← Previous Q51 - Servlet Security [3] Next →

51 / 67

Q51 - Servlet Security [3]

Consider the following Servlet code:

```
package com.nullhaus;

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

@ServletSecurity(value = @HttpConstraint(EmptyRoleSemantic.DENY),
        httpMethodConstraints = {@HttpMethodConstraint(methodName = "GET",  emptyRoleSemantic = EmptyRoleSemantic.ALLOW)})

@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 made to the NullServlet:

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` attribute value

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

e. The above code doesn't compile

Hide answer

e

Explanation: There is **no** `methodName` attribute for `@HttpMethodConstraint` annotation. The value should be used instead.

There is **no** `EmptyRoleSemantic.ALLOW` enum value. The `PERMIT` value should be used instead.