

Everything You Wanted to Know

About Writing Async, Concurrent HTTP Apps in Java



Agenda

- Mostly this:



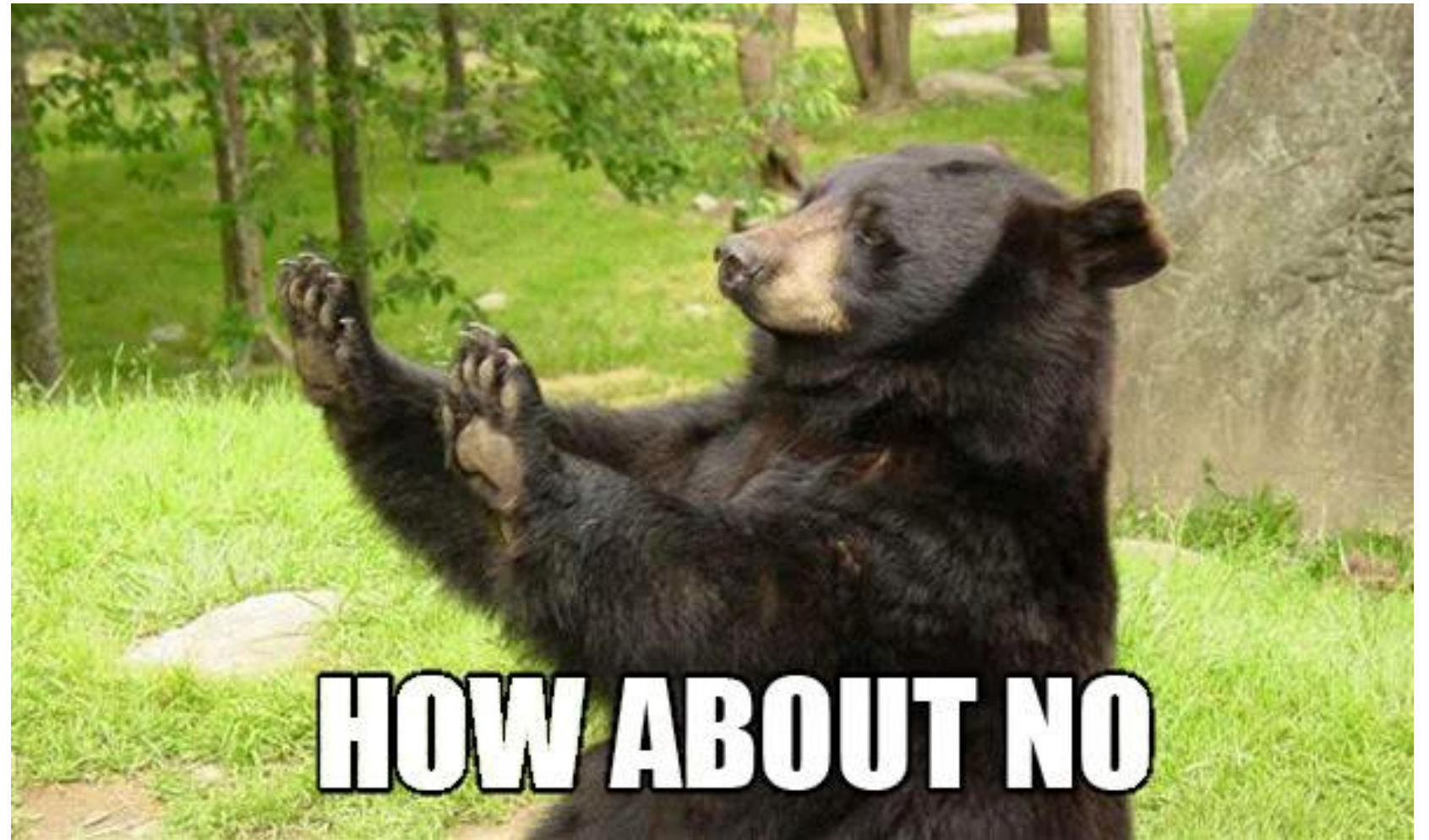
Agenda

- And this:



Agenda

- And this:




About your speaker

@jbaruch

github.com/jbaruch

linkd.in/jbaruch





Baruch Sadogursky
J*, G* and Public Speaking Geek with JFrog FTW.
Israel | Computer Software

Current **Developer Advocate at JFrog Ltd**

stackoverflow.com/users/402053/jbaruch

What Frog?



