Instructor Notes:

Add instructor notes here.

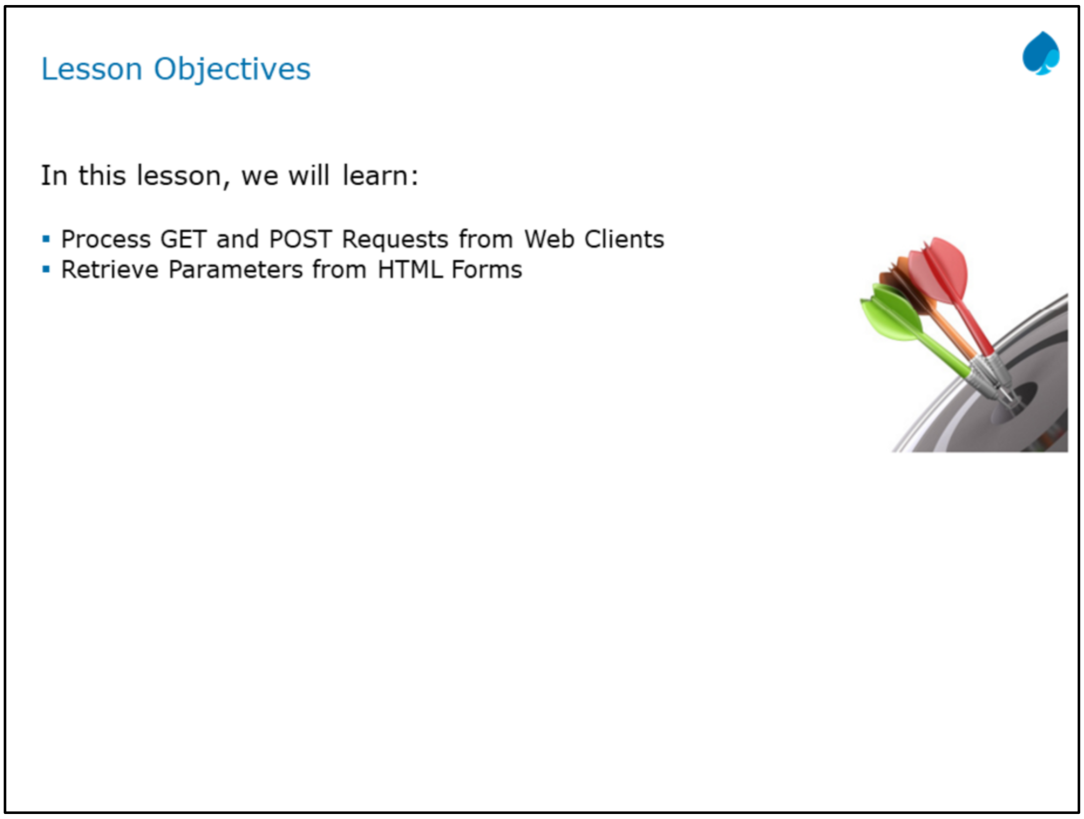
Java Servlets 3.0

Lesson 03: Request Object

Capgemini

Instructor Notes:

Explain the lesson coverage



Lesson Objectives

In this lesson, we will learn:

