The Internet
Chris Piech
CS106A, Stanford University
I came here to learn to program the internet...
For the third time ever in CS106A:
Learning Goals

1. Write a program that can make internet requests
2. Write a program that can respond to internet requests
How does your phone communicate with Facebook?
The Java program on your phone talks to the Java program at Facebook
* Android phones run Java. So do Facebook servers
Is this legit?

piech@cs.stanford.edu

piech@cs.stanford.edu is now logged in
Send me the full name for piech@cs.stanford.edu

Chris Piech

“Chris Piech”
Send me the **cover photo** for piech@cs.stanford.edu
Send the **profile photo** for piech@cs.stanford.edu
Send the status for piech@cs.stanford.edu

Face Book Server

“chillin”
Set the status for piech@cs.stanford.edu to be "lecturing".
Send me the status for piech@cs.stanford.edu

Face Book Server

“lecturing”
The internet is just many programs sending messages (as Strings)

Thanks Nick for the teaching YEAH
Background: The Internet

The internet is just many programs sending messages (as Strings)

Facebook datacenter ➔ Your computer (facebook.com)

Thanks Nick for the teaching YEAH
Background: The Internet

The internet is just many programs sending messages (as Strings)

Thanks Nick for the teaching YEAH
Background: The Internet

The internet is just many programs sending messages (as Strings)

Thanks Nick for the teaching YEAH
Background: The Internet

The internet is just many programs sending messages (as Strings)

Facebook datacenter

“Server”

“Get status for “Julia Daniel””

“request”

“Enjoying lecture”

“response”

Your computer

(facebook.com)

“Client”

Thanks Nick for the teaching YEAH
There are two types of internet programs. Servers and Clients
Internet 101
Servers are computers (running code)
Facebook’s closest datacenter is here
The Internet

Face Book Server

Java
Piech, CS106A, Stanford University

Get status for piech@cs.stanford.edu
The Internet

Face Book Server

Java

Status: Chris is chillin

Status: Chris is lecturing
Chris Piech

Face Book Server

The Internet

teaching
The Internet

Face Book Server

Java

Status: Chris is chillin

Status: Chris is lecturing
Face Book Server

The Internet
Many computers can connect to the same server
The Internet

Facebook datacenter

“Server”

Tate Ole Keko’s computer (facebook.com)

“Client”

Chris’ phone (facebook app)

“Client”

REQUEST

RESPONSE

REQUEST

RESPONSE

Your mom’s computer (linux shell)

“Client”

REQUEST
Most of the Internet

Server / Clients

Aka “the backend”
Aka “the cloud”
Aka “the brains”
Aka “the frontend”
Aka “the GUI”
There are two types of internet programs. Servers and Clients
First, the server
A Server’s Simple Purpose

Request
someRequest

Server

String
serverResponse

Java
A Server’s Simple Purpose

Request
someRequest

String
serverResponse

ChatServer

Starting server on port 8080...
getMsgs
newMsg
Added new message
getMsgs
Returned 1 messages
getMsgs
Returned 1 messages
newMsg
Added new message
getMsgs
Return 1 messages
getMsgs

Piech, CS106A, Stanford University
Servers on one slide

1. `public String requestMade(Request request) {
   // server code goes here
}

2. // make a Server object
   private SimpleServer server
       = new SimpleServer(this, 8000);

3. `public void run()
   { // start the server
      server.start();
   }`
public String requestMade(Request request) {
    // server code goes here
}

// make a Server object
private SimpleServer server
    = new SimpleServer(this, 8000);

public void run(){
    // start the server
    server.start();
}
What is a Request?

```java
/* Request has a command */
String command;

/* Request has parameters */
HashMap<String,String> params;

Request request

// methods that the server calls on requests
request.getCommand();
request.getParam(key); // returns associated value
```
Requests are like Remote Method Calls

Server has a bunch of discrete things it can do

makeToast

blendIngredients
Requests are like Remote Method Calls

Server has a bunch of discrete things it can do

getStatus

addUser
Requests are like Remote Method Calls

getStatus

addUser
Requests are like Remote Method Calls

```java
request.getCommand();
=> "getStatus"
```

Server

getStatus

addUser
Requests are like Remote Method Calls

To make toast, I need a parameter which is the kind of bread

getStatus
Requests are like Remote Method Calls

I was given a parameter!

getStatus
Requests are like Remote Method Calls

```java
request.getParam(“userName”) // getParam
getStatus
```
Requests are like Remote Method Calls

getStatus
Requests are like Remote Method Calls

cpiech

getStatus
Requests are like Remote Method Calls
public String requestMade(Request request) {
    String cmd = request.getCommand();
    if (cmd.equals("getStatus")) {
        String user = request.getParam("userName");
        String status = runGetStatus(user);
        return status;
    }
    ...
}

.toString()???

