

Socket

Java Collections
10 уровень, 7 лекция

ОТКРЫТА

— Привет, Амиго!

— Здорово, Риша!

— Тема сегодняшней лекции – Сокеты. Socket в переводе с английского – Розетка.

Ты уже знаешь, что у каждого компьютера в сети есть его уникальный IP-адрес.

— Ага.

— И вот представь, что у тебя есть несколько компьютеров и на каждом компьютере запущено с десяток программ, которые работают с интернетом: Skype, ICQ, и т.д.

И эти программы хотят общаться между собой.

Надо сделать так, чтобы они друг другу «не мешали». Чтобы Skype связывался со Skype’ом, ICQ с ICQ и т.д.

Помнишь, как это проблему решили с URL и веб-серверами?

— Ага, добавили порты.

— Именно.

Это все равно, что в доме сделать много маленьких комнат и объявить его многоквартирным. **Каждый порт – это как отдельная квартира.**

Так вот IP-адрес – это уникальный номер компьютера, а **IP-адрес+порт – это уникальный номер некой «квартиры» (ячейки) в компьютере, которую может занять программа.**

Вот такая уникальная ячейка и называется сокет.

У сокета есть его уникальный номер, и **состоит он из IP-адреса и номера порта.**

— Ага. Т.е. сокет – это номер некой виртуальной ячейки в компьютере, которую может занимать программа? А другая программа будет слать ей в эту ячейку сообщения, и таким образом они будут общаться?

— Не знаю, как ты это понял, но именно так и есть.

— Это моя робоинтуиция подсказала.

— Отлично. Тогда давай немного подробностей.

Сокеты – это фактически самый базовый и самый примитивный способ межсетевого взаимодействия программ.

В Java для работы с сокетами есть два класса. Это классы **Socket** и **ServerSocket**.

ServerSocket – это специальный класс, объекты которого выполняют роль сервера – т.е. могу обслуживать запросы, пришедшие на определенный сокет.

Класс **Socket** – это фактически Socket-клиент, с помощью него мы можем послать сообщение некоторому сокету и получить ответ.

Как послать сообщение какому-то сокету:

Пример

```
1 //создаем сокет
2 Socket clientSocket = new Socket("localhost", 4444);
3
4 //получаем OutputStream
5 OutputStream outputStream = clientSocket.getOutputStream();
6 PrintWriter out = new PrintWriter(outputStream, true);
7 out.println("Kiss my shiny metal ass!");
8 out.flush();
9
10 //читаем ответ
11 InputStream inputStream = clientSocket.getInputStream();
12 BufferedReader in = new BufferedReader(new InputStreamReader(inputStream));
13 String answer = in.readLine();
```

— Все очень напоминает скачивание файла из интернета.

— Это потому, детка, что там тоже используются сокеты.

Сокеты используются в основе всего связанного с сетью, ну или почти всего.

Вот [тут](#) можешь почитать дополнительную информацию.

— Спасибо за лекцию, Риша.

— Это еще не все. Размечтался.

Теперь мы разберемся, как работает серверный сокет.

Он работает чуток посложнее.

Пример

```
1 //создаем объект сервер-сокет
2 ServerSocket serverSocket = new ServerSocket(4444); //порт
3
4 //в цикле обрабатываем входящие соединения.
5 while (true)
6 {
7     //метод ассерт ждет, пока кто-то не подключится.
8     Socket socket = serverSocket.accept();
9
10    //читаем сообщение
11    InputStream inputStream = socket.getInputStream();
12    BufferedReader in = new BufferedReader(new InputStreamReader(inputStream));
13    String message = in.readLine();
14
15    //придумываем ответ – просто разворачиваем строку задом наперед
16    String reverseMessage = new StringBuilder(message).reverse().toString();
17
18    //отправляем ответ
19    OutputStream outputStream = socket.getOutputStream();
20    PrintWriter out = new PrintWriter(outputStream, true);
21    out.println(reverseMessage);
22    out.flush();
23 }
```

Хочу обратить твое внимание на несколько моментов.

Момент 1: Для создания сокета нужно указывать IP-адрес (или домен) и порт. Для создания серверного сокета – только порт. Серверный сокет появляется только на том компьютере, где его создали.

Момент 2: У класса `ServerSocket` есть метод `ассепт()`, который, если его вызвать, будет ждать входящее соединение. Т.е. метод будет выполняться бесконечно долго, пока какой-то клиентский сокет не попробует обратиться к нему. Тогда метод `ассепт()` примет соединение, создаст объект сокет для коммуникации и после этого вернет этот объект.

С точки зрения Java-программиста, сокет – это два потока – `InputStream` из которого можно читать сообщения/данные и `OutputStream`, куда можно писать сообщения/данные.

Когда ты создаешь серверный сокет, фактически появляется порт, к которому могут подключаться сокеты с других компьютеров. Но для этого им надо правильно указать номер порта нашего сокета и IP-адрес нашего компьютера. Ну, или его доменное имя.

