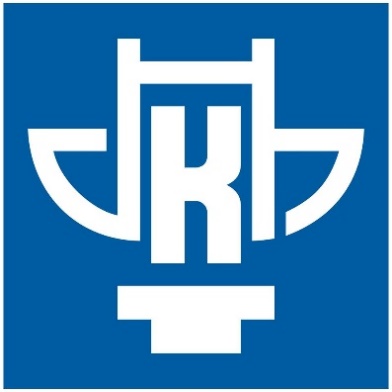
**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**Khoa Công Nghệ Thông Tin**

**\*\*\*\*\*\*\*\*\*\***

****

**Đề tài : Java Networking**

**Môn : Công Nghệ Java**

**Giáo viên hướng dẫn : Thầy Nguyễn Hồng Thanh**

**Sinh viên thực hiện : Nhóm 2**

**Lương Tuấn Minh Đỗ Đức Thắng**

**Chử Mạnh Thủy Ninh Công Hùng**

**Trần Văn Tú Trần Minh Hiếu**

**Lê Sơn Tùng**

**Trần Hồng Quân**

**Nguyễn Trần Đức Mạnh**

***Hà Nội, 22-12-2020***

# Quyền truy cập có lập trình vào các thông số mạng

1. Java Networking

Thuật ngữ **lập trình mạng** với java đề cập đến việc viết các chương trình thực hiện trên nhiều thiết bị (máy tính), trong đó các thiết bị được kết nối với nhau.

Gói **java.net** của J2SE APIs chứa một tập hợp các lớp và giao tiếp cung cấp giao thức truyền thông ở mức độ thấp.

Gói java.net cung cấp hỗ trợ cho hai giao thức mạng phổ biến sau:

* **TCP** - TCP là viết tắt của Transmission Control Protocol. TCP thường được sử dụng qua giao thức Internet (Internet Protocol), được gọi là TCP/IP. Giao thức này cho phép giao tiếp tin cậy giữa hai ứng dụng.
* **UDP** - UDP là viết tắt của User Datagram Protocol, một giao thức cho phép truyền dữ liệu giữa các ứng dụng. Giao thức này không kiểm tra đến việc gói tin đã được gửi hay chưa, nên đây là giao tiếp không tin cậy giữa hai hoặc nhiều ứng dụng.

TCP và UDP là các giao thức cốt lõi của việc kết nối các thiết bị công nghệ với nhau. Ngoài ra, trong việc lập trình mạng với java chúng ta có thể sử dụng [FTP](https://viettuts.vn/lap-trinh-mang-voi-java/ftp-la-gi), công nghệ J2EE...

1. Quyền truy cập có lập trình vào các thông số mạng

Các hệ thống thường chạy với nhiều kết nối mạng đang hoạt động, chẳng hạn như Ethernet có dây và bluetooth. Một số ứng dụng có thể cần phải truy cập thông tin này để thực hiện các hoạt động mạng cụ thể trên một kết nối cụ thể.802.11 b/g

Lớp [java.net.NetworkInterface](https://docs.oracle.com/javase/8/docs/api/java/net/NetworkInterface.html) cung cấp quyền truy cập vào thông tin này.

# Giao diện mạng là gì?

Giao *diện mạng* là điểm kết nối giữa máy tính và mạng riêng hoặc mạng công cộng. Một giao diện mạng nói chung là một card giao diện mạng (NIC), nhưng không cần phải có một hình thức vật lý. Thay vào đó, giao diện mạng có thể được thực hiện trong phần mềm. Ví dụ: giao diện loopback (127.0.0.1 cho IPv4 và ::1 cho IPv6) không phải là thiết bị vật lý mà là một phần mềm mô phỏng giao diện mạng. Giao diện loopback thường được sử dụng trong môi trường thử nghiệm.

Lớp [java.net.NetworkInterface](https://docs.oracle.com/javase/8/docs/api/java/net/NetworkInterface.html) đại diện cho cả hai loại giao diện.

NetworkInterface rất hữu ích cho một hệ thống đa homed, đó là một hệ thống với nhiều NIC. Sử dụng, bạn có thể chỉ định NIC nào để sử dụng cho một hoạt động mạng cụ thể. NetworkInterface

Ví dụ, giả sử bạn có một máy tính với hai cấu hình NIC, và bạn muốn gửi dữ liệu đến một máy chủ. Bạn tạo một ổ cắm như thế này:

Socket soc = new java.net.Socket();

soc.connect(new InetSocketAddress(address, port));

Để gửi dữ liệu, hệ thống xác định giao diện nào được sử dụng. Tuy nhiên, nếu bạn có một sở thích hoặc nếu không cần phải xác định NIC để sử dụng, bạn có thể truy vấn hệ thống cho các giao diện thích hợp và tìm thấy một địa chỉ trên giao diện bạn muốn sử dụng. Khi bạn tạo ổ cắm và liên kết nó với địa chỉ đó, Hệ thống sử dụng giao diện liên quan. Ví dụ:

NetworkInterface nif = NetworkInterface.getByName("bge0");

Enumeration<InetAddress> nifAddresses = nif.getInetAddresses();

Socket soc = new java.net.Socket();

soc.bind(new InetSocketAddress(nifAddresses.nextElement(), 0));

soc.connect(new InetSocketAddress(address, port));

Bạn cũng có thể sử dụng để xác định giao diện địa phương mà trên đó một nhóm phát đa hướng sẽ được tham gia. Ví dụ: NetworkInterface

NetworkInterface nif = NetworkInterface.getByName("bge0");

MulticastSocket ms = new MulticastSocket();

ms.joinGroup(new InetSocketAddress(hostname, port), nif);

NetworkInterface có thể được sử dụng với các API Java theo nhiều cách khác ngoài hai cách sử dụng được mô tả ở đây.

