1.

You want to post data to a server using java.net's HttpURLConnection. How would you accomplish this?

* Incorrect -

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

HttpPost post = new HttpPost(dataToPost);

conn.execute(post);

* Your choice: correct -

HttpURLConnection conn = (HttpURLConnection) url.openConnection(); conn.setRequestMethod("POST"); conn.setDoOutput(true); DataOutputStream stream = new DataOutputStream(conn.getOutputStream()); ... write data to post to stream ... stream.flush(); stream.close();

* Incorrect -

HttpURLConnection conn = (HttpURLConnection) url.openConnection(); conn.setRequestMethod("POST"); conn.performPost(dataToPost); conn.close();

* Incorrect -

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setRequestMethod(RequestMethod.POST);

conn.setDoOutput(true);

DataOutputStream stream = new DataOutputStream();

conn.setOutputStream(stream);

... write to stream ...

conn.transmitAndClose();

* Incorrect -

I don't know yet.

2.

What would you do to perform a post to a server with the Apache HttpClient library?

* Your choice: correct -

Construct an instance of HttpPost and pass it to the execute method of your HttpClient instance.

* Incorrect -

Construct an instance of HttpPost and call the execute method on it, passing your HttpClient instance as a parameter.

* Incorrect -

Construct an instance of HttpPost and call the post method on that instance.

* Incorrect -

Construct an instance of the HttpMethod class and call the post method on that instance.

* Incorrect -

I don't know yet.

3.

How can you send a request to a server using the HTTP/2 protocol with the HttpClient API introduced in Java 9?

* Incorrect -

Obtain an HttpClient instance from HttpClient.Builder by passing "HTTP/2" in as the protocol parameter to the newBuilder method. Use that client to send the request.

* Your choice: correct -

Create an HttpRequest and execute it using the HttpClient instance. If the server supports HTTP/2, HttpClient will use HTTP/2 behind the scenes.

* Incorrect -

Create an Http2Request and execute it using an HttpClient instance.

* Incorrect -

The HttpClient API introduced with Java 9 does not support HTTP/2.

* Incorrect -

I don't know yet.

4.

A servlet is annotated with:

@WebServlet({"/MyServlet", ""})

Assuming the servlet is deployed to localhost:8080, which URL(s) will request the above server?

* Incorrect -

http://localhost:8080/MyServlet

http://localhost:8080

* Incorrect -

http://localhost:8080/MyServlet

* Incorrect -

http://localhost:8080/MyServlet

http://localhost:8080/

http://localhost:8080

* Your choice: correct -

http://localhost:8080/MyServlet

http://localhost:8080/

* Incorrect -

I don't know yet.

5.

Given:

Cookie cookie = new Cookie("id","1"); cookie.setMaxAge(0); response.addCookie(cookie);

Which of the following statements about the creation of this cookie is TRUE?

* Your choice: correct -

The snippet deletes the cookie.

* Incorrect -

The snippet creates a cookie that will be deleted when the browser exits.

* Incorrect -

The snippet creates a cookie that will never be deleted.

* Incorrect -

The snippet creates a cookie that will not be included in the response.

* Incorrect -

I don't know yet.

6.

What's the main difference between the classes javax.net.ssl.SSLEngine and javax.net.ssl.SSLSocket?

* Your choice: incorrect -

If the handshake fails for any reason, the SSLSocket is closed, and no further communications can be done. With SSLEngine, the failure can be caught as an SSLException or an inbound closure handshake message, and the handshake can be retried.

* Incorrect -

When an SSLEngine is first created, no handshaking is done so that applications may first set their communication preferences, for example, what cipher suites to use. The first step for an SSLSocket is to establish the initial handshake.

* Correct -

SSLEngine operates on inbound and outbound byte streams, independent of the transport mechanism and non-blocking. With SSLSocket, all of the inbound and outbound data is automatically transported using the underlying Socket, which is blocking.

* Incorrect -

An SSLSocket must choose to operate in the client or server mode. An SSLEngine can only operate in client mode.

* Incorrect -

I don't know yet.

7.

How would you set the User-Agent header using the Apache HttpCore library?

* Incorrect -

Add a RequestContent interceptor to the protocol processor.

* Correct -

Add a RequestUserAgent interceptor to the protocol processor.

* Incorrect -

Call the setUserAgent method on the HttpRequest object.