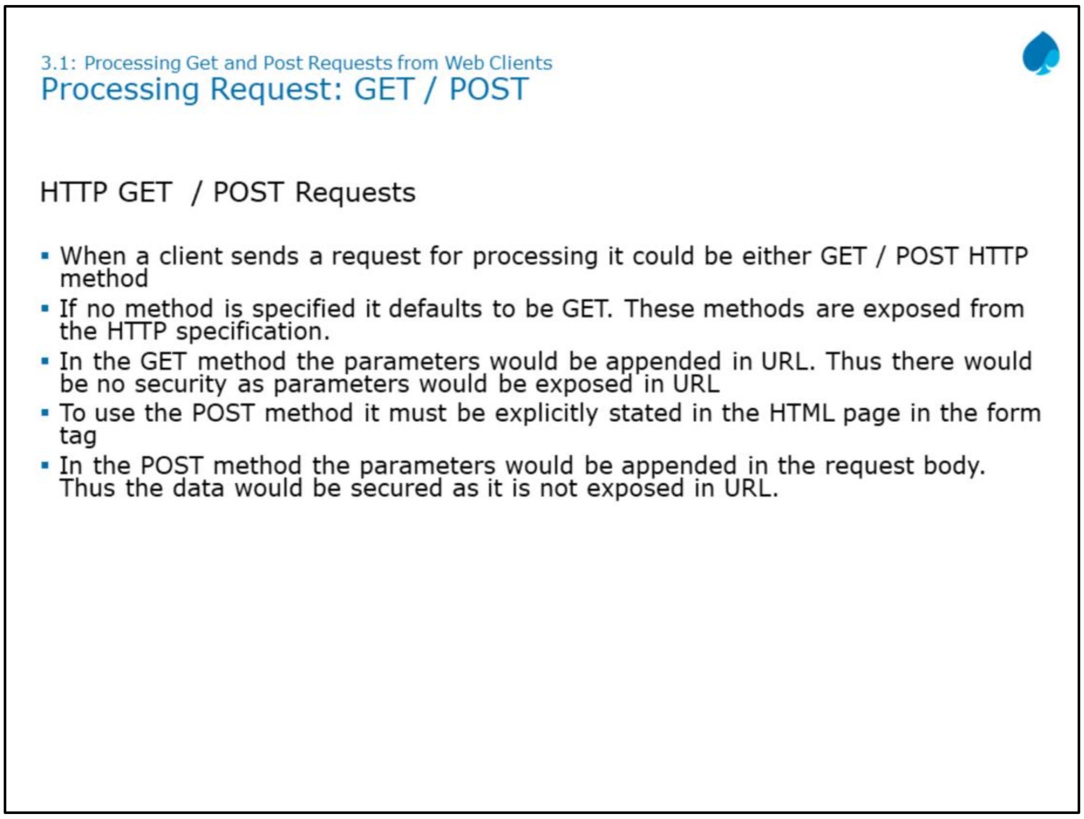
- Process GET and POST Requests from Web Clients
- Retrieve Parameters from HTML Forms

Lesson Objectives:
This lesson introduces the Servlet request object. The lesson contents are:
Lesson 04: Request Object

- 4.1: Processing Get and Post Requests from Web Clients
- 4.2: Retrieve Parameters from HTML Forms
- 4.3: Retrieving Path Information
- 4.4: Retrieve Request Headers

Instructor Notes:

Introducing GET / POST methods. Their usage and differences need to be explained here.



3.1: Processing Get and Post Requests from Web Clients

Processing Request: GET / POST

HTTP GET / POST Requests

- When a client sends a request for processing it could be either GET / POST HTTP method
- If no method is specified it defaults to be GET. These methods are exposed from the HTTP specification.
- In the GET method the parameters would be appended in URL. Thus there would be no security as parameters would be exposed in URL
- To use the POST method it must be explicitly stated in the HTML page in the form tag
- In the POST method the parameters would be appended in the request body. Thus the data would be secured as it is not exposed in URL.

Need to explain that 2 methods could be used in Servlet to process the request (GET / POST).