What Frog?



What Frog?



What Frog?





WE SERVE BINARIES FOR A LIVING

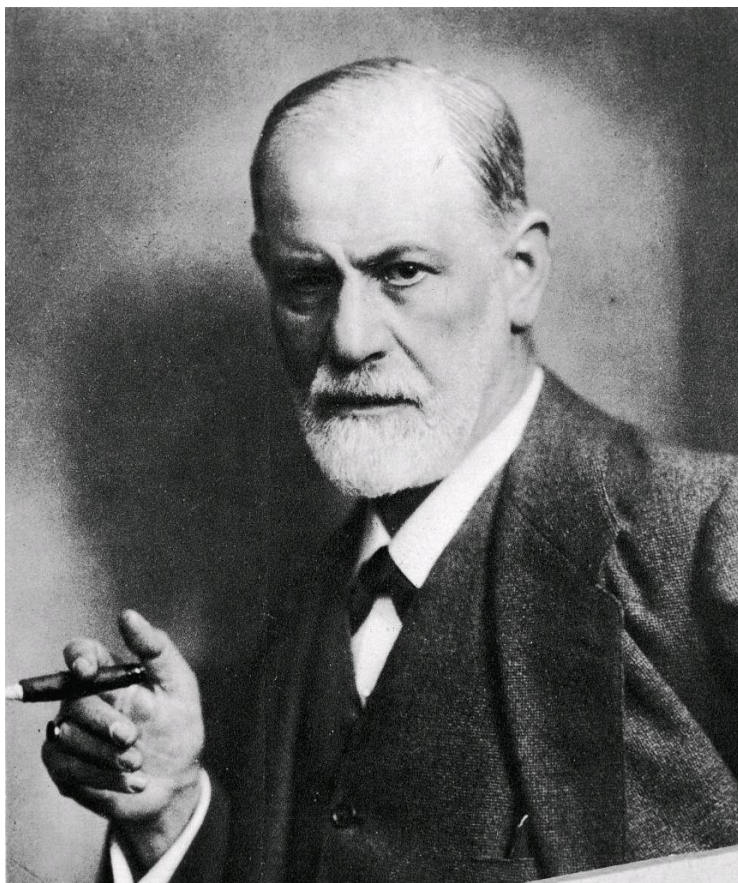
CONCURRENT DOWNLOADS



Y U NO SUPPORT THEM?!

Requirements

- parallel file Downloads
- Parallel file parts
- interrupt/pause/resume
- Progress events
- Checksums caching



First Association for “concurrent
downloader”

**Downloadr -
Download Manager**

★★★★★ (356)



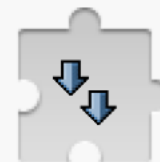
Download Manager

★★★★★ (238)



Download Master

★★★★★ (1118)



**Fruumo Download
Manager**

★★★★★ (185)



**GetThemAll
Downloader**

★★★★★ (687)



Downloads

★★★★★ (1982)



Chrome Downloads

★★★★★ (321)

