URL class in Java with Examples

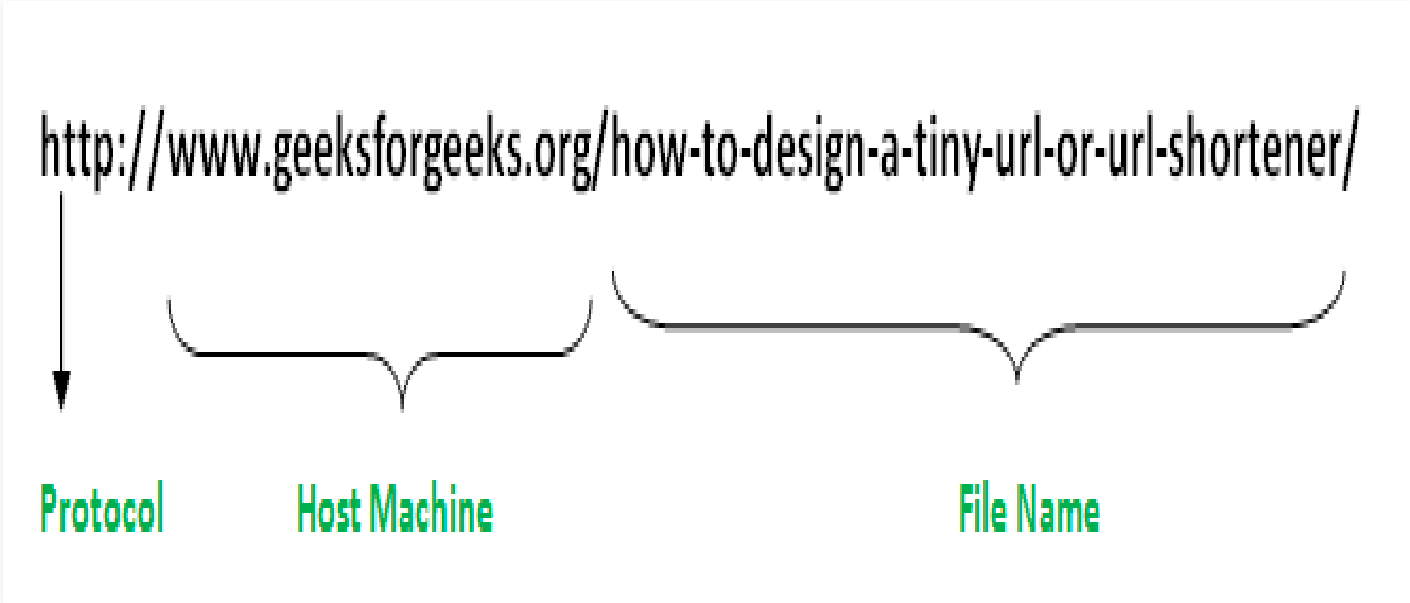
Difficulty Level : Easy Last Updated : 22 Jun, 2021

The URL class is the gateway to any of the resource available on the internet. A Class URL represents a Uniform Resource Locator, which is a pointer to a “resource” on the World Wide Web. A resource can point to a simple file or directory, or it can refer to a more complicated object, such as a query to a database or to a search engine

What is a URL?

As many of you must be knowing that Uniform Resource Locator-URL is a string of text that identifies all the resources on Internet, telling us the address of the resource, how to communicate with it and retrieve something from it.

A Simple URL looks like:



Components of a URL:-

A URL can have many forms. The most general however follows three-components system-

1. **Protocol:** HTTP is the protocol here
2. **Hostname:** Name of the machine on which the resource lives.
3. **File Name:** The path name to the file on the machine.
4. **Port Number:** Port number to which to connect (typically optional).

Some constructors for URL class:-

1. **URL(String address) throws MalformedURLException:** It creates a URL object from the specified String.
2. **URL(String protocol, String host, String file):** Creates a URL object from the specified protocol, host, and file name.
3. **URL(String protocol, String host, int port, String file):** Creates a URL object from protocol, host, port and file name.
4. **URL(URL context, String spec):** Creates a URL object by parsing the given spec in the given context.

