Victor-Cristian Catana, group 244  
Servlet-based URL Checker

**Short description**

The application features two servlets, UrlChecker and UrlPinger, and allows a user to check if a URL is valid and to check if a connection can be made to that URL. This is accomplished using built-in methods of the java.net.URL class, more specifically the constructor, which throws an exception should the provided string not be parsable to a URL object, and the toURI() method which further checks if the URL can be converted to an URI and throws an exception otherwise. Should the provided URL be valid, the user is then forwarded to the next page where he can choose to ping the webpage that URL points to and check if a connection can be made. This is done by opening a connection to that URL and creating an HttpUrlConnection object from which we can thereafter retrieve the response code and message to accurately determine the exact state of the connection.

The two servlets communicate with each other by means of forwarding the GET request in the doGet method of the UrlChecker class to the doGet method of the UrlPinger class which then processes the request and returns the response.

**Structure of the application**

The embedded deployment versions of the application also feature a class named JettyServer and TomcatServer respectively, for defining code-based server context and servlet mappings, and a ClientHtml class for initial rendering of the home page. All variants of the application, independent of deployment method, feature two main classes, UrlChecker and UrlPinger, which both extend the HttpServlet class and override the base doGet(HttpServletRequest, HttpServletResponse) and doPost(HttpServletRequest, HttpServletResponse) methods.

UrlChecker

Overrides base HttpServlet “*doPost*” method wherein the action performed is creating a URL(String spec) object, where the “*spec*” parameter is the string used as user input in the UI, if the user wrote anything in the text-box in the graphic interface. Afterwards, the method prints out a page where the user is either asked to enter a URL again or asked to use another text-box input if they choose to also ping the URL they wrote, should it be valid. If the user chooses to ping the URL, the overridden “*doGet*” method forwards the request to the UrlPinger class.

UrlPinger

Overrides base HttpServlet “*doGet*” method wherein it creates an HttpURLConnection object, “*conn*”, by opening a connection through an URL object, with the submitted “*spec*” value from UrlChecker, and casting it to the HttpURLConnection type. The request is the configured with a request method and connection timeout values. The method then calls the “*getResponseCode()*” and “*getResponseMessage()*” methods of the “*conn*” object, initiating a request, in order to print these to the page, should the request be completed successfully. Should the request fail, a failure message is then printed to the screen.

**Deployment and versions**

The application was developed in IntelliJ IDEA 2022.3 using SDK Java 11.

Gradle was used for building the project and configuring dependencies.

The deployment methods featured are as follows:

* Tomcat embedded, using tomcat-embed-core:8.5.23
* Tomcat external context, using xml file for defining context, version 9.0.70
* Tomcat war-based deployment, using manager app, version 9.0.70
* Jetty embedded, using jetty-server:9.4.3
* Jetty external context, using xml file for defining context in jetty-base, version 10.0.13
* Jetty war-based deployment, deploying war directly to jetty-base, version 10.0.13