

用户或者开发人员访问alloc服务的index.html页面，模拟发送PUSH请求：

GET http://xxxx:9999/index.html

返回内容：模拟发送PUSH请求的HTML页面

JDK HTTP server在接收到请求时，会根据context找到相应HttpHanlder，调用
IndexPageHandler#handle()

```
AllocServer  init()

@Override
public void init() {
    int port = CC.mp.net.cfg.getInt("alloc-server-port");
    boolean https = "https".equals(CC.mp.net.cfg.getString("alloc-server-protocol"));

    this.httpServer = HttpServerCreator.createServer(port, https);
    this.allocHandler = new AllocHandler();
    this.pushHandler = new PushHandler();

    httpServer.setExecutor(Executors.newCachedThreadPool()); //设置线程池，由于是纯内存操作，不需要队列
    httpServer.createContext("/", allocHandler); //查询mpush机器
    httpServer.createContext("/push", pushHandler); //模拟发送push
    httpServer.createContext("/index.html", new IndexPageHandler()); //查询mpush机器
}
```

```

public final class IndexPageHandler implements HttpHandler {

    private byte[] data;

    public IndexPageHandler() {
        try (InputStream in = this.getClass().getResourceAsStream("/index.html")) {
            this.data = new byte[in.available()];
            in.read(data);
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    @Override
    public void handle(HttpExchange httpExchange) throws IOException {
        if (data != null && data.length > 0) {
            httpExchange.getResponseHeaders().set("Content-Type", "text/html; charset=utf-8");
            httpExchange.sendResponseHeaders(200, data.length); // 200, content-length
            OutputStream out = httpExchange.getResponseBody();
            out.write(data);
            out.close();
            httpExchange.close();
        } else {
            byte[] data = "404 Not Found".getBytes(StandardCharsets.UTF_8);
            httpExchange.getResponseHeaders().set("Content-Type", "text/plain; charset=utf-8");
            httpExchange.sendResponseHeaders(404, data.length);
            OutputStream out = httpExchange.getResponseBody();
            out.write(data);
            out.close();
            httpExchange.close();
        }
    }
}

```

返回

To User Id:	<input type="text" value="user-0"/>
Push Content:*	<input type="text" value="hello mpush"/>
Broadcast:	<input type="checkbox"/>
Condition:	<input type="text" value="tags!=null&&tags.indexOf('mpush')>"/>
Send To Client	

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>MPush Message Sender</title>
6 </head>

```

```
7 <body>
8 <form onsubmit="return false;">
9   <label>
10    To User Id:
11    <input type="text" value="user-0" id="userId" required="required"
12    style="width: 100%">
13  </label>
14  <br/>
15  <label>
16    Push Content:<span style="color: red">*</span>
17    <input type="text" value="hello mpush" id="hello" required="required" s
18    tyle="width: 100%">
19  </label>
20  <br/>
21  <label>
22    Broadcast:
23    <input type="checkbox" value="1" id="broadcast">
24  </label>
25  <br/>
26  <label>
27    Condition:
28    <input type="text" id="condition" placeholder="tags!=null&&tags.indexOf
29    ('mpush')>0" style="width: 100%">
30  </label>
31  <br/>
32  <input type="button" value="Send To Client" style="width: 100%"
33  onclick="send()">
34 </form>
35 <script type="application/javascript">
36   function $(id) {
37     return document.getElementById(id);
38   }
39   function send() {
40     var broadcast = $('broadcast').checked;
41     var userId = $('userId').value;
42     var content = $('hello').value;
43     if (!userId && !broadcast) {
44       alert("please input userId.");
45     }
46     return false;
47   }
48 </script>
49 </body>
```

```
44  }
45
46  if (!content) {
47      alert("please input content.");
48      return false;
49  }
50
51
52  var xhr = new XMLHttpRequest();
53  xhr.open('POST', '/push', true);
54  xhr.setRequestHeader("context-type", "application/json; charset=utf-8");
55  xhr.onload = function (e) {
56      if (this.status == 200) {
57          alert(this.response)
58      }
59  };
60
61  xhr.send(JSON.stringify({
62      userId: userId,
63      hello: content,
64      broadcast: broadcast,
65      condition: $('#condition').value
66  }));
67  }
68 </script>
69 </body>
70 </html>
```