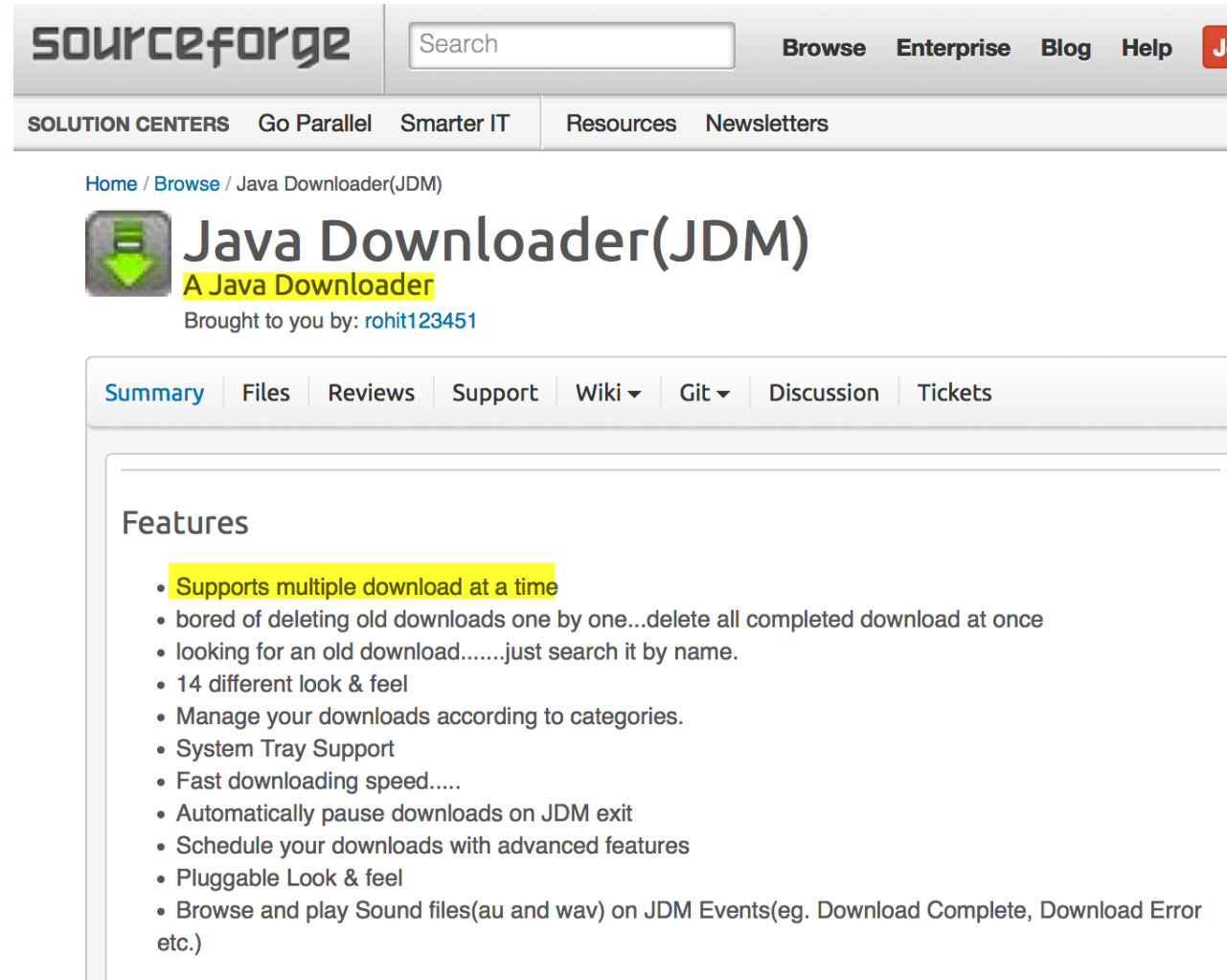


**Video Downloader
professional**

★★★★★ (25352)



Lucky day: Download manager written in Java!