# Truy xuất Giao diện Mạng

Lớp học không có nhà xây dựng công cộng. Vì vậy, bạn không thể chỉ cần tạo ra một trường hợp mới của lớp học này với các nhà điều hành. Thay vào đó, các phương pháp tĩnh sau đây có sẵn để bạn có thể lấy các chi tiết giao diện từ hệ thống. Hai phương pháp đầu tiên được sử dụng khi bạn đã biết địa chỉ IP hoặc tên của giao diện cụ thể. Phương pháp thứ ba, trả về danh sách đầy đủ các giao diện trên máy. NetworkInterfacenewgetByInetAddress()getByName()getNetworkInterfaces()getNetworkInterfaces()

Giao diện mạng có thể được tổ chức theo cấp bậc. Các lớp học bao gồm hai phương pháp, đó là thích hợp với một hệ thống phân cấp giao diện mạng. Phương thức trả về cha mẹ của một giao diện. Nếu một giao diện mạng là một subinterface, trả về một giá trị không null. Phương pháp trả về tất cả các bề mặt con của một giao diện mạng. NetworkInterfacegetParent()getSubInterfaces()getParent()NetworkInterfacegetParent()getSubInterfaces()

Chương trình ví dụ sau liệt kê tên của tất cả các giao diện mạng và subinterfaces (nếu có tồn tại) trên một máy tính.

import java.io.\*;

import java.net.\*;

import java.util.\*;

import static java.lang.System.out;

public class ListNIFs

{

public static void main (String args[]) throws SocketException {

Enumeration<NetworkInterface> nets = NetworkInterface.getNetworkInterfaces();

for (NetworkInterface netIf : Collections.list(nets)) {

out.printf ("Display name: %s\n", netIf.getDisplayName());

out.printf ("Name: %s\n", netIf.getName());

displaySubInterfaces(netIf);

out.printf ("\n");

}

}

static void displaySubInterfaces (NetworkInterface netIf) throws SocketException {

Enumeration<NetworkInterface> subIfs = netIf.getSubInterfaces();

for (NetworkInterface subIf : Collections.list(subIfs)) {

out.printf ("\tSub Interface Display name: %s\n", subIf.getDisplayName());

out.printf ("\tSub Interface Name: %s\n", subIf.getName());

}

}

}

Sau đây là mẫu đầu ra từ chương trình ví dụ:

Display name: bge0

Name: bge0

Sub Interface Display name: bge0:3

Sub Interface Name: bge0:3

Sub Interface Display name: bge0:2

Sub Interface Name: bge0:2

Sub Interface Display name: bge0:1

Sub Interface Name: bge0:1

Display name: lo0

Name: lo0

* 1. **Liệt kê địa chỉ giao diện mạng**

Một trong những thông tin hữu ích nhất mà bạn có thể nhận được từ giao diện mạng là danh sách các địa chỉ IP được gán cho nó. Bạn có thể lấy thông tin này từ một NetworkInterfacephiên bản bằng cách sử dụng một trong hai phương pháp. Phương pháp đầu tiên, getInetAddresses(), trả về một Enumerationsố InetAddress. Phương thức khác getInterfaceAddresses(), trả về một danh sách các [java.net.InterfaceAddress](https://docs.oracle.com/javase/8/docs/api/java/net/InterfaceAddress.html) trường hợp. Phương pháp này được sử dụng khi bạn cần thêm thông tin về địa chỉ giao diện ngoài địa chỉ IP của nó. Ví dụ: bạn có thể cần thông tin bổ sung về mặt nạ mạng con và địa chỉ quảng bá khi địa chỉ là địa chỉ IPv4 và độ dài tiền tố mạng trong trường hợp là địa chỉ IPv6.

