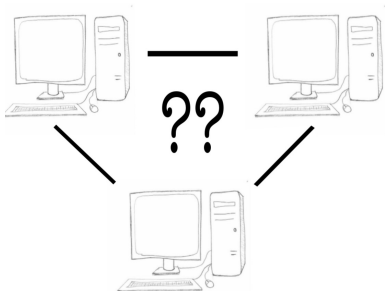


PROGRAMMATION RÉSEAU

Arnaud Sangnier
sangnier@irif.fr

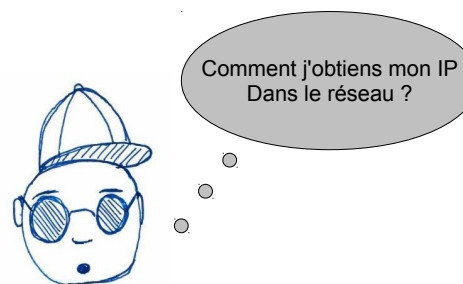
Obtenir son IP dans le réseau



Exemple java 1

```
public class OwnIp{
    public static void main(String[]args){
        try{
            InetAddress myIA=InetAddress.getLocalHost();
            System.out.println(myIA.toString());
            System.out.println(myIA.isAnyLocalAddress());
        } catch (Exception e){
            e.printStackTrace();
        }
    }
}
```

Sortir du réseau local



- En Java, normalement `InetAddress.getLocalHost();` suffit
- En C, il faut être plus subtil
- Il faut avoir les adresses disponibles et tester celles qui ne sont pas des loop-back

Exemple java 2

```
public class OwnIp2{
    public static void main(String[]args){
        try{
            Enumeration<NetworkInterface>
            listNi=NetworkInterface.getNetworkInterfaces();
            while(listNi.hasMoreElements()){
                NetworkInterface nic=listNi.nextElement();
                System.out.println("Network Interface :");
                System.out.println(nic.toString());
                Enumeration<InetAddress> listIa=nic.getInetAddresses();
                while(listIa.hasMoreElements()){
                    InetAddress iac=listIa.nextElement();
                    System.out.println("++++++ InetAddress :");
                    System.out.println("++++++ "+iac.toString());
                    if(iac instanceof Inet4Address){
                        System.out.println("IPV4");
                    }
                    if(iac.isLoopbackAddress()){
                        System.out.println("Loop Back Address");
                    }
                }
            }
        } catch (Exception e){
            e.printStackTrace();
        }
    }
}
```

Exemple C

```
int main() {
    struct ifaddrs *myaddrs, *ifa;
    struct sockaddr_in *s4;
    int status;
    char *ip=(char *)malloc(64*sizeof(char));
    status = getifaddrs(&myaddrs);
    if (status != 0){
        perror("Probleme de recuperation d'adresse IP");
        exit(1);
    }
    for (ifa = myaddrs; ifa != NULL; ifa = ifa->ifa_next){
        if (ifa->ifa_addr == NULL) continue;
        if ((ifa->ifa_flags & IFF_UP) == 0) continue;
        if ((ifa->ifa_flags & IFF_LOOPBACK) != 0) continue;
        if (ifa->ifa_addr->sa_family == AF_INET){
            s4 = (struct sockaddr_in *) (ifa->ifa_addr);
            if (inet_ntop(ifa->ifa_addr->sa_family, (void *)&(s4->sin_addr),
ip, 64*sizeof(char)) != NULL){
                printf("Adresse IP :%s\n",ip);
            }
        }
    }
    freeifaddrs(myaddrs);
}
```