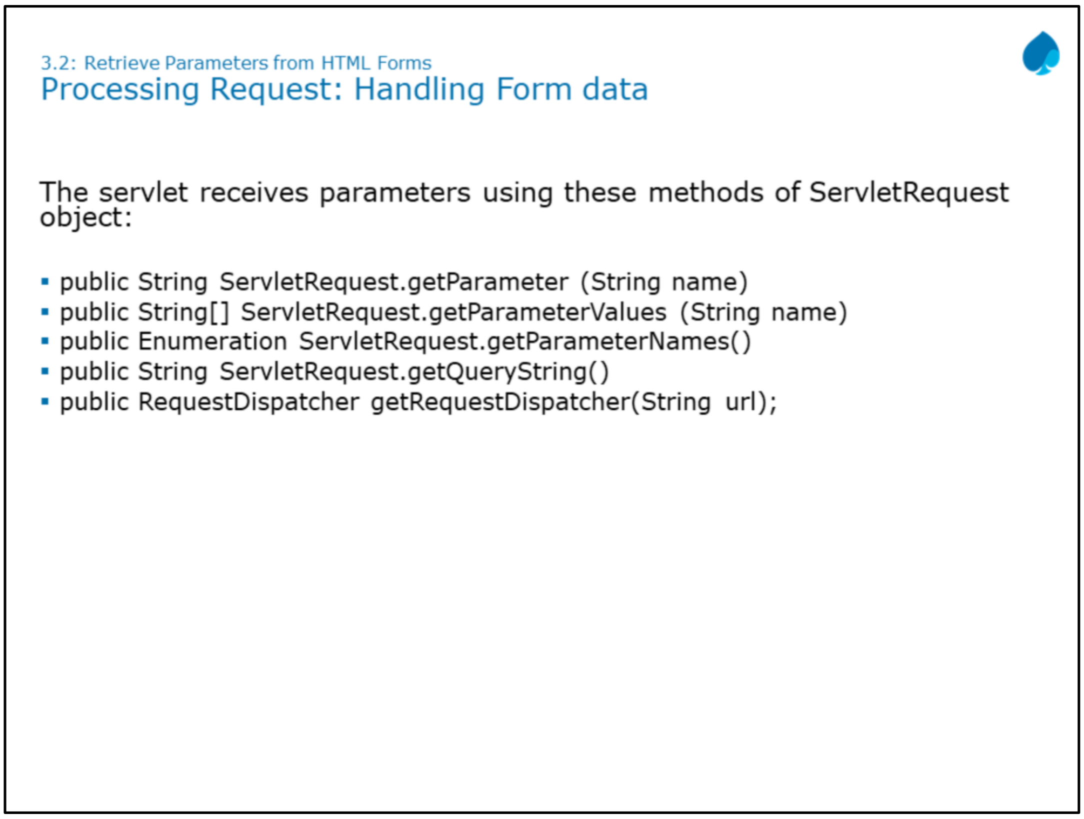
The methods are present in the HTTP specification which are exposed in Servlet API.

Differences between them being:
GET : not secured and POST: secured
GET: data gets appended in URL and POST: data is appended as a part of request body
GET: limitation of data to be appended and POST: unlimited data could be appended

Other methods of the Http Specification are:
PUT : Replaces all current representations of the target resource with the uploaded content
HEAD : Same as GET, but transfers the status line and header section only.
DELETE : Removes all current representations of the target resource given by a URI.
TRACE : Performs a message loop-back test along the path to the target resource
OPTIONS : Describes the communication options for the target resource
CONNECT : Establishes a tunnel to the server identified by a given URI

Instructor Notes:

We shall be seeing each of these methods in our demo coming next



3.2: Retrieve Parameters from HTML Forms

Processing Request: Handling Form data

The servlet receives parameters using these methods of ServletRequest object:

- public String ServletRequest.getParameter (String name)
- public String[] ServletRequest.getParameterValues (String name)
- public Enumeration ServletRequest.getParameterNames()
- public String ServletRequest.getQueryString()
- public RequestDispatcher getRequestDispatcher(String url);

Processing Get and Post Requests from Web Clients:

Request – Information / data / parameters sent from the client (browser) to Server for processing and formulating a response.
When a servlet accepts a request from a client (service(Request , Response) method), the container creates and hands over two objects, ServletRequest and ServletResponse. The ServletRequest object encapsulates the communication from the client to the server, while the ServletResponse object encapsulates the communication from the server back to the client.

