**点评CAT之URL监控埋点**

# URL监控埋点方案

## 1、作用

一个http请求来了之后，会自动打点，能够记录每个url的访问情况，并将以此请求后续的调用链路串起来，可以在cat上查看logview

请将catFilter存放filter的第一个，这样可以保证最大可能性监控所有的请求

## 2、在SpringBoot项目中书写Filter类

```

/\*\*

\* CAT相关的Filter

\*

\* @Author YUBIN

\* @create 2018-12-16

\*/

public class CatServletFilter implements Filter {

private String[] urlPatterns = new String[0];

@Override

public void init(FilterConfig filterConfig) throws ServletException {

String patterns = filterConfig.getInitParameter("CatHttpModuleUrlPatterns");

if (patterns != null) {

patterns = patterns.trim();

this.urlPatterns = patterns.split(",");

for(int i = 0; i < this.urlPatterns.length; ++i) {

this.urlPatterns[i] = this.urlPatterns[i].trim();

}

}

}

@Override

public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws IOException, ServletException {

HttpServletRequest request = (HttpServletRequest)servletRequest;

String uri = request.getRequestURI().toString();

String[] urlPatterns = this.urlPatterns;

int length = urlPatterns.length;

for(int i = 0; i < length; ++i) {

String urlPattern = urlPatterns[i];

if (uri.startsWith(urlPattern)) {

uri = urlPattern;

}

}

Transaction t = Cat.newTransaction("Service", uri);

try {

PropertyContext propertyContext = new PropertyContext();

propertyContext.addProperty("\_catRootMessageId", request.getHeader("X-CAT-ROOT-ID"));

propertyContext.addProperty("\_catParentMessageId", request.getHeader("X-CAT-PARENT-ID"));

propertyContext.addProperty("\_catChildMessageId", request.getHeader("X-CAT-CHILD-ID"));

Cat.logRemoteCallServer(propertyContext);

Cat.logEvent("Service.method", request.getMethod(), Message.SUCCESS, request.getRequestURL().toString());

Cat.logEvent("Service.client", request.getRemoteHost());

String clientApp = request.getHeader("X-PPD-CAT-APP");

Cat.logEvent("Service.app", clientApp == null ? "unknown" : clientApp);

filterChain.doFilter(servletRequest, servletResponse);

t.setStatus(Message.SUCCESS);

} catch (Exception ex) {

t.setStatus(ex);

Cat.logError(ex);

throw ex;

} finally {

t.complete();

}

}

@Override

public void destroy() {

}

private static class PropertyContext implements Cat.Context {

private Map<String, String> properties = new HashMap();

PropertyContext() {

}

public void addProperty(String key, String value) {

this.properties.put(key, value);

}

public String getProperty(String key) {

return (String)this.properties.get(key);

}

}

}

```

## 3、配置Filter

```

@Configuration

public class CatFilterConfigure {

@Bean

public FilterRegistrationBean catFilter() {

FilterRegistrationBean registration = new FilterRegistrationBean();

CatServletFilter filter = new CatServletFilter();

registration.setFilter(filter);

registration.addUrlPatterns("/\*");

registration.setName("cat-filter");

registration.setOrder(1);

return registration;

}

}

```

## 4、请求示例

