← Previous Q57 - Asynchronous Processing in Servlets [2] Next →

57 / 67

Q57 - Asynchronous Processing in Servlets [2]

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

```
package com.nullhaus;

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) {
        System.out.println("I'm inside!");

        final AsyncContext ac = req.startAsync();

        ac.start(new Runnable() {
            public void run() {
                System.out.println("*** I'm an async thread!");
                ac.complete();
            }
        });

        System.out.println("I'm leaving! Bye!");
    }
}
```

a. This code compiles and runs fine

b. There is no "asyncSupported" attribute of @WebServlet

c. The modifier "final" in final AsyncContext ac = req.startAsync(); is not necessary and can be safely removed

d. The guaranteed order of texts printed in the console/log file is: I'm inside!, *** I'm an async thread!, I'm leaving! Bye!

e. There is no HttpServletRequest#startAsync() method - there is only a HttpServletRequest#startAsync(ServletRequest, ServletResponse) method

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

g. This code doesn't compile

Hide answer

a

Reference: pages 10 - 20, 2.3.3.3 "Asynchronous processing"

Explanation: This code runs fine - there is an `asyncSupported` attribute and in fact it's the **only asynchronous related attribute in the `@WebServlet` annotation**. Things like timeouts must be dealt programmatically by using the `AsyncContext`.

The **final modifier is necessary**, because the "ac" variable must be accessible from the anonymous inner class.

The order of messages is **not guaranteed**. There are two threads, and the Container/JVM can choose how and when execute them; the only guaranteed thing is that the "I'm inside" message will be printed first.