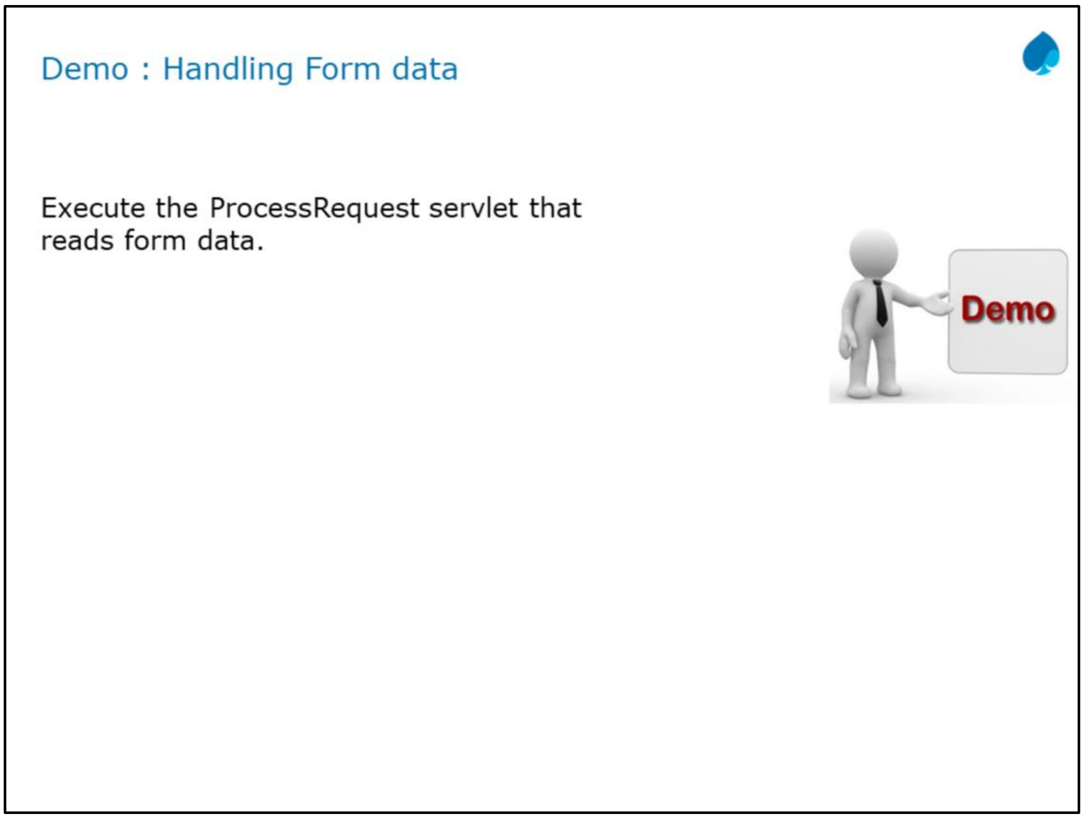
Since we are using HTTP protocol, the container would handover HttpServletRequest and HttpServletResponse objects to the service() method of HttpServlet class. Thias is done by downcasting ServletRequest to HttpServletRequest and ServletResponse to HttpServletResponse.

Request parameters:
The ServletRequest (or HttpServletRequest object if using HttpServlet) contains a lot of information. We shall see how servlet processes form parameters.

public String ServletRequest.getParameter (String name): This returns the value of the named parameter as a String or null if parameter wasn't specified.
public String[] ServletRequest.getParameterValues (String name): It returns values of named parameter as a String array if parameter could have multiple values, as in the case of list element with multiple select.
public Enumeration ServletRequest.getParameterNames(): It returns a list of parameter names.
public String ServletRequest.getQueryString(): It returns the raw query string of the request.

Instructor Notes:

Run the ProcessRequest Servlet and userInput.html to show how to retrieve form parameters



Demo : Handling Form data

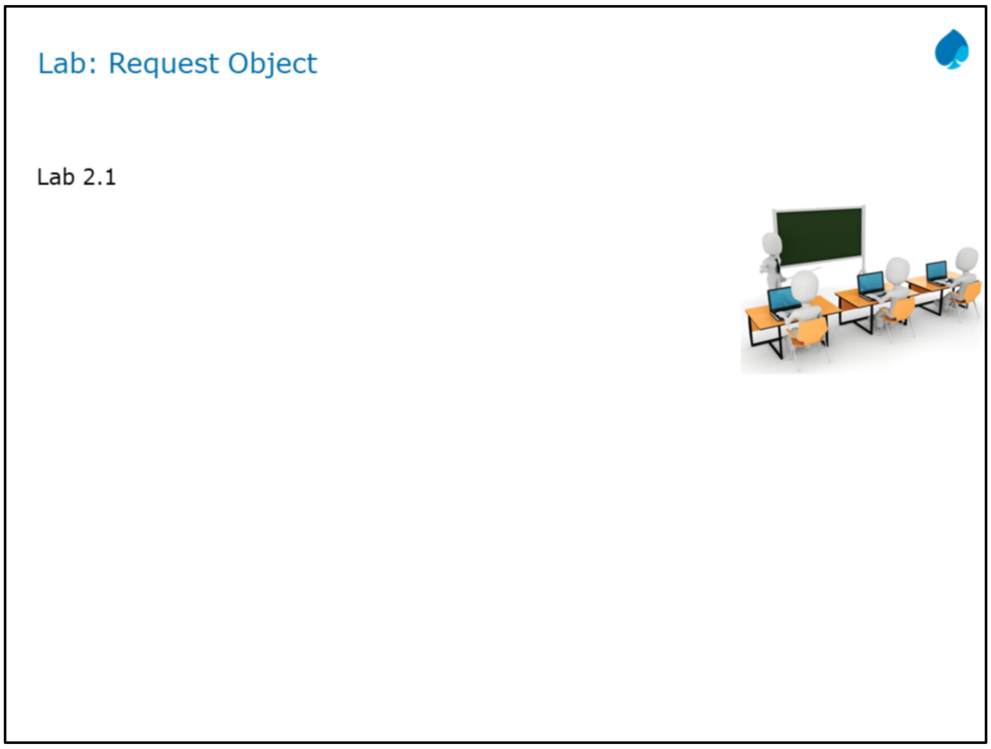
Execute the ProcessRequest servlet that reads form data.

