

# 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

# Reminder . . .

If you party on a class night, having  
a hangover is no excuse for being  
late!

# The Koans

Lessons or questions?

# Re-Cap

Arrays & Hashes

Deck of cards (objects)

Rake

Networking

# Your Turn

Code a simple web server

# Web Server

Listens on port 80

Reads document request (e.g. index.html)

Reads files from the "HTML" directory

Passes contents to the client

Threaded

Test with your web browser

fix the web server example. it doesn't run. :)

# Hint . . .

<http://www.tuxradar.com/content/code-project-create-web-server-ruby>

```
require 'socket'

webserver = TCPServer.new('127.0.0.1', 7125)

while (session = webserver.accept)

    session.print "HTTP/1.1 200/OK\r\nContent-type:text/html\r\n\r\n"

    request = session.gets

    trimmedrequest = request.gsub(/GET\ \//, '').gsub(/\ HTTP.*/, '')

    filename = trimmedrequest.chomp

    if filename == ""

        filename = "index.html"

    end

    begin

        displayfile = File.open(filename, 'r')

        content = displayfile.read()

        session.print content

    rescue Errno::ENOENT

        session.print "File not found"

    end

    session.close

end
```

# Testing?

How do we test web applications?

# runit

put a known file on disk

socket connection

ask for the file

check the contents

write it

# Your Turn

Write a time server

Threaded

Tested

“Raw” time on port 123

HTML page on port 80

Serve time UTC time

[http://en.wikipedia.org/wiki/Network\\_Time\\_Protocol](http://en.wikipedia.org/wiki/Network_Time_Protocol)

# Next...

Networked apps with Sinatra