← Previous

Q60 - Asynchronous Processing in Servlets [5]

Next →

60 / 67

Q60 - Asynchronous Processing in Servlets [5]

Considering the following Asynchronous Servlets code, choose which statements are true after a GET request to the NullServlet2 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 {

        resp.getWriter().println("Howdy from NullServlet1!");
    }
}

package com.nullhaus;

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

@WebServlet(urlPatterns = "/baz/*", name = "NullServle2", asyncSupported = false)
public class NullServlet2 extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {

        resp.getWriter().println("Howdy from NullServlet2!");
        req.getRequestDispatcher("/foo").forward(req, resp);
    }
```

}

a. This code compiles

b. The "Howdy from NullServlet2" will be included in the response

c. The "Howdy from NullServlet1" will be included in the response

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

e. This code doesn't compile

Hide answer

a, c

Reference: pages 11, 2.3.3.3 "Asynchronous processing"

Explanation: It is **illegal** to dispatch the request from the synchronous servlet to the asynchronous, but the exception throwing is **delayed to the moment of actually using the asynchronous nature of the servlet** - like `AsyncContext.startAsync(-)`. If the asynchronous servlet doesn't use any of the asynchronous features, it is legal to do such dispatch.

The "Howdy from NullServlet2" will **not** be included in the response, as the uncommitted output in **response buffer is cleared** during the forwarding.