

Rails from the Ground Up!

with Jared Richardson
RoleModel Software

Networking

Part One

Talking over the Wire

We have . . .

A working console app

But that's boring

Let's play another machine!

Ruby Sockets

Abstracted

Powerful

Open 2 irb sessions

In both sessions

```
require 'socket'  
host = "127.0.0.1"  
port = "9999"
```

Server Sockets

```
server = TCPServer.open(port)  
socket = server.accept  
while line = socket.gets  
  puts line  
end
```

Client Socket

```
s = TCPSocket.open(host, port)  
s.puts("Hello world")
```

War v2.0

Write a server

Write a client

War v2.0 Server

`ruby war-server.rb <port> [<hands_to_play>]`

`hands_to_play` is optional

default to infinity

War v2.0 Client

```
ruby war-client.rb <port> [<hands_to_play>]
```

Then Play Each Other

War v3.0

A threaded server

Handles multiple clients

```
server = TCPServer.open(port)

loop do

  s = server.accept

  # start a thread to handle the client
  Thread.start do

    while line = s.gets

      puts line.chomp + " from #{s.addr[3]}"

    end
  end
end
```

War v3.0

ruby war-server.rb <port> [<hands_to_play>]

ruby war-clients.rb <port> [<hands_to_play>]
[<threads_to_play>]

More on Sockets

```
s=TCPSocket.new(Socket::AF_INET,Socket::SOCK_STREAM, 0)  
opt=[1, 0].pack("i,i")  
s.setsockopt(Socket::SOL_SOCKET, Socket::SO_RCVTIMEO, opt)
```

Whut?

Ruby sockets accept most C options

Native implementation

Very fast

Stopped Here