Note:
Refer the com.igate.ch4.ProcessRequest.java class and userInput.html for data to be submitted from html page.

Invoke this servlet as follows:
http://localhost:9090/RequestResponseDemoApp/userInput.html.
Make appropriate choices and key in data. On form submit, the ProcessRequest servlet retrieves all form parameters and displays them.

Instructor Notes:

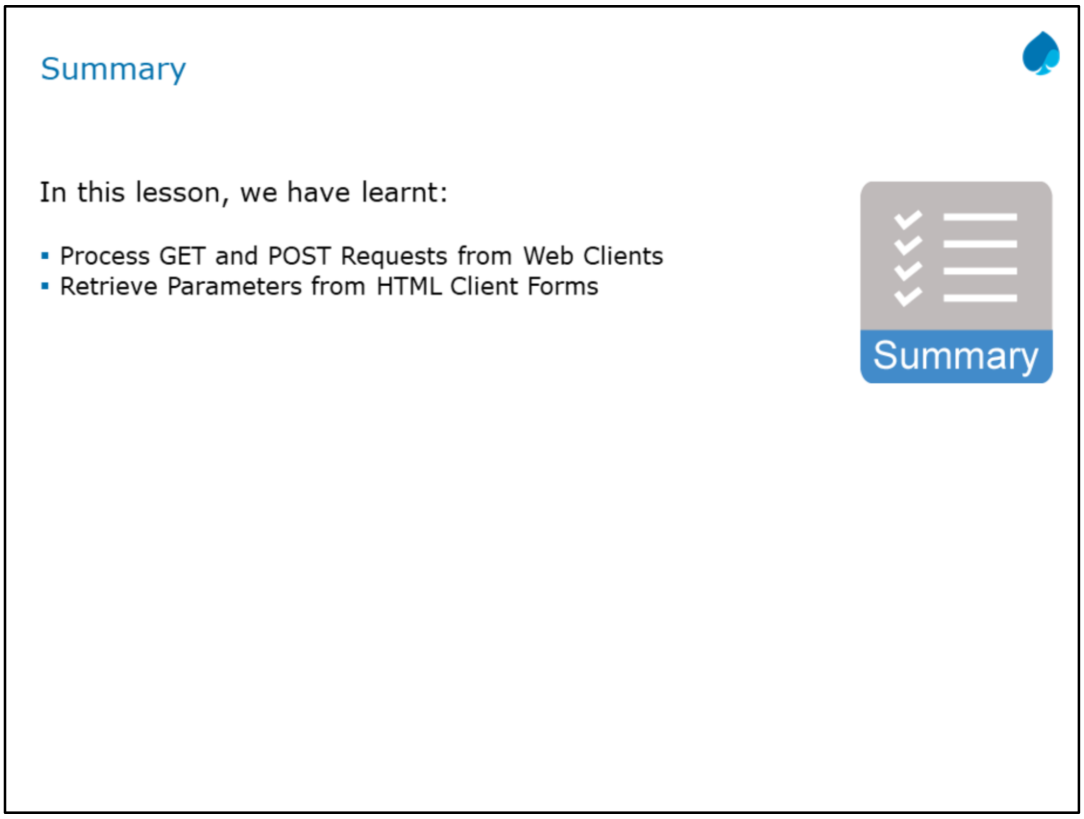
Lab on all the three forms of Scripting elements.