* Incorrect -

Call the setUserAgent method on the HttpClient instance.

* Incorrect -

I don't know yet.

8.

Which type of Java objects can be encoded into a WebSocket binary message using an encoder?

* Incorrect -

Objects that implement the java.io.Serializable interface

* Incorrect -

Objects of type java.nio.ByteBuffer

* Correct -

Any Java object

* Incorrect -

Byte arrays

* Incorrect -

I don't know yet.

9.

You have a JAX-RS application set up with annotations, the web.xml file is empty. It is deployed in a Glassfish server. However, when you deploy this application in a Tomcat server, the application doesn't start, it throws the following exception:

java.lang.ClassCastException: com.example.rest.MyApplication cannot be cast to javax.servlet.Servlet

com.example.rest.MyApplication is the class annotated with javax.ws.rs.ApplicationPath.

Which of the following is the most likely problem?

* Incorrect -

Tomcat cannot work with an empty web.xml file. You need to add a dummy servlet.

* Your choice: correct -

Tomcat is not a full Java EE server, it's just a servlet container. You need to include a REST implementation as a dependency.

* Incorrect -

JAX-RS applications only work with servers like Glassfish or Wildfly that fully support the JAX-RS specification. The application cannot work with Tomcat.

* Incorrect -

You need to add the Servlet API as a runtime dependency.

* Incorrect -

I don't know yet.

10.

Which of the following headers can be used to push a file using the server push feature of HTTP/2?

* Your choice: correct -

Link

* Incorrect -

After-Html

* Incorrect -

Prefetch

* Incorrect -

Preload

* Incorrect -

I don't know yet.

11.

What happens when the below method is executed?

import javax.ws.rs.container.AsyncResponse; import javax.ws.rs.container.Suspended; @Path("/products") public class ProductService { @PUT @Consumes("application/json") public void put(final Order order, final @Suspended AsyncResponse response) { } }

* Incorrect -

An empty response is immediately sent to the client.

* Incorrect -

An error is immediately sent to the client.

* Incorrect -

A successful response is immediately sent to the client.

* Your choice: correct -

A response is never sent to the client, it just times out.

* Incorrect -

I don't know yet.

12.

Using the programmatic version of the Java WebSocket API, which of the following options correctly processes a text message?

* Your choice: correct -

public class Server extends Endpoint {

@Override

public void onOpen(Session session, EndpointConfig config) {

session.addMessageHandler(new MessageHandler.Whole<String>() {

@Override

public void onMessage(String message) {

// Process message

}

});

}

// ...

}

* Incorrect -

public class Server extends Endpoint {

@Override

public void onMessage(String message) {

// Process message

}

// ...

}

* Incorrect -

public class Server extends Endpoint {

@Override

public void onMessage(Session session, EndpointConfig config) {

session.addMessageHandler(new MessageHandler.Whole<String>() {

@Override

public void onHandled(String message) {

// Process message

}

});

}

// ...

}

* Incorrect -

public class Server extends Endpoint {

@Override

public void onConfiguration(Session session, EndpointConfig config) {

session.addTextMessageHandler(new TextMessageHandler() {

@Override

public void onMessage(String message) {

// Process message

}

});

}

// ...

}

* Incorrect -

I don't know yet.

13.

You coded a class that will act as a server-side WebSocket endpoint:

public class ChatServer {

// Endpoint code...

}

What would you do to configure this endpoint with the path/chat and ensure that it only supports the superchat WebSocket protocol?

* Correct -

Annotate the class with:

@ServerEndpdoint(

value = "/chat",

subprotocols = {"superchat"}

)

* Incorrect -

Annotate the class with:

@ServerEnpdoint(

value = "/chat",

configurator = SuperChatConfigurator.class

)

Where SuperChatConfigurator configures the endpoint so it can only support the superchat WebSocket protocol.

* Your choice: incorrect -

Configure the endpoint with:

ServerEndpointConfig serverConfig = ServerEndpointConfig.Builder

.create(ChatServer.class , "/chat")

.configurator(new SuperChatConfigurator())

.build();

Where SuperChatConfigurator configures the endpoint so it can only support the superchat WebSocket protocol.

* Incorrect -

Configure the endpoint with:

ServerEndpointConfig serverConfig = ServerEndpointConfig.Builder

.create(ChatServer.class , "/chat")

.subprotocols(Arrays.asList("superchat"))

.build();

* Incorrect -

I don't know yet.

