后台跨域调用API接口方案：

运用功能：后台发言评论即时通讯调用APP的API接口

参数：①url：API接口的url，

例如：[http://’IP](http://'IP)地址’:8080/ehomeapp/app/msgSend/commentMsg

②param：需要传输的参数，字符串类型

例如："type="+type+"&userId="+userId+"&userNickname="+userNickname+""

代码实现：

@Controller

public Map save() {

String url = “[http://’IP](http://'IP)地址’:8080/ehomeapp/app/msgSend/commentMsg”;

String param = "type="+type+"&userId="+userId+"&userNickname="+userNickname+""

String msg = sendPost(url, param);

Map map = new HashMap();

map.put(“”,””);

map.put(“”,””);

return map;

}

public static String sendPost(String url, String param) {

PrintWriter out = null;

BufferedReader in = null;

String result = "";

String charset = "UTF-8";

try {

URL realUrl = new URL(url);

// 打开和URL之间的连接

URLConnection conn = realUrl.openConnection();

// 设置通用的请求属性

conn.setRequestProperty("accept", "\*/\*");

conn.setRequestProperty("connection", "Keep-Alive");

conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");

// 发送POST请求必须设置如下两行

conn.setDoOutput(true);

conn.setDoInput(true);

// 获取URLConnection对象对应的输出流

out = new PrintWriter(conn.getOutputStream());

// 发送请求参数

out.print(param);

// flush输出流的缓冲

out.flush();

// 定义BufferedReader输入流来读取URL的响应

in = new BufferedReader(new InputStreamReader(conn.getInputStream(), charset));

String line;

while ((line = in.readLine()) != null) {

result += line;

}

if (StringUtils.isBlank(result)) {

Map<String, String> mm = new HashMap<String, String>();

mm.put("status", "0");

mm.put("msg", "服务器异常，请稍后再试！");

System.out.println(mm);

}

} catch (Exception e) {

System.out.println("发送 POST 请求出现异常！" + e);

e.printStackTrace();

} finally {

try {

if (out != null) {

out.close();

}

if (in != null) {

in.close();

}

} catch (IOException ex) {

ex.printStackTrace();

}

}

return result;

}