Chương trình ví dụ sau liệt kê tất cả các giao diện mạng và địa chỉ của chúng trên một máy:

nhập java.io. \*;

nhập java.net. \*;

nhập java.util. \*;

nhập tĩnh java.lang.System.out;

lớp công khai ListNets {

public static void main (String args []) ném SocketException {

Enumeration <NetworkInterface> nets = NetworkInterface.getNetworkInterfaces ();

cho (NetworkInterface netint: Collections.list (nets))

displayInterfaceInformation (cư dân mạng);

}

static void displayInterfaceInformation (NetworkInterface netint) ném SocketException {

out.printf ("Tên hiển thị:% s \ n", netint.getDisplayName ());

out.printf ("Tên:% s \ n", netint.getName ());

Liệt kê <InetAddress> inetAddresses = netint.getInetAddresses ();

cho (InetAddress inetAddress: Collections.list (inetAddresses)) {

out.printf ("InetAddress:% s \ n", inetAddress);

}

out.printf ("\ n");

}

}

Sau đây là kết xuất mẫu từ chương trình ví dụ:

Tên hiển thị: Giao diện TCP Loopback

Tên: lo

InetAddress: /127.0.0.1

Tên hiển thị: Kết nối mạng không dây

Tên: eth0

InetAddress: /192.0.2.0

# Tham số giao diện mạng

Bạn có thể truy cập các thông số mạng về giao diện mạng ngoài tên và địa chỉ IP được gán cho nó

Bạn có thể khám phá xem giao diện mạng có “lên” (nghĩa là đang chạy) hay không bằng isUP()phương pháp này. Các phương pháp sau chỉ ra loại giao diện mạng:

* isLoopback () cho biết giao diện mạng có phải là giao diện lặp lại hay không.
* isPointToPoint () cho biết giao diện có phải là giao diện điểm-điểm hay không.
* isVirtual () cho biết giao diện có phải là giao diện ảo hay không.

Các supportsMulticast ()phương pháp cho biết các mạng giao diện hỗ trợ multicasting. Các getHardwareAddress ()phương thức trả về địa chỉ phần cứng vật lý của giao diện mạng, thường được gọi là địa chỉ MAC, khi nó có sẵn. Các getMTU()phương pháp trả về Maximum Transmission Unit (MTU), đó là kích thước gói lớn nhất.

Ví dụ sau mở rộng trên ví dụ trong [Danh sách địa chỉ giao diện mạng](https://docs.oracle.com/javase/tutorial/networking/nifs/listing.html) bằng cách thêm các tham số mạng bổ sung được mô tả trên trang này:

nhập java.io. \*;

nhập java.net. \*;

nhập java.util. \*;

nhập tĩnh java.lang.System.out;

public class ListNetsEx {

public static void main (String args []) ném SocketException {

Enumeration <NetworkInterface> nets = NetworkInterface.getNetworkInterfaces ();

cho (NetworkInterface netint: Collections.list (nets))

displayInterfaceInformation (cư dân mạng);

}

static void displayInterfaceInformation (NetworkInterface netint) ném SocketException {

out.printf ("Tên hiển thị:% s \ n", netint.getDisplayName ());

out.printf ("Tên:% s \ n", netint.getName ());

Liệt kê <InetAddress> inetAddresses = netint.getInetAddresses ();

cho (InetAddress inetAddress: Collections.list (inetAddresses)) {

out.printf ("InetAddress:% s \ n", inetAddress);

}

out.printf ("Lên?% s \ n", netint.isUp ());

out.printf ("Loopback?% s \ n", netint.isLoopback ());

out.printf ("PointToPoint?% s \ n", netint.isPointToPoint ());

out.printf ("Hỗ trợ phát đa hướng?% s \ n", netint.supportsMulticast ());

out.printf ("Ảo?% s \ n", netint.isVirtual ());

out.printf ("Địa chỉ phần cứng:% s \ n",

Arrays.toString (netint.getHardwareAddress ()));

out.printf ("MTU:% s \ n", netint.getMTU ());

out.printf ("\ n");

}

}

Sau đây là kết xuất mẫu từ chương trình ví dụ:

Tên hiển thị: bge0

Tên: bge0

InetAddress: / fe80: 0: 0: 0: 203: baff: fef2: e99d% 2

InetAddress: /129.156.225.59

Lên? thật

Lặp lại? sai

PointToPoint? sai

Hỗ trợ phát đa hướng? sai

Ảo? sai

Địa chỉ phần cứng: [0, 3, 4, 5, 6, 7]

MTU: 1500

Tên hiển thị: lo0

Tên: lo0

InetAddress: / 0: 0: 0: 0: 0: 0: 0: 1% 1

InetAddress: /127.0.0.1

Lên? thật

Lặp lại? thật

PointToPoint? sai

Hỗ trợ phát đa hướng? sai

Ảo? sai

Địa chỉ phần cứng: null

MTU: 8232

**MỤC LỤC**

[**MỤC LỤC**](file:///C:\Users\Admin\Downloads\Buồn%20ngủ%20vl.doc#page5)[**i**](file:///C:\Users\Admin\Downloads\Buồn%20ngủ%20vl.doc#page5)

[**MỞ ĐẦU**](file:///C:\Users\Admin\Downloads\Buồn%20ngủ%20vl.doc#page7)[**1**](file:///C:\Users\Admin\Downloads\Buồn%20ngủ%20vl.doc#page7)

1. Java Netwoking
2. [TRUY CẬP CÓ LẬP TRÌNH VÀO CÁC THÔNG SỐ MẠNG 2](#_Toc59557692)

[1. Giao diện mạng là gì? 3](#_Toc59557693)

[2. Truy xuất Giao diện Mạng 4](#_Toc59557694)

[3. Liệt kê địa chỉ giao diện mạng 5](#_Toc59557695)

[4. Tham số giao diện mạng 6](#_Toc59557696)