Вот тебе интересный пример, можешь попробовать разобрать его и выполнить:

<http://habrahabr.ru/post/69136/>

Там весь смысл в том, что с помощью серверного сокета пишется примитивнейший веб-сервер, к которому можно просто обращаться из браузера.

— Ого! Веб-сервер? Круто! Я изучу его очень внимательно.

Спасибо, Риша.

— Все, Амиго, иди отдыхать!

 +24 

Комментарии (24)

популярные

новые

старые

JavaCoder

Введите текст комментария

Igor Petrashevsky

Уровень 47

31 августа, 01:44

...

:facepalm: а вот и снова сокетами поводили по губам

Ответить

 +1 



LuneFox

Уровень 41, Москва, Россия

EXPERT

1 апреля, 00:32

...

Смог обмануть клиент приложения OwnCloud (облачное хранилище), подсунув ему из сокета ответ, который он хочет слышать от настоящего сервера. Узнал я его, попытавшись подключиться к серверу из примера и выведя запрос на экран, а затем выполнив его из браузера (GET /status.php).

Он даже не ругнулся, а предложил продолжить вход.

При нажатии на Click here сокет получил дальнейший запрос:

1	GET /index.php/settings/personal?sectionid=security HTTP/1.1
2	Host: localhost:8082

Теперь я хотя бы примерно представляю, как это работает :) То есть, чтобы создать с нуля свой сервер,

нужно читать заголовки запросов и реагировать на них определённым образом, а неизвестные запросы отклонять.

Ответить

0

Igor PetrashevskyУровень 47

31 августа, 01:47

http, ftp, snmp, pop3, smtp - это все надстройки над древнейшим протоколом telnet , который появился до tcp/ip и умел только в текст.
Сделаете telnet - сделаете все остальное, путем добавки/смены костылей. Это полезно в учебных целях, т.к. для практики есть готовые фрэймворки со всеми необходимыми велосипедами.

Ответить

0

Fomka_WyvernoУровень 41, Киев, Украина

27 ноября 2021, 22:14

Отсылка на Бэндера? out.println("Kiss my shiny metal ass!");

Ответить

0

ArsУровень 41

5 декабря 2021, 13:19

Тут весь курс - отсылка на Футураму. А точнее раньше все персонажи были оттуда. Но потом переделали.

Ответить

+3

Valua SinicynУровень 41, Харьков, Украина

4 марта 2021, 13:32

[Сокеты](#) на человеческом.

Ответить

+4

ЖеняУровень 41, Минск

23 февраля 2021, 10:59

"— *Не знаю, как ты это понял, но именно так и есть*".
Наверно, если бы у валидатора было сознание, он бы то же самое после успешной валидации сообщал.

Ответить

+13

Alexey PrilesskyУровень 40, Минск

15 декабря 2020, 19:44

Интересно про это читать, когда уже в Multithreading Написали свой онлайн чат)

Ответить

+8

RomanBackend Developer в **Yandex**

22 февраля 2021, 13:12

То чувство, когда сначала пошел в коллекции

Ответить

0

KonstantinУровень 41, Одесса

28 сентября 2020, 10:11

"http://habrahabr.ru/post/69136/" - шёл 2020 год

Ответить

+1

ВладиславBackend Developer

5 августа 2020, 21:21

По крупицам, со временем, то, что было не понятно еще уровнем 5 или 8 назад, то, что казалось абсолютной дичью, начинает выстраиваться в целостную картину. И так с любой максимально не понятной по-началу темой. Спасибо создателям за такие уроки.

Ответить

+6

ДворникУровень 41, Санкт-Петербург, Россия

9 апреля 2020, 12:05

http://www.quizful.net/post/java-socket-programming помогло. А статья на хабре ничего не поясняет, что и почему, и как должно быть.

Ответить

+6

Anonymous #2413783Java Developer

24 декабря 2020, 18:18

Спасибо за ссылку, осознанно переписал, почитал комменты, запустил у себя на компе. Только после этого пришло понимание

Ответить

+1

Svetlana VydrinaУровень 41

26 июня, 18:16

Ссылка не работает (

Ответить

0

ДаниилSalesforce Developer в **Viseven**MASTER

29 октября 2019, 23:25

По примеру с хабра запустил свой сервер, но так и не понял как сделать так что бы с других IP адресов получать к нему доступ. Пробовал узнать свой IP и с других устройств (на на своём Wi-Fi) заходить на этот сервер - ничего не получилось. Может кто знает как это сделать?

Yuriy

Уровень 41, Санкт-Петербург, Россия

EXPERT

30 октября 2019, 14:09

Если серверу и клиент находятся в одной локальной сети, то на сервере необходимо разрешить такие подключения в настройках firewall. Если клиент находится за пределами локальной сети, то дополнительно необходимо настроить переадресацию на роутере (открыть и пробросить порт).

Ответить

+2

Dinar

Уровень 41, Уфа

9 ноября 2019, 21:18

Да, на примере Windows 7: в "Центре управления сетями и общим доступом" необходимо выбрать расположение для своей сети "Домашняя сеть"

Ответить

+2

Показать еще комментарии

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

