import java.io.IOException;

import java.sql.Connection;

import java.util.Collection;

import java.util.Map;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRegistration;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.annotation.WebFilter;

import javax.servlet.http.HttpServletRequest;

@WebFilter(filterName = "jdbcFilter", urlPatterns = { "/\*" })

public class JDBCFilter implements Filter {

public JDBCFilter() {

}

@Override

public void init(FilterConfig fConfig) throws ServletException {

}

@Override

public void destroy() {

}

// Check the target of the request is a servlet?

private boolean needJDBC(HttpServletRequest request) {

System.out.println("JDBC Filter");

//

// Servlet Url-pattern: /spath/\*

//

// => /spath

String servletPath = request.getServletPath();

// => /abc/mnp

String pathInfo = request.getPathInfo();

String urlPattern = servletPath;

if (pathInfo != null) {

// => /spath/\*

urlPattern = servletPath + "/\*";

}

// Key: servletName.

// Value: ServletRegistration

Map<String, ? extends ServletRegistration> servletRegistrations = request.getServletContext()

.getServletRegistrations();

// Collection of all servlet in your Webapp.

Collection<? extends ServletRegistration> values = servletRegistrations.values();

for (ServletRegistration sr : values) {

Collection<String> mappings = sr.getMappings();

if (mappings.contains(urlPattern)) {

return true;

}

}

return false;

}

@Override

public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)

throws IOException, ServletException {

HttpServletRequest req = (HttpServletRequest) request;

// Only open connections for the special requests.

// (For example, the path to the servlet, JSP, ..)

//

// Avoid open connection for commons request.

// (For example: image, css, javascript,... )

//

if (this.needJDBC(req)) {

System.out.println("Open Connection for: " + req.getServletPath());

Connection conn = null;

try {

// Create a Connection.

conn = ConnectionUtils.getConnection();

// Set outo commit to false.

conn.setAutoCommit(false);

// Store Connection object in attribute of request.

MyUtils.storeConnection(request, conn);

// Allow request to go forward

// (Go to the next filter or target)

chain.doFilter(request, response);

// Invoke the commit() method to complete the transaction with the DB.

conn.commit();

} catch (Exception e) {

e.printStackTrace();

ConnectionUtils.rollbackQuietly(conn);

throw new ServletException();

} finally {

ConnectionUtils.closeQuietly(conn);

}

}

// With commons requests (images, css, html, ..)

// No need to open the connection.

else {

// Allow request to go forward

// (Go to the next filter or target)

chain.doFilter(request, response);

}

}

}