**Servlet – RequestDispatcher**

The RequestDispatcher is an Interface that comes under package***javax.servlet***. Using this interface we get an object in servlet after receiving the request. Using the RequestDispatcher object we send a request to other resources which include (*servlet, HTML file, or JSP file*). A RequestDispatcher object can be used to forward a request to the resource or to include the resource in a response. The resource can be dynamic or static.

**How to Create an Object of RequestDispatcher?**

There are **three ways** to get an object:

***1.*** *RequestDispatcher requestDispatcher=ServletContext.getRequestDispatcher(String path);*

***Description:***

* *public interface ServletContext. Defines a set of methods that a servlet uses to communicate with its servlet container.*
* *path is a string specifying the pathname to the resource(servlet, HTML file, or JSP file).*

***2.*** *RequestDispatcher requestDispatcher=ServletContext.getNamedDispatcher(String name);*

***Description:***

* *public interface ServletContext. Defines a set of methods that a servlet uses to communicate with its servlet container.*
* *name is a string specifying the name of a servlet to wrap.*

***3.*** *RequestDispatcher requestDispatcher=request.getRequestDispatcher(“String path”);*

***Description:***

* *request is the HttpServletRequest type object.*
* *path is a string specifying the pathname to the resource. If it is relative, it must be relative to the current servlet.*

**Method and Description**

The class contains **two methods**:

**1. forward**

***Syntax:***

*void forward(ServletRequest request,ServletResponse response) throws ServletException,IOException*

***Description:***

* ***Modifier and Type:-*** *void*
* *This method is used to forward a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.*
* *The method get called before the response has been sent to the client. If the response is already sent then the method will throws an IllegalStateException.*
* *The parameter request(HttpServletRequest type) and response(HttpServletResponse type) are the same objects as were passed to the calling servlet’s service method.*
* *This method sets the dispatcher type of the given request to DispatcherType.FORWARD.*

**Example:**

**Java**

|  |
| --- |
| **import** java.io.\*;  **import** javax.servlet.\*;  **import** javax.servlet.http.\*;    **public** **class** GFG **extends** HttpServlet {  **public** **void** doPost(HttpServletRequest request,                         HttpServletResponse response)      {          // Perform all the work as per your            // application's architecture  **try** {              RequestDispatcher requestDispatcher;                // path is a string specifying the pathname to              // the resource. If it is relative, it must be              // relative against the current servlet              requestDispatcher=request.getRequestDispatcher("path");              requestDispatcher.forward(request, response);          }  **catch** (ServletException servletException) {          }  **catch** (IOException ioException) {          }  **catch** (IllegalStateException illegalStateException) {          }      }  } |

***Note****: The above code will not run in online IDE this is server-side code.*

**2. include**

***Syntax:***

*void include(ServletRequest request,ServletResponse response) throws ServletException,IOException*

***Description:***

* *Modifier and Type:- void*
* *This method is used to include the response of resource(for which the request passed servlet, JSP page, HTML file) in the current servlet response.*
* *The parameter request(HttpServletRequest type) and response(HttpServletResponse type) are the same objects as were passed to the calling servlet’s service method.*
* *This method sets the dispatcher type of the given request to DispatcherType.INCLUDE.*

**Example:**

**Java**

|  |
| --- |
| **import** java.io.\*;  **import** javax.servlet.\*;  **import** javax.servlet.http.\*;    **public** **class** GFG **extends** HttpServlet {  **public** **void** doPost(HttpServletRequest request,                         HttpServletResponse response)      {          // Perform all the work as          // per your application's architecture  **try** {              RequestDispatcher requestDispatcher;                // path is a string specifying the pathname to              // the resource. If it is relative, it must be              // relative against the current servlet              requestDispatcher=request.getRequestDispatcher("path");              requestDispatcher.include(request, response);          }  **catch** (ServletException servletException) {          }  **catch** (IOException ioException) {          }      }  } |

***Note****: The above code will not run in online IDE this is server-side code.*

**Fields and Description**

| **Type** | **Name of Field** | **Description** |
| --- | --- | --- |
| static String | FORWARD\_REQUEST\_URI | The string contains the name of the request attribute under which the original request URI is made available to the target of a forward. |
| static String | FORWARD\_CONTEXT\_PATH | The string contains the name of the request attribute under which the original context path is made available to the target of a forward. |
| static String | FORWARD\_PATH\_INFO | The string contains the name of the request attribute under which the original path info is made available to the target of a forward. |
| static String | FORWARD\_SERVLET\_PATH | The string contains the name of the request attribute under which the original servlet path is made available to the target of a forward. |
| static String | FORWARD\_QUERY\_STRING | The string contains the name of the request attribute under which the original query string is made available to the target of a forward. |
| static String | INCLUDE\_REQUEST\_URI | The string contains the name of the request attribute under which the request URI of the target of include is stored. |
| static String | INCLUDE\_CONTEXT\_PATH | The string contains the name of the request attribute under which the context path of the target of an include is stored. |
| static String | INCLUDE\_PATH\_INFO | The string contains the name of the request attribute under which the path info of the target of an include is stored. |
| static String | INCLUDE\_SERVLET\_PATH | The string contains the name of the request attribute under which the servlet path of the target of an include is stored. |
| static String | INCLUDE\_QUERY\_STRING | The string contains the name of the request attribute under which the query string of the target of an include is stored. |
| static String | ERROR\_EXCEPTION | The string contains the name of the request attribute under which the exception object is propagated during an error dispatch. |
| static String | ERROR\_EXCEPTION\_TYPE | The string contains the name of the request attribute under which the type of the exception object is propagated during an error dispatch. |
| static String | ERROR\_MESSAGE | The string contains the name of the request attribute under which the exception message is propagated during an error dispatch. |
| static String | ERROR\_REQUEST\_URI | The string contains the name of the request attribute under which the request URI whose processing caused the error is propagated during an error dispatch. |
| static String | ERROR\_SERVLET\_NAME | The string contains the name of the request attribute under which the name of the servlet in which the error occurred is propagated during an error dispatch. |
| static String | ERROR\_STATUS\_CODE | The string contains the name of the request attribute under which the response status is propagated during an error dispatch. |