← Previous Q50 - Servlet Security [2] Next →

50 / 67

Q50 - Servlet Security [2]

Consider the following Servlet code:

```
package com.nullhaus;

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

@HttpConstraint(EmptyRoleSemantic.DENY)
@WebServlet(value = "/foo/*", name = "NullServlet")
public class NullServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {
        resp.getWriter().print("Howdy Stragers!");
    }
}
```

Choose statements which are true about the GET HTTP request:

a. This servlet is accessible for all users

b. This servlet is not accessible for any users

c. The `EmptyRoleSemantic.DENY` is not a valid `@HttpConstraint` main ("value") attribute value

d. A runtime exception will be thrown while trying to access the servlet

e. The above code doesn't compile

Hide answer

a

Explanation: This might seem like a valid Http constraint which denies access for all users, but in fact it is **inappropriate usage of** `@HttpConstraint` annotation. The compiler wont' complain, a runtime exception will not be thrown, but the servlet will act like **there are no security constraints defined**. This is because the `@HttpConstraint` and `@HttpMethodConstraint` can be used only as **the** `@ServletConstraint` **annotation attributes**.