BILBOKO INGENIARITZA ESKOLA

OHIKO AZTERKETA - 2019/06/24

IKASGAIA: Web Sistemak

GRADUA: Kudeaketaren eta Informazio Sistemen Ingeniaritza

Izena eta Abizenak:

Oharrak:

- *Azterketaren iraupena: 60 min.*
- *Galdera bakoitzak puntu bana balio du.*
- *Test galdera batzuk erantzun zuzen bat baino gehiago dute. Galdera zuzentzat hartua izan daiten, erantzun zuzen guztiak bakarrik markatu behar dira.*
- *Azterketa honek ikasgaiaren azkenengo kalifikazioaren %60 suposatzen du. Ikagaiaren irakaskuntza gidan adierazten denaren arabera, "Ikasgaia gainditzeko, azterketa idatzia gainditu behar da".*

1. Un servidor Web recibe la siguiente solicitud :

```
GET /html/Sistemas Web.html HTTP/1.1
Host: sw2016.com
Accept: text/html
Accept-Encoding: gzip,identity;q=0.5
Accept-Language: en-US,es-ES;q=0.8
User-Agent: Mozilla Windows Escritorio
```

¿Cual será la respuesta?

a. 200 - OK, la página HTML se enviará en el cuerpo del mensaje.
b. 400 - Bad Request,solicitud incorrecta.
c. 501 - Not Implemented, el servidor no conoce el método GET.
d. 404 - Not Found, , el servidor no encontró la página.
e. 405 - Method Not Allowed, el servidor no implementa el método GET.

2. De las siguientes afirmaciones, marca las correctas:

a. **HTTP es un protocolo sin estado. Para mantener la relación entre las solicitudes, el cliente web define las cookies y las envía al servidor..**

b. Una vez definida la cookie en el cliente, cada solicitud que hace el cliente al servidor envía información almacenada en la cookie, y el servidor utiliza esta información para identificar al cliente..

c. **Las cookies resuelven el problema del protocolo HTTP: es un protocolo de última generación, lo que significa que no tiene forma de mantener información permanente entre diferentes solicitudes..**

d. Las cookies son un tipo de software espía que puede leer la información personal almacenada en las computadoras de los usuarios.

e. **Las cookies son solo datos (no código), por lo que no pueden leer ni eliminar información de las computadoras de los usuarios.**

3. La siguiente aplicación de Python solicita a la API de Gmail que obtenga un mensaje no leído de SW2019@gmail.com:

```
cabeceras['User-Agent'] = 'Python Client'
cabeceras['Authorization'] = 'Bearer ' + access_token
params = { 'q': 'from:SW2019@gmail.com is:unread'}
uri='https://www.googleapis.com/gmail/v1/users/me/messages?'+params
respuesta = requests.get(uri, headers=cabeceras)
print respuesta.request
```

Ante la solicitud se obtiene la siguiente respuesta:

```
Response status: 403 Forbidden
{error: {code: 403, message: Invalid userid specified in
request/Delegation denied'}}
```

¿Dónde está el problema?

a. Escribimos mal el nombre del remitente en el programa.

b. Olvidamos incluir el encabezado Content_Type.

c. No pasamos los parámetros a la función request.get().

d. **En los parámetros, entre otros, no codificamos el carácter @.**

e. No tenemos mensajes sin leer.

4. Un servlet…

a. **Puede responder a cualquier solicitud, pero generalmente se usa en servidores web que reciben, procesan y responden a solicitudes HTTP..**

b. Solo puede responder a solicitudes GET y POST.

c. **HttpServlet es un objeto Java que extiende la clase.**

d. No se pueden leer los datos enviados a través de un formulario; esto es manejado por el contenedor de servlet (TomCat).

5. En la aplicación miApp, desea asignar el siguiente URI a un nuevo servlet implementado en la clase miSitio.NewServlet:

http://mydomain.com/miApp/DoLogin/NewServlet

Por favor complete el siguiente archivo de expansión:

```
<web-app>
<display-name> miApp</display-name>
    <servlet>
       <servlet-name>____[Nombre del servlet]__</servlet-name>
       <servlet-class>____miSitio.NewServlet_____</servlet-class>
    </servlet>
    <servlet-mapping>
       <servlet-name>_____[Nombre del servlet]__</servlet>
       <url-pattern>_____/DoLogin/NewServlet____</url-pattern>
    <servlet-mapping>
</web-app>
```

6. **Objeto(s) HttpServletResponse**…

a. Le permite convertir un documento XML en una solicitud HTTP.

b. **Utiliza el método getWriter () para traducir datos a formato de texto.**

c. Le permite realizar solicitudes HTTP que contienen el contenido de la respuesta en formato XML.

d. Los métodos doGet () y doPost () de Tomcat se utilizan para hacer referencia a solicitudes HTTP.

e. **Los métodos doGet () y doPost () de Tomcat se utilizan para hacer referencia a las respuestas HTTP.**

7. En OAuth 2.0, ¿quién identifica las variables consumer_key y consumer_secret?

a. usuario de la aplicación.
b. Instancia del servidor de aplicaciones.
c. programador
d. **Aplicacion.**
e. Ninguna es correcta.

8. Cree una aplicación de Google App Engine que solo tenga acceso a la API de Google Calendar... (elija la respuesta más adecuada)

a. Debe crear un proyecto de Google Cloud y registrar la aplicación con Google Calendar.
b. Callback_uri o redirect_uri deben estar definidos en el proyecto de Google Cloud.
c. La aplicación solo necesita estar registrada en Google Calendar.
d. **Cree un proyecto en Google Cloud para iniciar App Engine y crear credenciales para la aplicación.**
e. Crea un proyecto de Google Cloud y no necesitas crear credenciales para la aplicación porque perteneces a Google.

9. Escriba una solicitud a Twitter cuando se esté ejecutando una instancia de la siguiente clase PublishTweetHandler.

```
class PublishTweetHandler(BaseHandler):
    def get(self):

        oauth_token = self.session['oauth_token']
        oauth_token_secret = self.session['oauth_token_secret']
        oauth_headers = {'oauth_token': oauth_token}
        autorizacion= createAuthHeader(...)}

        method = 'POST'
        base_url = 'https://api.twitter.com/1.1/statuses/update.json'
        status=’ PruebaJunio2019’
        params = {'status': status}
        cabeceras = {'User-Agent': 'Google App Engine',
                     'Content-Type': 'application/x-www-form-urlencoded',
                     'Authorization': autorizacion}

        respuesta = requests.post(base_url, headers=cabeceras, data=params)

        logging.info(respuesta.text)
```

Nota: El valor del encabezado 'Autorización' se calcula con la función createAuthHeader (), y su valor es autorización

Responder:

POST /1.1/statuses/update.json HTTP / 1.1 Host: api.twitter.com
Agente de usuario: Google App Engine
Tipo de contenido: aplicación / x-www-form-urlencoded Longitud del contenido: 22
Autorización: estado <autorización> = PruebaJunio2019