SOURCEFORGE Search [Browse](#) [Enterprise](#) [Blog](#) [Help](#) [Jo](#)

[SOLUTION CENTERS](#) [Go Parallel](#) [Smarter IT](#) [Resources](#) [Newsletters](#)

[Home](#) / [Browse](#) / [Java Downloader\(JDM\)](#)

Java Downloader(JDM)

A Java Downloader

Brought to you by: [rohit123451](#)

[Summary](#) | [Files](#) | [Reviews](#) | [Support](#) | [Wiki](#) | [Git](#) | [Discussion](#) | [Tickets](#)

Features

- **Supports multiple download at a time**
- bored of deleting old downloads one by one...delete all completed download at once
- looking for an old download.....just search it by name.
- 14 different look & feel
- Manage your downloads according to categories.
- System Tray Support
- Fast downloading speed.....
- Automatically pause downloads on JDM exit
- Schedule your downloads with advanced features
- Pluggable Look & feel
- Browse and play Sound files(au and wav) on JDM Events(eg. Download Complete, Download Error etc.)

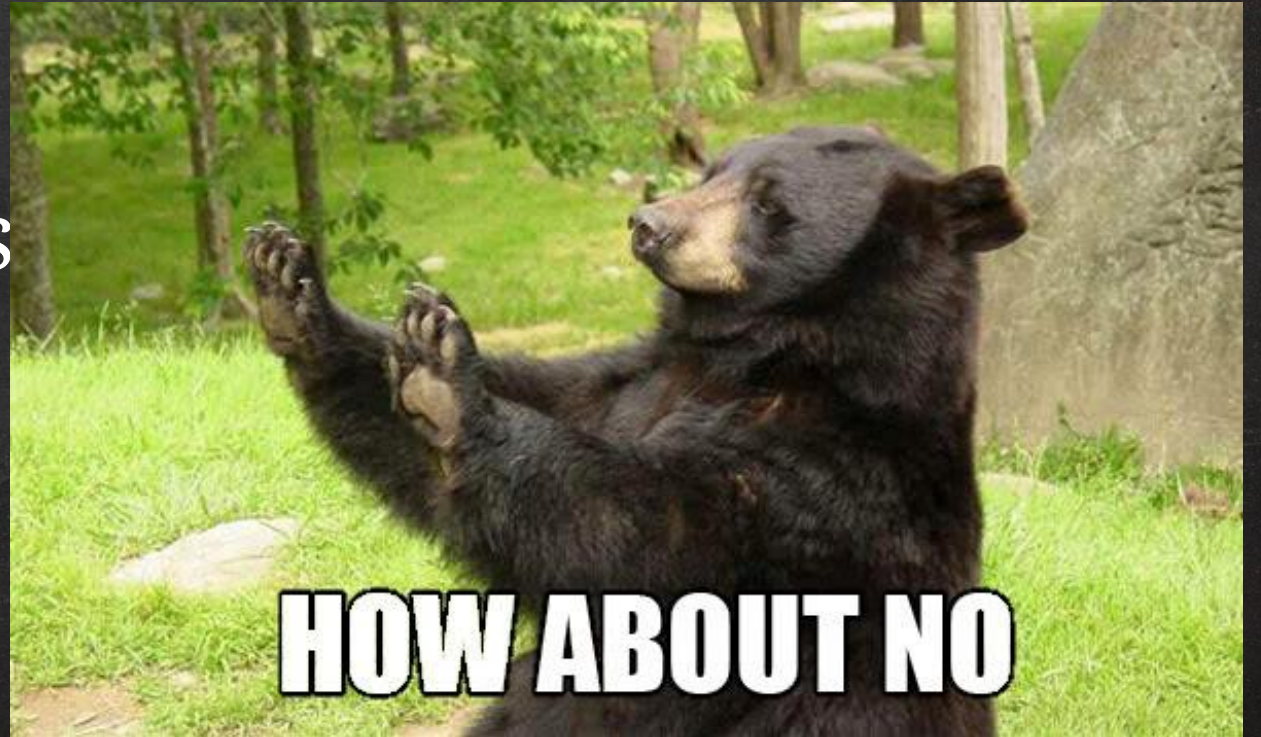
A meme featuring Aragorn from The Lord of the Rings. He is shown from the chest up, looking slightly to the left with a subtle, knowing smile. He has long, wavy brown hair and a light beard. He is wearing a dark, textured tunic. His right hand is raised near his chin, with fingers slightly curled. The background is a dimly lit, stone-walled interior with arches. The text "ONE DOES NOT SIMPLY" is overlaid at the top in a large, white, bold, sans-serif font with a black outline.