Lab: Request Object

Lab 2.1

Instructor Notes:



Summary

In this lesson, we have learnt:

- Process GET and POST Requests from Web Clients
- Retrieve Parameters from HTML Client Forms

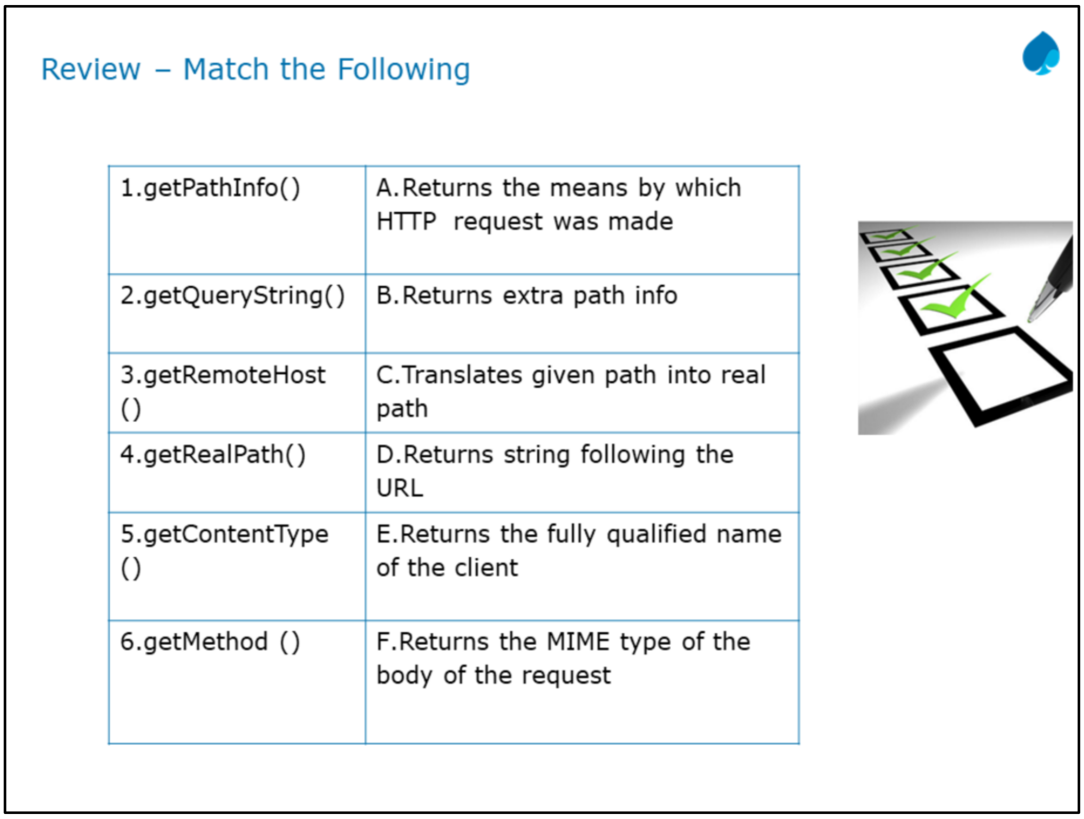
Summary

Instructor Notes:

Answers for Match the Following:

Please give participants about 5 minutes to refer to Appendix-B and extra API's for the Request object. Then only conduct review quiz on this slide.

1-b
2-d
3-e
4-c
5-f
6-a



Review – Match the Following

1.getPathInfo()	A.Returns the means by which HTTP request was made
2.getQueryString()	B.Returns extra path info
3.getRemoteHost ()	C.Translates given path into real path
4.getRealPath()	D.Returns string following the URL
5.getContentType ()	E.Returns the fully qualified name of the client
6.getMethod ()	F.Returns the MIME type of the body of the request

To Participants : refer to Appendix-B and extra API's for the Request object.