14.

Which Apache HttpCore library provides low-level components for asynchronous messaging?

* Your choice: correct -

HttpCore NIO

* Incorrect -

HttpClient

* Incorrect -

HttpCore Blocking

* Incorrect -

HttpAsyncClient

* Incorrect -

I don't know yet.

15.

How can you execute different types of HTTP methods with Apache's HttpClient?

* Your choice: correct -

Construct an instance of a different subclass of HttpRequestBase for a given HTTP method.

* Incorrect -

Construct an instance of a different subclass of HttpMethodType for a given HTTP method.

* Incorrect -

Construct an instance of an HttpClient passing a String parameter indicating the HTTP method to be executed.

* Incorrect -

Call different methods on an instance of an HttpClient based on a given HTTP method.

* Incorrect -

I don't know yet.

16.

What does the following code snippet do?

import java.time.Instant; import java.io.File; import java.nio.file.Files; import java.nio.file.Path import java.nio.file.attribute.BasicFileAttributeView; import java.nio.file.attribute.FileTime; //... FileTime time = FileTime.from(Instant.now()); Path path = new File("1.txt").toPath(); Files.getFileAttributeView( path, BasicFileAttributeView.class ).setTimes(time, null, null);

* Incorrect -

It accesses the file attributes of the file 1.txt and sets the time the attributes were last accessed.

* Incorrect -

It creates the file 1.txt and sets its time attributes (lastModifiedTime, lastAccessTime, and creationTime) to the current time.

* Incorrect -

It creates the file 1.txt and sets its creation time attribute to the current time.

* Your choice: correct -

It changes the last modified time attribute of the file 1.txt to the current time.

* Incorrect -

I don't know yet.

17.

What are the three types of proxies supported by java.net's Proxy.Type enumeration?

* Incorrect -

PASS\_THROUGH, REDIRECT, and CACHING

* Incorrect -

HTTP, HTTPS, and SOCKET

* Your choice: correct -

DIRECT, HTTP, and SOCKS

* Incorrect -

SSL, FTP, and HTTPS

* Incorrect -

I don't know yet.

18.

Given:

Path path = Paths.get("c:\\temp\\deleted");

Which of the following will return the 'temp' segment of the path?

* Incorrect -

path.getRoot()

* Incorrect -

path.getName(1)

* Incorrect -

path.subpath(0, 0)

* Your choice: correct -

path.getName(0)

* Incorrect -

I don't know yet.

1. Which of the following options represent a class that fetches an init parameter, sets it in the servlet context at initialization, and compiles successfully?

Incorrect -

@WebListener

public class MyListener implements ServletContextAttributeListener {

        @Override

        public void contextInitialized(ServletContextEvent sce) {

                ServletContext sc = sce.getServletContext();

                String initParameter = sc.getInitParameter("myInitParameter");

                sc.setAttribute("myContextParameter", initParameter);

        }

}

Incorrect -

@WebListener

public class MyListener implements ServletContextListener {

        @Override

        public ServletContextEvent contextInitialized(ServletContextEvent sce) {

                ServletContext sc = sce.getServletContext();

                String initParameter = sc.getInitParameter("myInitParameter");

                sc.setAttribute("myContextParameter", initParameter);

        return sce;

        }

        @Override

        public ServletContextEvent contextDestroyed(ServletContextEvent sce) { return sce; }

}

Correct -

@WebListener

public class MyListener implements ServletContextListener {

        @Override

        public void contextInitialized(ServletContextEvent sce) {

                ServletContext sc = sce.getServletContext();

                String initParameter = sc.getInitParameter("myInitParameter");

                sc.setAttribute("myContextParameter", initParameter);

        }

        @Override

        public void contextDestroyed(ServletContextEvent sce) { }

}

Your choice: incorrect -

@WebListener

public class MyListener implements ServletContextAttributeListener {

        @Override

        public ServletContextEvent contextInitialized(ServletContextEvent sce) {

                ServletContext sc = sce.getServletContext();

                String initParameter = sc.getInitParameter("myInitParameter");

                sc.setAttribute("myContextParameter", initParameter);

        return sce;

        }

        @Override

        public ServletContextEvent contextDestroyed(ServletContextEvent sce) { return sce; }

}

Incorrect -I don't know yet.

