Advanced Programming

Week 8

Network Programming

- Introduction
- Java Networking
- Manipulating URLs
- Java Socket Programming
- InetAddress

Tilahun Melak(PhD)

October, 2025

Objectives

At the end of this lecture, students will be able to:

- Explain Java networking classes and interfaces.
- Explain advantages of Java networking
- Manipulate URLs
- Describe different types of sockets and their functionalities.

Introduction

- Java provides:-
  - Stream-based communications
    - a process establishes a connection to another process. While the connection is in place, data flows between the processes
  - Connection-based protocol - Uses TCP
  - Packet-based communications – use UDP
    - Individual packets transmitted

Oracle. (n.d.). Java networking overview. In Java Platform, Standard Edition Documentation.

Introduction…

- Client-server relationship
  - Client requests some action to be performed
  - Server performs the action and responds to client
  - Request-response model
    - Common implementation: Web browsers and Web servers

Oracle. (n.d.). Java networking: Client-server communication. Oracle Java Documentation.

Java Networking

- Java Networking is a concept of connecting two or more computing devices together so that we can share resources.
- Java Network programming provides facility to share data between different computing devices.

Oracle. Java Networking Overview. In The Java™ Tutorials: Custom Networking. Oracle, 2024.

Java Networking…

- **Advantage of Java Networking**
  - ✓ sharing resources
  - ✓ centralize software management

  Java Networking terminology
  - IP Address
  - Protocol
  - Port number
  - MAC Address, connection-oriented and connection-less protocol

Oracle. (2023). Java platform, standard edition documentation: Networking. Oracle.

Java Networking Classes and Interfaces

Table :Java networking classes and interfaces

NetworkInterface	InetAddress	ServerSocket	Socket
DatagramSocket	DatagramPacket	URL	URLConnection
HttpURLConnection	InetSocketAddress	SocketAddress	Authenticator
PasswordAuthentication	MulticastSocket	URLEncoder	URLDecoder
Remote	Naming	Registry	LocateRegistry

Oracle. (2023). Java platform, standard edition documentation: Networking. Oracle.

Java Networking Classes and Interfaces…

Table :Java networking classes and interfaces

UnicastRemoteObject	RMISecurityManager	RMISocketFactory	RMIClassLoader
RemoteServer	RemoteObject	Session	MimeMessage
Transport	Store	Folder	Message
URLName	Header	ContentType	InternetAddress
MimeBodyPart	MimeMultipart	InternetHeaders	MimeUtility

Oracle. (2023). Java platform, standard edition documentation: Networking. Oracle.

Manipulating URLs

- The HTTP protocol uses URIs to locate data on the Internet.
- A URI that represents a document is called a URL
- An object of the java.net.URL class represents a URL.

Oracle. (2023). *Java platform, standard edition documentation: The URL class*. Oracle.

Manipulating URLs…

- The URL class
  - Has constructors that constructs a URL object:
    - By taking an absolute URL in string form
    - By taking the URL's component parts:- protocol, host name …
    - By taking a base and relative URL

Oracle. (2023). *Java platform, standard edition documentation: The URL class*. Oracle.

Manipulating URLs…

- The URL class
  - Constructors throw a MalformedURLException if
    - The protocol is an unsupported protocol or
    - The URL is syntactically incorrect.
  - Has methods to
    - Parse the different parts of a URL
    - Gets an input stream from a URL, so you can read data from server
    - Get content from a server as a java object.

Oracle. (2023). *Java platform, standard edition documentation: The URL class*. Oracle.

Example: Manipulating URLs

```
import java.io.*;
import java.net.*;
public class URLDemo{
public static void main(String[] args){
try{
URL url=new
    URL("https://www.tutorialspoint.com/java/java_networking.htm");
```

Example: Manipulating URLs

```
System.out.println("Protocol: "+url.getProtocol());
System.out.println("Host Name: "+url.getHost());
System.out.println("Port Number: "+url.getPort());
System.out.println("File Name: "+url.getFile());

}catch(Exception e){System.out.println(e);}
}
}
```

Manipulating URLs…

- The openStream( ) method
  - connects to the resource referenced by the URL,
  - and returns an InputStream from which data can be read.
  - The data you get from this InputStream is the raw contents of the file the URL references:

Oracle. (2023). *Java platform, standard edition documentation: The URL class*. Oracle.

Manipulating URLs…

- The openConnection( ) method
  - opens a socket to the specified URL
  - and returns a URLConnection object.

Oracle. (2023). *Java platform, standard edition documentation: The URL class*. Oracle.

