Network programming (IT423+IT432)

Spring 2017

Dr. Islam Taj-Eddin

IT Dept., FCI, Assiut Univ.

Internet Addresses

Internet addresses

- Every host on the Internet is identified by a unique, four-byte Internet Protocol (IP) address.
- This is written in *dotted quad* format like 199.1.32.90 where each byte is an unsigned integer between 0 and 255.
- There are about four billion unique IP addresses, but they aren't very efficiently allocated
- IPv4
- IPv6 (eight blocks of four hexadecimal digits separated by colons)

InetAddress Class

- ***Java.net.InetAddress***
  - ***Represents an IP address (xxx.xxx.xxx.xxx)***
- **Converts:**
  - ***xxx.xxx.xxx.xxx → machineName.domainName***
  - ***machineName.domainName → xxx.xxx.xxx.xxx***
- **Used by other network classes :**
  - ***Socket***
  - ***ServerSocket***
  - ***…***

InetAddress Class

- **No public *InetAddress( )* Constructors**
  - **Arbitrary addresses may not be created**
  - **All addresses checked with *DNS***

- **Provides objects that you can use to manipulate and deal with IP addresses and domain names.**

- **Class provides several static methods that return an object of type InetAddress*.***

InetAddress Class Methods

- **getByName ()**
  - ***Public static InetAddress getByName(host)***
    - **Throws UNknownHostException**
    - **Returns an InetAddress object representing *host***
    - **Can be used to determine the IP address of a host, given the host's name.**
    - **Host:**
      - **machine name: *java.sun.com***
      - **IP address: *206.26.48.100***

InetAddress Class
getByName (host)

```
InetAddress java1, java2;

try {
java1 = InetAddress.getByName(“java.sun.com");
java2 = InetAddress.getByName("128.238.2.92");
}
catch (UnknownHostException e) { System.err.println(e);
 }
{System.out.println(java1);
 ...
 }
```

InetAddress Class
getAllByName (host)

- **Returns an array of InetAddress objects.**
  - **IP addresses of the specified host.**

InetAddress Class
getLocalHost (host)

- **Returns an InetAddress object representing the local host computer.**

InetAddress Class

➔ Show Url001.java

```
Get and display IP address of URL by name
wpi.wpi.edu/130.215.24.6
Do reverse lookup on the IP address
wpi.WPI.EDU/130.215.24.6
Get and display current IP address of LocalHost
grover.WPI.EDU/130.215.25.67
Do reverse lookup on current IP address of LocalHost
grover.wpi.edu/130.215.25.67
Get and display current name of LocalHost
grover.wpi.edu
Get and display current IP address of LocalHost
130 215 25 67
```

Java 1.4 Added Methods

- Java 1.4 adds two more factory methods that do not check their addresses with the local DNS server.
- The first creates an InetAddress object with an IP address and no hostname.
  - public static InetAddress getByAddress(byte[ ] address) throws UnknownHostException
- The second creates an InetAddress object with an IP address and a hostname.
  - public static InetAddress getByAddress(String hostName, byte[] address) throws UnknownHostException

Getter Methods

They return the hostname as a string and the IP address as both string and a byte array

- **public String getHostName()**
- **public byte[ ] getAddress()**
- **public String getHostAddress()**

Address Types

- **Public boolean isAnyLocalAddress()**
- **Public boolean isLoopbackAddress()**
- **Public boolean isLinkLocalAddress()**
- **Public boolean isSiteLocalAddress()**
- **Public boolean isMulticastAddress()**
- **Public boolean isMCGloabl()**
- **Public boolean isMCNodeLocal()**
- **Public boolean isMCLinkLocal()**
- **Public boolean isMCSiteLocal()**
- **Public boolean isMCOrgLocal()**

Object Methods

- Public boolean equals (Object o)
- Public int hashCode ()
- Public String toString ()

Inet4Address and Inet6Address

- Public final class Inet4Address extends InetAddress

- Public final class Inet6Address extends InetAddress

Using InetAddress objects

```
import java.net.InetAddress;
import java.net.UnknownHostExcepion;

public static void main(String[] args)
{
   try {
      InetAddress inet1 =
            InetAddress.getByName("asp.ee.uwa.edu.au");
      System.out.println(
            "HostAddress=" + inet1.getHostAddress());
      InetAddress inet2 =
            InetAddress.getByName("130.95.72.134");
      System.out.println("HostName=" + inet2.getHostName());
      if (inet1.equals(inet2))
         System.out.println("Addresses are equal");
   }
   catch (UnknownHostException uhe) {
      uhe.printStackTrace();
   }
}
```