Atitit jetty9 best pratice 最佳实践

目录

[1.1. Maven 1](#_Toc15792)

[1.2. JettyWebserver 2](#_Toc977)

[1.3. QlSpelHandle 3](#_Toc29717)

## Maven

<!-- https://mvnrepository.com/artifact/org.springframework.data/spring-data-jpa -->

<dependency>

<groupId>org.springframework.data</groupId>

<artifactId>spring-data-jpa</artifactId>

<version>2.3.0.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.eclipse.jetty/jetty-http -->

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-http</artifactId>

<version>9.4.29.v20200521</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.eclipse.jetty/jetty-io -->

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-io</artifactId>

<version>9.4.29.v20200521</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.eclipse.jetty/jetty-util -->

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-util</artifactId>

<version>9.4.29.v20200521</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.eclipse.jetty/jetty-servlet -->

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-servlet</artifactId>

<version>9.4.29.v20200521</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.eclipse.jetty/jetty-server -->

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-server</artifactId>

<version>9.4.29.v20200521</version>

</dependency>

## JettyWebserver

**package** com.kok.sport.utils;

**import** org.eclipse.jetty.server.Server;

**public** **class** JettyWebserver {

/\*\*

\* <pre>

\* 一个简单的 jetty 服务

\* </pre>

\*

\* **@author** saligia

\* **@date** 17-10-10

\*/

**public** **static** **void** main(String [] args) **throws** Exception {

// int port = 13140;

**int** port=Integer.*parseInt*(args[0].trim()) ;

Server server = **new** Server(port); // 创建一个 jetty 服务

server.setHandler(**new** QlSpelHandle());

server.start();

server.dumpStdErr();

server.join();

}

}

## QlSpelHandle

package com.kok.sport.utils;

import org.eclipse.jetty.server.Request;

import org.eclipse.jetty.server.handler.AbstractHandler;

import org.springframework.expression.Expression;

import org.springframework.expression.ExpressionParser;

import org.springframework.expression.spel.standard.SpelExpressionParser;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.io.PrintWriter;

import java.io.Writer;

/\*\*

\* <pre>

\* </pre>

\*

\* @author saligia

\* @date 17-10-10

\*/

public class QlSpelHandle extends AbstractHandler {

/\*\*

\* <pre>

\* </pre>

\*

\* @author

\* @param target - 目标请求，可以是一个URI或者是一个转发到这的处理器的名字

\* @param request - Jetty自己的没有被包装的请求，一个可变的Jetty请求对象

\* @param httpServletRequest - 被filter或者servlet包装的请求，一个不可变的Jetty请求对象

\* @param httpServletResponse - 响应，可能被filter或者servlet包装过

\* @return

\*/

public void handle(String target, Request request, HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse) throws IOException, ServletException {

System.out.println("target :" + target);

System.out.println("request : " + request.getRequestURI());

System.out.println("requestServ : " + httpServletRequest.getRequestURI());

/\*\*

\* target :/

request : /

requestServ : /

\*/

/\*\*

\* target :/run

request : /run

requestServ : /run

\*/

ExpressionParser parser = new SpelExpressionParser();

String e=httpServletRequest.getParameter("e");

Expression exp = parser.parseExpression(e);

Object value = exp.getValue();

System.out.println(value);

httpServletResponse.setContentType("text/html; charset=utf-8");

httpServletResponse.setStatus(HttpServletResponse.SC\_OK);

PrintWriter out = httpServletResponse.getWriter();

out.println(value);

request.setHandled(true);

}

}