Manipulating URLs…

- The URLConnection class
  - represents an active connection to a resource specified by a URL.
  - Its instance can be used both to read from and to write to the resource.

Oracle. (2023). *Java platform, standard edition documentation: The URL class*. Oracle.

Example: Manipulating URLs

```
import java.io.*;
    import java.net.*;
    public class URLConnectionDemo {
    public static void main(String[] args){
    try{
    URL url=new
     URL("https://www.tutorialspoint.com/java/java_networking.htm");
```

Example: Manipulating URLs

```
 URLConnection urlcon=url.openConnection();
InputStream stream=urlcon.getInputStream();
int i;
while((i=stream.read())!=-1){
System.out.print((char)i);
}
}catch(Exception e){System.out.println(e);}
}
}
```

Java Socket Programming

- A socket, in network terminology, is an end-point for communication between two processes on the network.
- Java **Socket programming** is used for communication between the applications running on different JRE.
- Java Socket programming can be connection-oriented or connection-less.

Oracle. (2023). Java platform, standard edition documentation: Sockets. Oracle.

Java Socket Programming…

- **Socket** and **ServerSocket** classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket classes are used for connection-less socket programming.
- The client in socket programming must know two information:
    1. IP Address of Server, and
    2. Port Number

Oracle. (2023). Java platform, standard edition documentation: Sockets. Oracle.

The InetAddress class

- Represents IP address
- Performs name lookup and reverse lookup
- Has no public constructor
  - has static methods that return InetAddress objects given a little information.

Oracle. (2023). Java platform, standard edition documentation: The InetAddress class. Oracle.

The InetAddress class…

- InetAddress getByName(String hostName)
  - hostName can be “java.sun.com”, or “130.95.72.134”
  - these methods may make a connection to the local DNS server to fill out the information in the InetAddress object

Oracle. (2023). Java platform, standard edition documentation: The InetAddress class. Oracle.

The InetAddress class…

- InetAddress getLocalHost()
  - Returns the localhost
- getByAddress (byte[] addr)
  - returns an InetAddress object given the raw IP address
- getByAddress(String host, byte[] addr)
  - Create an InetAddress based on the provided host name and IP address No name service is checked for the validity of the address.

Oracle. (2023). Java platform, standard edition documentation: The InetAddress class. Oracle.

Example: InetAdress

```
import java.io.*;
import java.net.*;
public class InetIPFinder{
public static void main(String[] args){
try{
InetAddress ip=InetAddress.getByName("www.google.com");

System.out.println("Host Name: "+ip.getHostName());
System.out.println("IP Address: "+ip.getHostAddress());
}catch(Exception e){System.out.println(e);}
}
}
```

The java.net.Socket class

- A socket is a connection between two hosts.
- The Socket class allows you to create a socket that performs the following operations:
  - Connect to a remote machine
  - Send data
  - Receive data
  - Close a connection

Oracle. (2023). *Java platform, standard edition documentation: The Socket class*. Oracle.

The java.net.Socket class…

- It's constructors
  - specify the host and the port number you want to connect to.
    - Hosts may be specified as an InetAddress or a String.
    - Ports are always specified as int values from 0 to 65,535.

```
Socket s = new Socket(host, port);
```

Oracle. (2023). Java platform, standard edition documentation: The Socket class. Oracle.

The java.net.Socket class…

- Two of the constructors also specify the local address and local port from which data will be sent.

```
Socket s = new Socket(host, port, lAddr, lPort);
```

Oracle. (2023). *Java platform, standard edition documentation: The Socket class*. Oracle.

The java.net.Socket class…

- The constructors do not just create a socket object. They also attempts to connect the underlying socket to the remote server.
- The getInputStream( ) method
  - returns an input stream that can read data from the socket into a program.
- The getOutputStream() method
  - returns a raw OutputStream for writing data from your application to the other end of the socket.

Oracle. (2023). *Java platform, standard edition documentation: The Socket class*. Oracle.

Summary

- In today's lecture we have discussed about;
  - Java networking concepts
  - Java Networking Classes and Interfaces
  - Key advantages
  - URL manipulation, and the types and functionalities of sockets.
  - Examples

References

- Oracle. (n.d.). Java networking overview. In Java Platform, Standard Edition Documentation.
- Oracle. (n.d.). *Java networking: Client-server communication*. Oracle Java Documentation.
- Oracle. *Java Networking Overview*. In *The Java™ Tutorials: Custom Networking*. Oracle, 2024.
- Oracle. (2023). *Java platform, standard edition documentation: Networking*. Oracle.
- Oracle. (2023). Java platform, standard edition documentation: Sockets. Oracle.