ONE DOES NOT SIMPLY

EMBED JDM

Let's look if we can use it!

1. No traceable license
2. No website or docs
3. No traceable sources
4. It's an app, not a lib



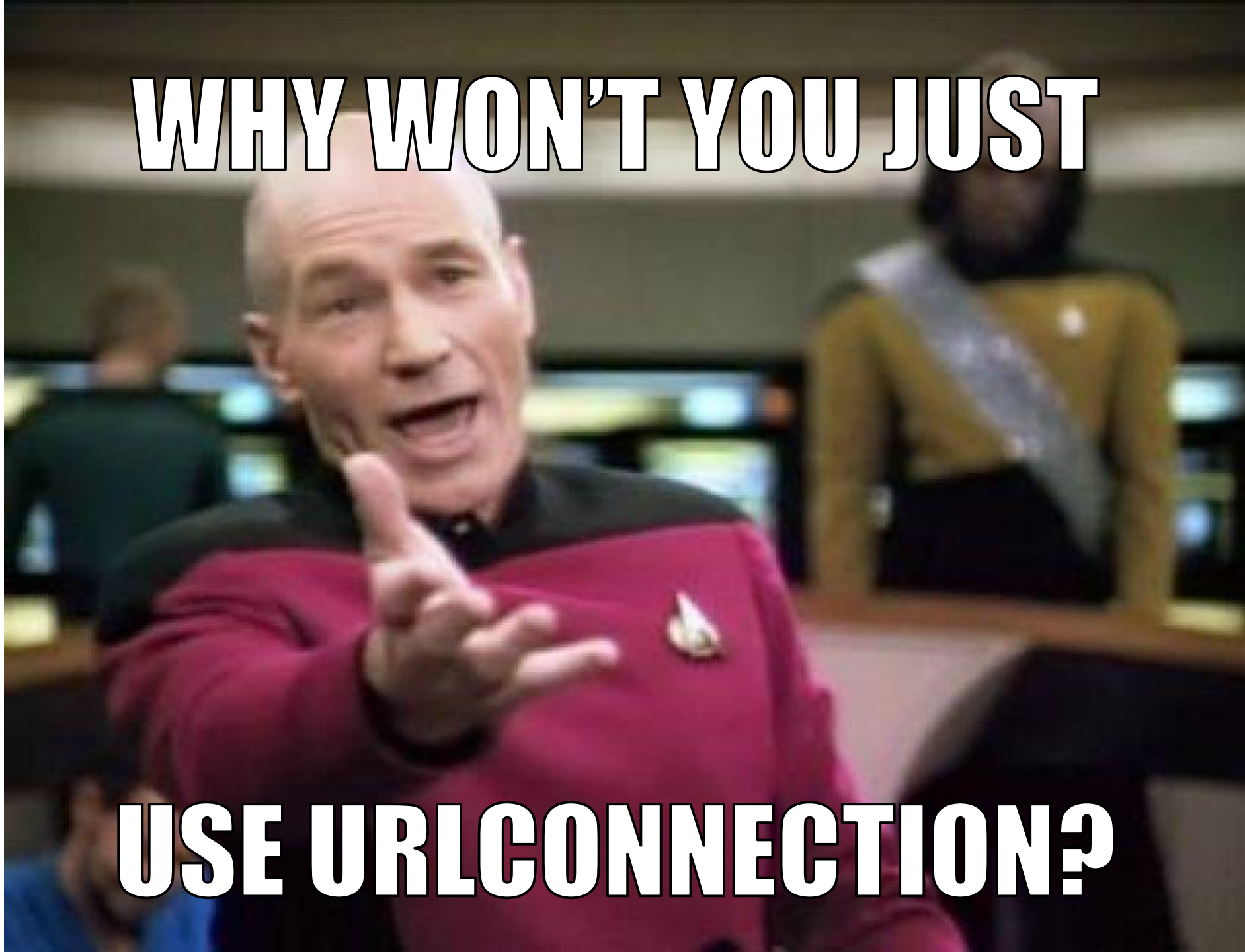
IF YOU WANT CONCURRENT DOWNLOADER