2. Given:

public class TestServlet extends HttpServlet {

        protected void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

                response.setContentType("text/html");

                PrintWriter out = response.getWriter();

                out.println("<h1>Hello!</h1>");

                String action = request.getParameter("action");

                if (action != null) {

                        RequestDispatcher rd = request.getRequestDispatcher("index.jsp");

                        if ("include".equalsIgnoreCase(action)) {

                                rd.include(request, response);

                        } else if ("forward".equalsIgnoreCase(action)) {

                                rd.forward(request, response);

                        }

                }

        }

}

And index.jsp has the following content:

<h1>John!</h1>

Which of the following statements is TRUE?

Incorrect -If the servlet is executed with "action=forward" as an URL parameter, <h1>Hello!</h1> will be rendered.

Incorrect -If the servlet is executed with "action=include" as an URL parameter,<h1>John!</h1> will be rendered.

Incorrect -If the servlet is executed with "action=forward" as an URL parameter, <h1>Hello!</h1><h1>John!</h1> will be rendered.

Your choice: correct -If the servlet is executed with "action=include" as an URL parameter, <h1>Hello!</h1><h1>John!</h1> will be rendered.

Incorrect -I don't know yet.

3. What does the web container do when it wants to remove a servlet?

Incorrect -The server calls the finalize method of the servlet immediately, aborting all service requests.

Your choice: correct -The server calls the destroy method of the servlet but only after all service requests have returned or after a timeout period, whichever comes first.

Incorrect -The server calls the destroy method of the servlet immediately, aborting all service requests.

Incorrect -The server calls the finalize method of the servlet but only after all service requests have returned or after a timeout period, whichever comes first.

Incorrect -I don't know yet.

4. How would you read data from a java.net HttpURLConnection?

Incorrect -Call execute on the connection, and read the data from the returned stream.

Incorrect -Call the getResponse method on the connection.

Correct -Open the connection if needed, then read from the InputStream of the connection.

Your choice: incorrect -Open the connection if needed, then call the read method on the connection.

Incorrect -I don't know yet.

5. All the methods of the javax.ws.rs.client.AsyncInvoker interface return an instance of:

Correct -java.util.concurrent.Future

Incorrect -javax.ws.rs.client.Invocation

Incorrect -java.util.concurrent.Callable

Your choice: incorrect -javax.ws.rs.client.WebTarget

Incorrect -I don't know yet.

6. What data types can be used to send messages from client to server via SOAP?

Your choice: incorrect -HTML, XML, or JSON

Incorrect -Only JSON

Incorrect -JSON or XML

Correct -Only XML

Incorrect -I don't know yet.

7. Which HTTP request method submits an entity to a server and is not idempotent, i.e. may have additional effects each time it is called?

Your choice: correct -POST

Incorrect -PATCH

Incorrect -GET

Incorrect -PUT

Incorrect -I don't know yet.

8. An init parameter is passed to a servlet:

<servlet>

  <servlet-name>MyServlet</servlet-name>

  <servlet-class>com.example.MyServlet</servlet-class>

  <init-param>

    <param-name>myParam</param-name>

    <param-value>paramValue</param-value>

  </init-param>

</servlet>

Which servlet method would you use to receive this parameter?

Your choice: incorrect -doGet(HttpServletRequest request, HttpServletResponse response)

Incorrect -service(ServletRequest request, ServletResponse response)

Incorrect -destroy()

Correct -init(ServletConfig servletConfig)

Incorrect -I don't know yet.

9. You need to store the identifier of a WebSocket session when it is opened. Which of the following methods gets the session identifier correctly?

Incorrect -

@OnOpen

public void onOpen (@PathParam String sessionId) {

   // Store the session id ...

}

Incorrect -

@OnOpen

public void onOpen (EndpointConfig config) {

   String id = config.getSessionId();

   // Store the session id ...

}

Incorrect -

@OnOpen

public void onOpen (SessionConfig config) {

   String id = config.getId();

   // Store the session id ...

}

Your choice: correct -

@OnOpen

public void onOpen (Session session) {

   String id = session.getId();

   // Store the session id ...

}

Incorrect -I don't know yet.

10. What is the default HTTP protocol version for the HttpClient API introduced in Java 9?

Incorrect -HTTP/1.2

Correct -HTTP/2

Incorrect -HTTP/2.2

Your choice: incorrect -HTTP/1.0

Incorrect -I don't know yet.