.toString()???
Requests are like Remote Method Calls
Requests are like Remote Method Calls
Requests are like Remote Method Calls
Requests are like Remote Method Calls
Requests are like Remote Method Calls
Requests are like Remote Method Calls
What is a Request?

/* Request has a command */
String command;

/* Request has parameters */
HashMap<String,String> params;

Request request

// methods that the server calls on requests
request.getCommand();
request.getParam(key);  // returns associated value
public String requestMade(Request request) {
    // server code goes here
}

// make a Server object
private SimpleServer server = new SimpleServer(this, 8000);

public void run(){
    // start the server
    server.start();
}
A Server’s Simple Purpose

1. public String requestMade(Request request) {
   // server code goes here
}

2. // make a Server object
private SimpleServer server
   = new SimpleServer(this, 8000);

3. public void run(){
   // start the server
   server.start();
}
What is a Port?
public String requestMade(Request request) {
    // server code goes here
}

// make a Server object
private SimpleServer server
    = new SimpleServer(this, 8000);

public void run(){
    // start the server
    server.start();
}
Echo Server

Your command was 11 chars long.
Echo Server

Request
Any Request

String
Length of the cmd

Starting server...
Request received hello
Request received this+is+a+test
Request received whatsGood
Request received ping
Request received ping
Request received ping
Request received pong
Request received ping
There are two types of internet programs. Servers and Clients
Then, the client
A Clients’s Purpose

1. Interact with the user
2. Get data from its server
3. Save data to its server
try {

    // 1. construct a new request
    Request example = new Request("getStatus");

    // 2. add parameters to the request
    example.addParam("name", "chris");

    // 3. send the request to a computer on the internet
    String result = SimpleClient.makeRequest(HOST, example);

} catch (IOException e) {

    // The internet is a fast and wild world my friend

}
try {
    // 1. construct a new request
    Request example = new Request("getStatus");

    // 2. add parameters to the request
    example.addParam("name", "chris");

    // 3. send the request to a computer on the internet
    String result = SimpleClient.makeRequest(HOST, example);
} catch(IOException e) {
    // The internet is a fast and wild world my friend
}
try {

    // 1. construct a new request
    Request example = new Request("getStatus");

    // 2. add parameters to the request
    example.addParam("name", "chris");

    // 3. send the request to a computer on the internet
    String result = SimpleClient.makeRequest(HOST, example);

} catch(IOException e) {

    // The internet is a fast and wild world my friend

}
try {

    // 1. construct a new request
    Request example = new Request("getStatus");

    // 2. add parameters to the request
    example.addParam("name", "chris");

    // 3. send the request to a computer on the internet
    String result = SimpleClient.makeRequest(HOST, example);
}

} catch (IOException e) {

    // The internet is a fast and wild world my friend

}
try {

    // 1. construct a new request
    Request example = new Request("getStatus");

    // 2. add parameters to the request
    example.addParam("name", "chris");

    // 3. send the request to a computer on the internet
    String result = SimpleClient.makeRequest(HOST, example);

} catch (IOException e) {

    // The internet is a fast and wild world my friend

}
try {

    // 1. construct a new request
    Request example = new Request("getStatus");

    // 2. add parameters to the request
    example.addParam("name", "chris");

    // 3. send the request to a computer on the internet
    String result = SimpleClient.makeRequest(HOST, example);

} catch(IOException e) {

    // The internet is a fast and wild world my friend

}
try {

    // 1. construct a new request
    Request example = new Request("getStatus");

    // 2. add parameters to the request
    example.addParam("name", "chris");

    // 3. send the request to a computer on the internet
    String result = SimpleClient.makeRequest(HOST, example);

} catch (IOException e) {

    // The internet is a fast and wild world my friend

}
try {

    // 1. construct a new request
    Request example = new Request("getStatus");

    // 2. add parameters to the request
    example.addParam("name", "chris");

    // 3. send the request to a computer on the internet
    String result = SimpleClient.makeRequest(HOST, example);

} catch (IOException e) {

    // The internet is a fast and wild world my friend

}
Time for a little chat
Chat Server and Client

> CJP: Testing
> LMS: Hellooooo
> LMS: I'm online!
> LMS: This is great
> LMS: And this is going to make it into lecture
addMsg
msg = C: Hello world

history = []
history = [
    C: Hello world
]

getMsgs
index = 0
history = [ 
  C: Hello world
]

[C: Hello world]
history = [
    C: Hello world
]
```java
history = [
    C: Hello world,
    B: I'm here too
]

getMsgs
index = 1

Chat Client
> C: Hello world

Chat Client
> C: Hello world
```
history = [
  C: Hello world,
  B: I'm here too
]

[B: I'm here too]
Chat Server

Chat Server

addMsg
msg = text

getMsgs
index = startIndex