5. **URL(String protocol, String host, int port, String file, URLStreamHandler handler):-** Creates a URL object from the specified protocol, host, port number, file, and handler.
6. **URL(URL context, String spec, URLStreamHandler handler):-** Creates a URL by parsing the given spec with the specified handler within a specified context.

Sample Program:

```
// Java program to demonstrate working of URL
import java.net.MalformedURLException;
import java.net.URL;

public class URLclass1
{
    public static void main(String[] args)
                throws MalformedURLException
    {

        // creates a URL with string representation.
        URL url1 =
        new URL("https://www.google.co.in/?gfe_rd=cr&ei=ptYq" +
                "WK26I4fT8gfth6CACg#q=geeks+for+geeks+java");

        // creates a URL with a protocol,hostname,and path
        URL url2 = new URL("http", "www.geeksforgeeks.org",
                           "/jvm-works-jvm-architecture/");

        URL url3 = new URL("https://www.google.co.in/search?"+
                           "q=gnu&rlz=1C1CHZL_enIN71" +
                           "4IN715&oq=gnu&aqs=chrome..69i57j6" +
                           "9i60l5.653j0j7&sourceid=chrome&ie=UTF" +
                           "-8#q=geeks+for+geeks+java");

        // print the string representation of the URL.
        System.out.println(url1.toString());
```

```
            System.out.println(url2.toString());
            System.out.println();
            System.out.println("Different components of the URL3-");

            // retrieve the protocol for the URL
            System.out.println("Protocol:- " + url3.getProtocol());

            // retrieve the hostname of the url
            System.out.println("Hostname:- " + url3.getHost());

            // retrieve the default port
            System.out.println("Default port:- " +
                                        url3.getDefaultPort());

            // retrieve the query part of URL
            System.out.println("Query:- " + url3.getQuery());

            // retrieve the path of URL
            System.out.println("Path:- " + url3.getPath());

            // retrieve the file name
            System.out.println("File:- " + url3.getFile());

            // retrieve the reference
            System.out.println("Reference:- " + url3.getRef());
        }
    }
```

Output:

```
https://www.google.co.in/?gfe_rd=cr&ei=ptYqWK26I4fT8gfth6CACg#q=geeks+for+gee
https://www.geeksforgeeks.org/jvm-works-jvm-architecture/

Different components of the URL3-
Protocol:- https
Hostname:- www.google.co.in
Default port:- 443
Query:- q=gnu&rlz=1C1CHZL_enIN714IN715&oq=gnu&aqs=chrome..69i57j69i60l5.653j0
Path:- /search
File:- /search?q=gnu&rlz=1C1CHZL_enIN714IN715&oq=gnu&aqs=chrome..69i57j69i60l
Reference:- q=geeks+for+geeks+java
```

Explanation of some methods used in the above program is as follows:-

1. **public String toString():** As in any class, toString() returns the string representation of the given URL object.
2. **public String getAuthority():** returns the authority part of URL or null if empty.
3. **public String getPath():** returns the path of the URL, or null if empty.
4. **public String getQuery():** returns the query part of URL. Query is the part after the '?' in the URL. Whenever logic is used to display the result, there would be a query field in URL. It is similar to querying a database.
5. **public String getHost():** return the hostname of the URL in IPv6 format.
6. **public String getFile():** returns the file name.
7. **public String getRef():** Returns the reference of the URL object. Usually, the reference is the part marked by a '#' in the URL. You can see the working example by querying anything on Google and seeing the part after '#'.
8. **public int getPort():** returns the port associated with the protocol specified by the URL.
9. **public int getDefaultPort:** returns the default port used.
10. **public String getProtocol():** returns the protocol used by the URL.

References

http://docs.oracle.com/javase/7/docs/api/java/net/URL.html#getAuthority()

This article is contributed by **Rishabh Mahrsee**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.