

通过request获取用户ip

```
package com.example.demo.multi.springboot.controller;

import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import javax.servlet.http.HttpServletRequest;

@RestController
@RequestMapping("hello")
public class HelloController {

    @RequestMapping("/{id}")
    public String index(HttpServletRequest request, @PathVariable Integer id) {
        String ip = request.getHeader("x-forwarded-for");
        if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getHeader("Proxy-Client-IP");
        }
        if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getHeader("WL-Proxy-Client-IP");
        }
        if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getHeader("HTTP_CLIENT_IP");
        }
        if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getHeader("HTTP_X_FORWARDED_FOR");
        }
        if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getRemoteAddr();
        }
        System.out.println(ip);
        return "Conquer the world from here!";
    }
}
```

解释

X-Forwarded-For (XFF)

用来识别通过HTTP代理或负载均衡方式连接到Web服务器的客户端最原始的IP地址的HTTP请求头字段。

- 格式

X-Forwarded-For: client1, proxy1, proxy2

其中的值通过一个逗号+空格把多个IP地址区分开，最左边（client1）是最原始客户端的IP地址，代理服务器每成功收到一个请求，就把请求来源IP地址添加到右边。在上面这个例子中，这个请求成功通过了三台代理服务器：proxy1, proxy2 及 proxy3。请求由client1发出，到达了proxy3（proxy3可能是请求的终点）。请求刚从client1中发出时，XFF是空的，请求被发往proxy1；通过proxy1的时候，client1被添加到XFF中，之后请求被发往proxy2；通过proxy2的时候，proxy1被添加到XFF中，之后请求被发往proxy3；通过proxy3时，proxy2被添加到XFF中，之后请求的去向不明，如果proxy3不是请求终点，请求会被继续转发。

WL-Proxy-Client-IP

在apache+WebLogic整合系统中，apache会对request对象进行再包装，附加一些WLS要用的头信息。这种情况下，直接用request.getRemoteAddr()是无法取到真正的客户IP的。需要使用WL-Proxy-Client-IP。

HTTP_CLIENT_IP、HTTP_X_FORWARDED_FOR、RemoteAddr

REMOTE_ADDR 不可以显式的伪造，虽然可以通过代理将ip地址隐藏，但是这个地址仍然具有参考价值，因为它就是与你的服务器实际连接的ip地址。

生产环境中很多服务器隐藏在负载均衡节点后面，你通过 REMOTE_ADDR 只能获取到负载均衡节点的ip地址，一般的负载均衡节点会把前端实际的ip地址通过 HTTP_CLIENT_IP 或者 HTTP_X_FORWARDED_FOR 这两种http头传递过来

注：如果你的服务器直接暴露在客户端前面的时候，请不要信任这两种读取方法，只需要读取 REMOTE_ADDR 就行了。

先记录方法，每一项解释需要好好了解一系// todo