11. What is the purpose of the Apache HttpClient CredentialsProvider interface?

Incorrect -It is a concrete cache of authentication information for a single authorization scope.

Incorrect -It is a username and password holder for requests requiring Basic authentication.

Incorrect -It is a type of response that may be returned from an HTTP method execution if authentication is required by the server.

Your choice: correct -It is an abstract credentials provider that holds and returns credentials for different authorization scopes.

Incorrect -I don't know yet.

12. What are the transmission protocol options most commonly used for SOAP and REST?

Incorrect -REST and SOAP both only function over HTTP.

Correct -REST is almost always used with HTTP, SOAP can function over multiple protocols including HTTP, SMTP, and TCP.

Incorrect -REST is primarily used over HTTP, whereas SOAP functions only over TCP.

Your choice: incorrect -REST is primarily used over HTTP or TCP, and SOAP is restricted to HTTP.

Incorrect -I don't know yet.

13. How can you set a response timeout of 3 seconds for a GET request using the HttpClient API introduced with Java 9?

Incorrect -

HttpRequest request = ... code to build request ...

request.execute(Duration.of(3, SECONDS);

Correct -

HttpRequest.newBuilder().GET().timeout(Duration.of(3, SECONDS).build();

Your choice: incorrect -

HttpClient.timeout(Duration.of(3, SECONDS);

HttpClient.newBuilder().... code to build request....

Incorrect -

HttpRequest request = ... code to build request...

request.timeout(Duration.of(3, SECONDS);

Incorrect -I don't know yet.

14. Which of the following methods from HttpServlet allows you to handle HTTP GET requests?

Your choice: incorrect -executeGet

Incorrect -request

Correct -doGet

Incorrect -forward

Incorrect -I don't know yet.

15. Annotations like javax.ws.rs.GET or javax.ws.rs.POST can be used:

Incorrect -On other annotations (they are meta-annotations)

Your choice: incorrect -On both classes and methods

Correct -On methods only

Incorrect -On classes only

Incorrect -I don't know yet.

16. How can you set headers for a request using the HttpClient API introduced with Java 9?

Incorrect -Pass the desired header(s) in to either the header or headers method on the HttpRequest.

Correct -Pass the desired header(s) in to the header or headers method on an HttpRequest.Builder.

Your choice: incorrect -Pass in a map of header names and values to the constructor of the HttpRequest.

Incorrect -Pass in a map of header names and values to the execute method on the HttpRequest.

Incorrect -I don't know yet.

17. What's the effect of creating a session in the following way: HttpSession session = request.getSession(true);

Incorrect -It forces the session management mechanism to use URL rewriting.

Incorrect -If a session object is not already created, it returns null.

Incorrect -It forces the session management mechanism to use cookies.

Your choice: correct -It creates a session object if it is not already created.

Incorrect -I don't know yet.

18. What is Application Layer Protocol Negotiation?

Incorrect -It is part of HTTP/1.0 which specifies how client and server negotiate the HTTP protocol version to use.

Incorrect -It is part of HTTP/1.0 which determines how to transmit the contents of a particular file based on its MIME type.

Correct -It is part of HTTP/2 which specifies how client and server negotiate the HTTP protocol version to use.

Your choice: incorrect -It is part of HTTP/2 which specifies whether or not multiplexing can occur for a specific connection.

Incorrect -I don't know yet.

19. What method do you call on a java.net HttpURLConnection to obtain a stream of data to read from the server?

Correct -

getInputStream

Incorrect -

read

Incorrect -

getData

Your choice: incorrect -

getResponseStream

Incorrect -

I don't know yet.

20.What would you do to ensure resources are closed when using Apache's HttpClient library after several HTTP methods finish executing against a server?

Incorrect -

Set a timeout on an instance of the CloseableHttpClient class that will cause all resources to close after a period of disuse.

Correct -

Use an instance of the CloseableHttpClient class and call the close method in a final block or include the instance in a try-with-resources statement.

Incorrect -

Call the evict method on the HttpClientBuilder class with your HttpClient instance as an argument.

Your choice: incorrect -

Use an instance of the CloseableHttpClient class that will allow the resources to be closed when the instance goes out of scope.

Incorrect -

I don't know yet.

21.How would you access the HTTP status code for a response using the HttpClient API introduced with Java 9?

Correct -

Call the statusCode method on an HttpResponse.

