Java HTTP函数

更新时间：2020-09-15 15:44:11

本页目录

* [HTTP函数接口](https://help.aliyun.com/document_detail/158958.html?spm=a2c4g.11186623.6.583.156047cdI2CeMD#title-8vr-tvk-14a)
* [HTTP触发器的简单示例](https://help.aliyun.com/document_detail/158958.html?spm=a2c4g.11186623.6.583.156047cdI2CeMD#title-bki-mxd-ulc)
* [HTTP触发器支持传统Web应用](https://help.aliyun.com/document_detail/158958.html?spm=a2c4g.11186623.6.583.156047cdI2CeMD#title-oo4-2py-ep5)
* [更多信息](https://help.aliyun.com/document_detail/158958.html?spm=a2c4g.11186623.6.583.156047cdI2CeMD#title-96s-njp-tum)

在函数计算中使用Java编程时，需要定义一个函数作为入口函数。为函数设置HTTP触发器后，可以方便处理发来的HTTP请求。

**HTTP函数接口**

函数计算提供基于Servlet协议的HTTP入口，形式如下所示。

**public** **interface** **HttpRequestHandler** {

/\*\*

\* The entrance function of fc http trigger

\* **@param** request The servlet request

\* **@param** response The servlet response

\* **@param** context The fc context

\* **@throws** IOException If IO exception happened

\* **@throws** ServletException If servlet exception happened

\*/

**public** **void** **handleRequest**(HttpServletRequest request, HttpServletResponse response, Context context) **throws** IOException, ServletException;

}

使用HTTP触发器需要将fc-java-core库版本升级到1.3.0及以上。

**HTTP触发器的简单示例**

**package** com.aliyun.fc.example;

**import** java.io.IOException;

**import** java.io.OutputStream;

**import** javax.servlet.ServletException;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** com.aliyun.fc.runtime.Context;

**import** com.aliyun.fc.runtime.HttpRequestHandler;

**public** **class** **Hello** **implements** **HttpRequestHandler** {

**public** **void** **handleRequest**(HttpServletRequest request, HttpServletResponse response, Context context)

**throws** IOException, ServletException {

String requestPath = (String) request.getAttribute("FC\_REQUEST\_PATH");

String requestURI = (String) request.getAttribute("FC\_REQUEST\_URI");

String requestClientIP = (String) request.getAttribute("FC\_REQUEST\_CLIENT\_IP");

response.setStatus(200);

response.setHeader("header1", "value1");

response.setHeader("header2", "value2");

String body = String.format("Path: %s\n Uri: %s\n IP: %s\n", requestPath, requestURI, requestClientIP);

OutputStream out = response.getOutputStream();

out.write((body).getBytes());

out.flush();

out.close();

}

}

* **HttpServletRequest**

函数计算HTTP触发器的接口使用标准的Servlet协议。您的请求会封装成**HttpServletRequest**对象，请求参数、请求头等均可通过此对象获取。除此之外，函数计算在**HttpServletRequest**中预封装了一些属性，您可以通过**getAttribute**方法来获取，具体包括以下内容：

* + **FC\_REQUEST\_PATH**：获取请求的路径。
  + **FC\_REQUEST\_URI**：获取请求的URI。
  + **FC\_REQUEST\_CLIENT\_IP**：获取请求的Client IP地址。
* **HttpServletResponse**

您可以通过标准的**HttpServletResponse**协议对象来返回响应头和响应体。

* **context**参数

[context参数](https://help.aliyun.com/document_detail/157704.html#section-orb-m3o-ec9)中包含一些函数的运行时信息（例如Request ID、临时AccessKey等），其类型是com.aliyun.fc.runtime.Context。

**HTTP触发器支持传统Web应用**

基于Servlet协议的传统Web应用能很方便地迁移到函数计算平台，目前支持的主流框架包括Spring、SpringBoot、Struts2等。以下示例演示如何通过函数计算提供的fc-java-common库加载Web应用。

1. 打包您的Web工程，生成demo.war。
2. 上传demo.war包到OSS，例如上传到demo-bucket目录下。
3. 创建函数，并设置函数Initializer入口和入口函数。
   * 函数示例代码如下。
   * **package** com.aliyun.fc.example;
   * **import** java.io.IOException;
   * **import** javax.servlet.ServletException;
   * **import** javax.servlet.http.HttpServletRequest;
   * **import** javax.servlet.http.HttpServletResponse;
   * **import** com.aliyun.fc.runtime.Context;
   * **import** com.aliyun.fc.runtime.FcAppLoader;
   * **import** com.aliyun.fc.runtime.FunctionComputeLogger;
   * **import** com.aliyun.fc.runtime.FunctionInitializer;
   * **import** com.aliyun.fc.runtime.HttpRequestHandler;
   * **public** **class** **HelloWeb** **implements** **FunctionInitializer**, **HttpRequestHandler** {
   * **private** FcAppLoader fcAppLoader = **new** FcAppLoader();
   * **private** String ossEndPoint = "YourOSSEndPoint";
   * **private** String bucket = "YourOSSBucket";
   * **private** String key = "YourWarName";
   * **private** String userContextPath = "/2016-08-15/proxy/{YourServiceName}/{YourFunctionName}";
   * @Override
   * **public** **void** **initialize**(Context context) **throws** IOException {
   * FunctionComputeLogger fcLogger = context.getLogger();
   * fcAppLoader.setFCContext(context);
   * // Load code from OSS
   * fcAppLoader.loadCodeFromOSS(ossEndPoint, bucket, key);
   * // Init webapp from code
   * fcAppLoader.initApp(userContextPath, HelloWeb.**class**.**getClassLoader**());
   * }
   * @Override
   * **public** **void** **handleRequest**(HttpServletRequest request, HttpServletResponse response, Context context)
   * **throws** IOException, ServletException {
   * **try** {
   * fcAppLoader.forward(request, response);
   * } **catch** (Exception e) {
   * e.printStackTrace();
   * }
   * }

}

* + 引用Maven库
  + <dependency>
  + <groupId>com.aliyun.fc.runtime</groupId>
  + <artifactId>fc-java-core</artifactId>
  + <version>1.3.0</version>
  + </dependency>
  + <dependency>
  + <groupId>com.aliyun.fc.runtime</groupId>
  + <artifactId>fc-java-common</artifactId>
  + <version>2.2.2</version>

</dependency>