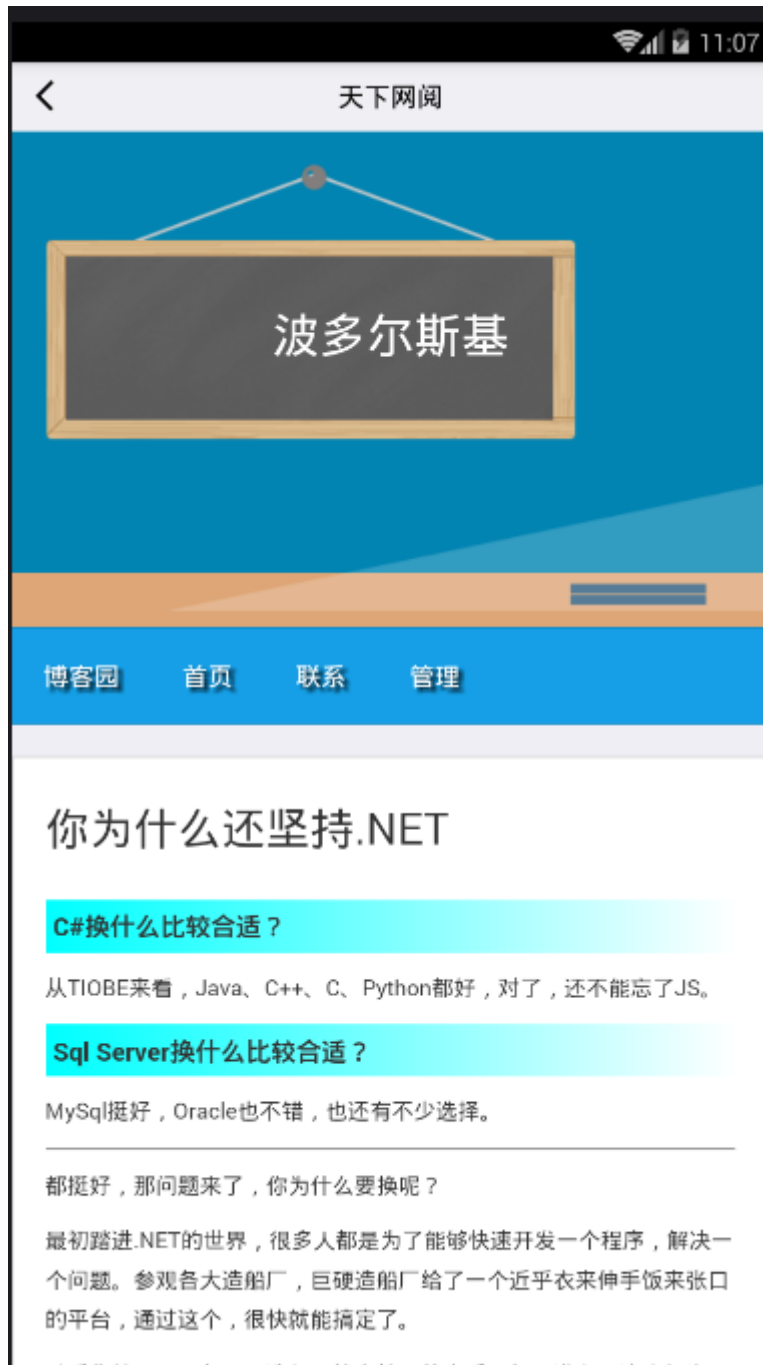
我的主页



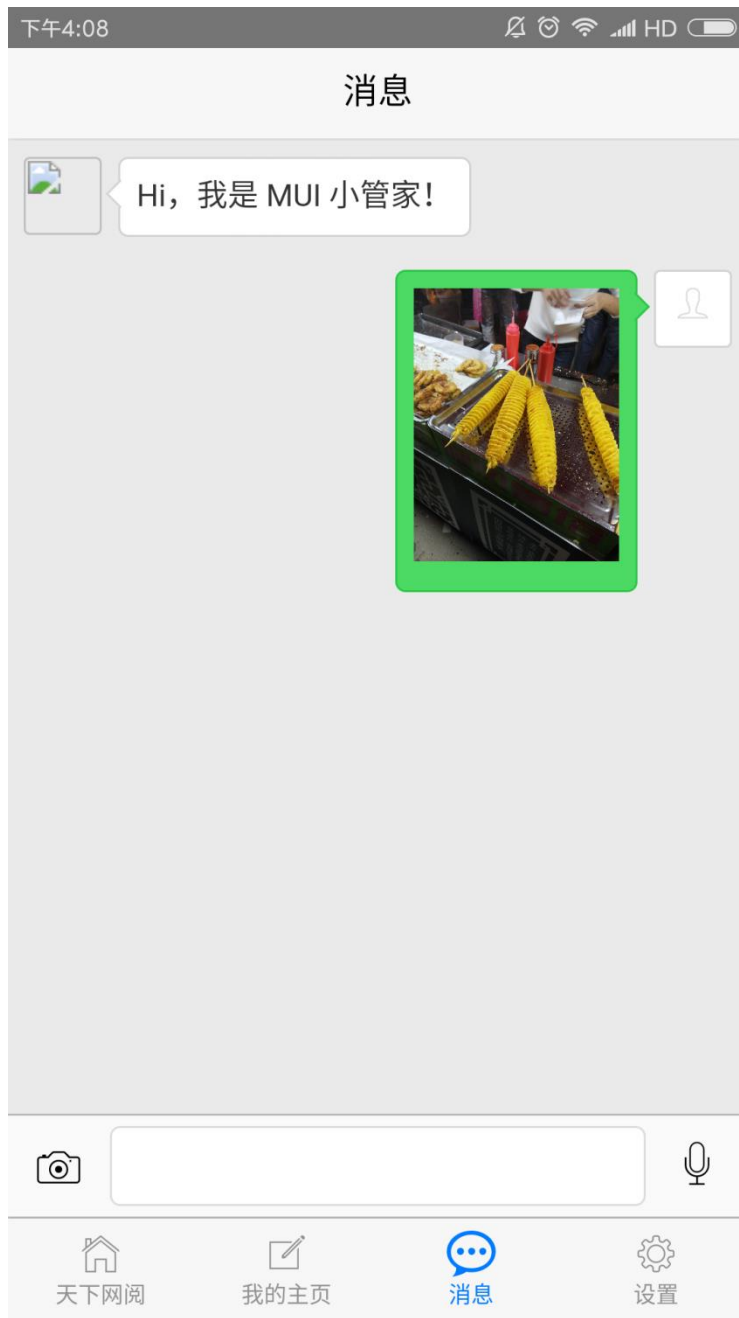
消息



设置









9 总结

通过 Nginx 集群, 再使用 MUI 前端框架打包生成 android 的 APP(也可以生成 IOS 的 APP), WebApi 提供了集群 API 的数据流访问, 通过 API 能够动态抓取博客园的文章, 并反馈到 APP 主页上, 从而完成了一个简单的“天下网阅”APP 设计。

源代码下载:

http://download.csdn.net/download/ruby_matlab/10175001

PDF 下载:

[Nginx 集群之 .Net 打造 WebApp\(支持 IOS 和安卓\).pdf](#)