Incorrect -

Call the statusCode method on an HttpResponse.BodyProcessor.

Your choice: incorrect -

Get the integer return value of the send method on an HttpRequest.

Incorrect -

Get integer return value of the send method on HttpClient.

Incorrect -

I don't know yet.

22.Which class of HTTP status codes indicate a request was successfully received, understood, and accepted by a server?

Correct -2xx status codes

Incorrect -3xx status codes

Incorrect -4xx status codes

Your choice: incorrect -xOK status codes

Incorrect -I don't know yet.

23.What is the return type for the response returned from a synchronous Apache HttpClient request execution?

Incorrect -HttpResult

Your choice: incorrect -HttpEntity

Correct -HttpResponse

Incorrect -HttpOutputStream

Incorrect -I don't know yet.

24.Which class of HTTP status codes indicate a client-side error?

Your choice: incorrect -xxx-CLIENT status codes

Correct -4xx status codes

Incorrect -3xx status codes

Incorrect -2xx status codes

Incorrect -I don't know yet.

25.What is the purpose of the Stream Identifier in an HTTP/2 frame?

Incorrect -

It is used to sequence separate frames into their correct order.

Correct -

It is used to identify which stream a given frame belongs to, allowing the message to be reassembled.

Your choice: incorrect -

It is used to identify which client a given message is being sent from, allowing the response to go back to the correct client.

Incorrect -

It is used to establish the priority of one request incoming on a given connection vs another request on the same connection.

Incorrect -

I don't know yet.

26.Which Apache library assists with asynchronous HTTP 1.1 requests?

Your choice: incorrect -

HttpClient

Incorrect -

HttpCoreAsync

Incorrect -

HttpClientCore

Correct -

HttpAsyncClient

Incorrect -

I don't know yet.

27.Which interface do you have to implement in order to receive a notification when the servlet context has been initialized?

Correct -

javax.servlet.ServletContextListener

Incorrect -

javax.servlet.ServletContextAttributeListener

Your choice: incorrect -

javax.servlet.ServletContextInitializationListener

Incorrect -

javax.servlet.ServletListener

Incorrect -

I don't know yet.

28.What is the general purpose and role of Apache's HttpCore library?

Incorrect -

It is an alternative to the Java Servlet specification.

Correct -

It is a set of low-level components for building synchronous and asynchronous client and server HTTP services.

Your choice: incorrect -

It is a set of low-level components and APIs for building synchronous client and server HTTP services.

Incorrect -

It is a set of low-level components for building HTTP client-side applications.

Incorrect -

I don't know yet.

29.Which of the following is an example of one of the values defined in the enum javax.ws.rs.core.Response.Status?

Correct -

NO\_CONTENT

Incorrect -

COOKIE

Incorrect -

CONFIDENTIAL

Your choice: incorrect -

AUTHORIZED

Incorrect -

I don't know yet.

30.Which of the following options is an example of a valid WebSocket URI that a client can use?

Correct -

ws://example.com:1234/socket

Incorrect -

wsp://socket.example.com:5125

Incorrect -

<http://socket.example.com:3000>

Your choice: incorrect -

file://example.com:80/index.socket

Incorrect -

I don't know yet.

Incorrect -I don't know yet.

31.How do servlets handle multiple requests from clients?

* Your choice: incorrect -

Multiple instances of a servlet are created by one thread, then each instance handles a request.

* Incorrect -

A single instance of a servlet is created by one thread, then that thread handles multiple requests at the same time.

* Incorrect -

Multiple instances of a servlet are created by multiple threads, then each thread (and its corresponding instance) handles a request.

* Correct -

A single instance of a servlet is created, then that instance is shared across multiple threads, each handling a request.

* Incorrect -

I don't know yet.

32.What is the purpose of the RequestContent interceptor in the Apache HttpCore library?

* Incorrect -

It sets the Accept and User-Agent headers on outgoing requests from a client.

* Correct -

It sets the Content-Length or Transfer-Content headers based on the enclosed entity and protocol version.

* Your choice: incorrect -

It examines if the content of the request has been encrypted, and if not encrypts it.

* Incorrect -

It compresses the entity content of a request.

* Incorrect -

I don't know yet.

33.What are the basic HttpResponse.BodyProcessor types provided by the Java 9 HttpClient API?

