← Previous Q62 - Asynchronous Processing in Servlets [7] Next →

62 / 67

Q62 - Asynchronous Processing in Servlets [7]

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

```
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 {

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

        final AsyncContext ac = req.startAsync();
        ac.setTimeout(-2);

        ac.start(new Runnable() {
            public void run() {
                String att = (String)ac.getRequest().getAttribute("Hello");

                try {
                    PrintWriter pw = ac.getResponse().getWriter();
                    pw.println("Async! Value of Hello is: " + att);
                } catch (IOException e) {
                    e.printStackTrace();
                }

                ac.complete();
            }
        });
    }
}
```

a. This code compiles

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

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

d. The asynchronous operation will be timed out after the server default timeout value

e. The asynchronous operation will be never timed out

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

g. This code doesn't compile

Hide answer

a, c, e

Reference: page 14, 2.3.3.3 "Asynchronous processing"

Explanation: If the `AsyncContext#setTimeout(-)` argument is **<= 0**, it means that the asynchronous operation will **never be timed out**.

The `ServletRequest#startAsync()` method uses the unwrapped `ServletRequest` and `ServletResponse`, so the attributes set before asynchronous thread start are **accessible** from within the thread.