Q63 - Asynchronous Processing in Servlets [8]

Considering the following Asynchronous Servlets code, choose which statements are true after a GET request to the NullServlet is made:

com.nullhaus.NullServlet

```
package com.nullhaus;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;

@WebServlet(urlPatterns = "/foo/*", name="NullServlet", asyncSupported=true)
public class NullServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {

        if (req.getDispatcherType() == DispatcherType.REQUEST) {
            req.getRequestDispatcher("/baz").forward(req, resp);
        } else if (req.getDispatcherType() == DispatcherType.ASYNC) {
            String hello = (String)req.getAttribute("Hello");

            resp.getWriter().println("Phew, that was a ride!");
            resp.getWriter().println("Value of Hello is: " + hello);
        }
    }
}
```

com.nullhaus.NullServlet2

```
package com.nullhaus;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;

@WebServlet(urlPatterns = "/baz/*", name="NullServlet2", asyncSupported=true)
public class NullServlet2 extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {

        req.setAttribute("Hello", "World");

        final AsyncContext ac = req.startAsync();
        ac.dispatch();
    }
}
```

a. This code compiles

b. The "Value of Hello is: null" will be included in the response

c. The "Value of Hello is: World" will be included in the response

d. The infinite loop dispatch-loop will be created

e. The request will be served fine, but no text will be included in the response

f. A runtime exception will be thrown when accessing this servlet

g. This code doesn't compile

Hide answer

a, c

Reference: page 15, 2.3.3.3 "Asynchronous processing"

Explanation: This code is an example of how to use the `ServletRequest#getDispatcherType()` to recognize if the request was made initialy (`REQUEST`) or through asynchronous dispatch (`ASYNC`).

Another thing is that if the **unwrapped** `ServletRequest` and `ServletResponse` are used in the `ServletRequest#startAsync(-)`, the following `AsyncContext#dispatch()` will dispatch the to the URL of the **original** request (/foo/*).

The last thing shown is that the `ServletRequest` and the `ServletResponse` are the same objects in the dispatch chain, so the attributes are saved properly.