* Incorrect -

ByteArray, String, Char[], and Object

* Your choice: correct -

ByteArray, String, File, and InputStream

* Incorrect -

Only the interface is provided. All implementations must be created by the requesting application.

* Incorrect -

String

* Incorrect -

I don't know yet.

34.A REST operation to create a company resource with the properties:

ID: 1 Name:Pluralsight It's defined in the following way: PUT /company/1 { "name": "Pluralsight" }

The first time a request with this information is received, the server creates a company. If the server receives a request with the same ID, it will update the company information.

Which of the following statements is TRUE?

* Incorrect -

Replacing PUT with PATCH and removing the ID (1) from the resource path will make this operation idempotent.

* Correct -

Since the operation specifies the ID, if the client issues the same request again, no new resource will be created, so the operation is idempotent.

* Incorrect -

Since the operation specifies the ID, if the client issues the same request again, no new resource will be created, so the operation is non-idempotent.

* Your choice: incorrect -

This operation is not syntactically correct, the PUT HTTP method cannot be used to create new resources in any case.

* Incorrect -

I don't know yet.

35.What HTTP methods are supported by a java.net HttpURLConnection?

* Your choice: correct -

GET, POST, PUT, DELETE, HEAD, TRACE, and OPTIONS

* Incorrect -

GET, POST, and PUT

* Incorrect -

GET, POST, PATCH, PUT, DELETE, and FETCH

* Incorrect -

Only GET and POST

* Incorrect -

I don't know yet.

36.You are implementing a web service that will be consumed exclusively by browser-based clients. Would SOAP or REST be more appropriate for the service?

* Incorrect -

Neither has differences that would make them easier or harder to consume by browser-based clients.

* Your choice: correct -

REST, since it supports JSON and no complex XML message structures to send messages need be built.

* Incorrect -

REST, since it can function using HTTP protocol message types, whereas SOAP requires socket-level connections.

* Incorrect -

SOAP, since it provides simple HTTP and XML or JSON based APIs requiring less code to be written on the client-side than REST.

* Incorrect -

I don't know yet.

37.Which class would you construct an instance of to get a default implementation for accepting and storing cookies for a java.net HttpURLConnection?

* Incorrect -

CookieStore

* Your choice: incorrect -

CookieHandler

* Correct -

CookieManager

* Incorrect -

CookiePolicy

* Incorrect -

I don't know yet.

38.Which of the following is a valid uniform resource identifier (URI) scheme defined in the WebSocket protocol specification?

* Incorrect -

wps

* Correct -

wss

* Incorrect -

wsc

* Your choice: incorrect -

wsp

* Incorrect -

I don't know yet.

39.What might you use an HttpAsyncRequestExecutor for in the Apache HttpCore NIO library?

* Incorrect -

Handling multiple responses from a client asynchronously

* Incorrect -

Executing multiple requests to a server in sequence

* Your choice: correct -

Making multiple asynchronous requests to a server

* Incorrect -

Handling multiple responses on a server in a sequence

* Incorrect -

I don't know yet.

40.Which of the following classes contains constants that represent types that the annotation javax.ws.rs.Consumes may use?

* Your choice: incorrect -

javax.ws.rs.core.ApplicationType

* Correct -

javax.ws.rs.core.MediaType

* Incorrect -

javax.ws.rs.core.MimeType

* Incorrect -

javax.ws.rs.core.ProducesType

* Incorrect -

I don't know yet.

41.Given:

WebSocketContainer.connectToServer( MyClient.class, URI.create("wss://example.com") );

Which of the following statements is TRUE about the above snippet?

* Your choice: incorrect -

This snippet must reside in a method annotated with @Endpoint.

* Correct -

MyClient.class must be a client endpoint created using the annotated WebSocket API.

* Incorrect -

MyClient.class must be a client endpoint created using the programmatic WebSocket API.

* Incorrect -

It will throw an exception at runtime, the URI is not formatted correctly.

* Incorrect -

I don't know yet.

42.What is the binary framing layer of HTTP/2?

* Incorrect -

It is how HTTP/2 transmits meta-data packets separate from payload packets for a given message to be passed over a connection.

* Incorrect -

It is how HTTP/2 handles multipart form submissions.

* Incorrect -

It is how HTTP/2 identifies which requests to direct to which connections in a connection pool.

* Your choice: correct -

It is how HTTP/2 encapsulates and transmits data over a connection via multiple binary frames.

* Incorrect -

I don't know yet.