WRITE IT YOURSELF

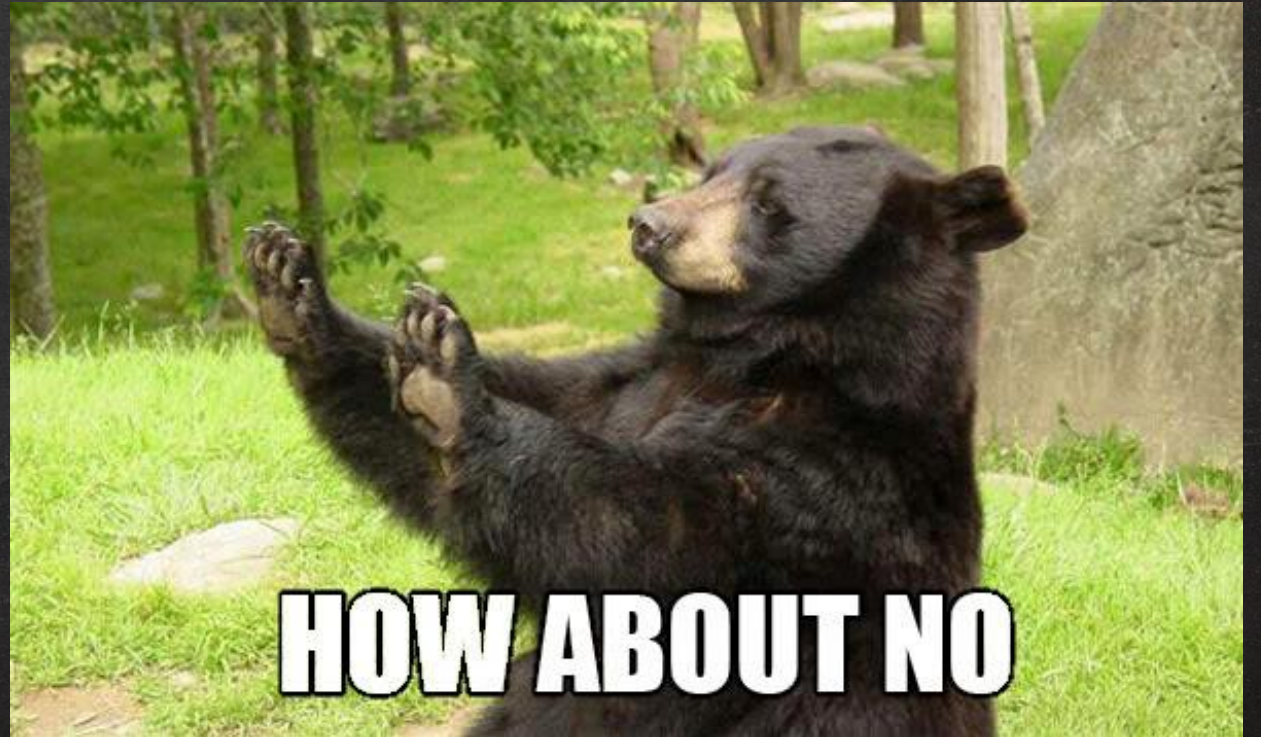
WHY WON'T YOU JUST

USE URLCONNECTION?



Java.net.urlconnection

1. Memory wasteful (buffering)
2. Minimal API
3. Blocking streams





**DO NOT
BLOCK**

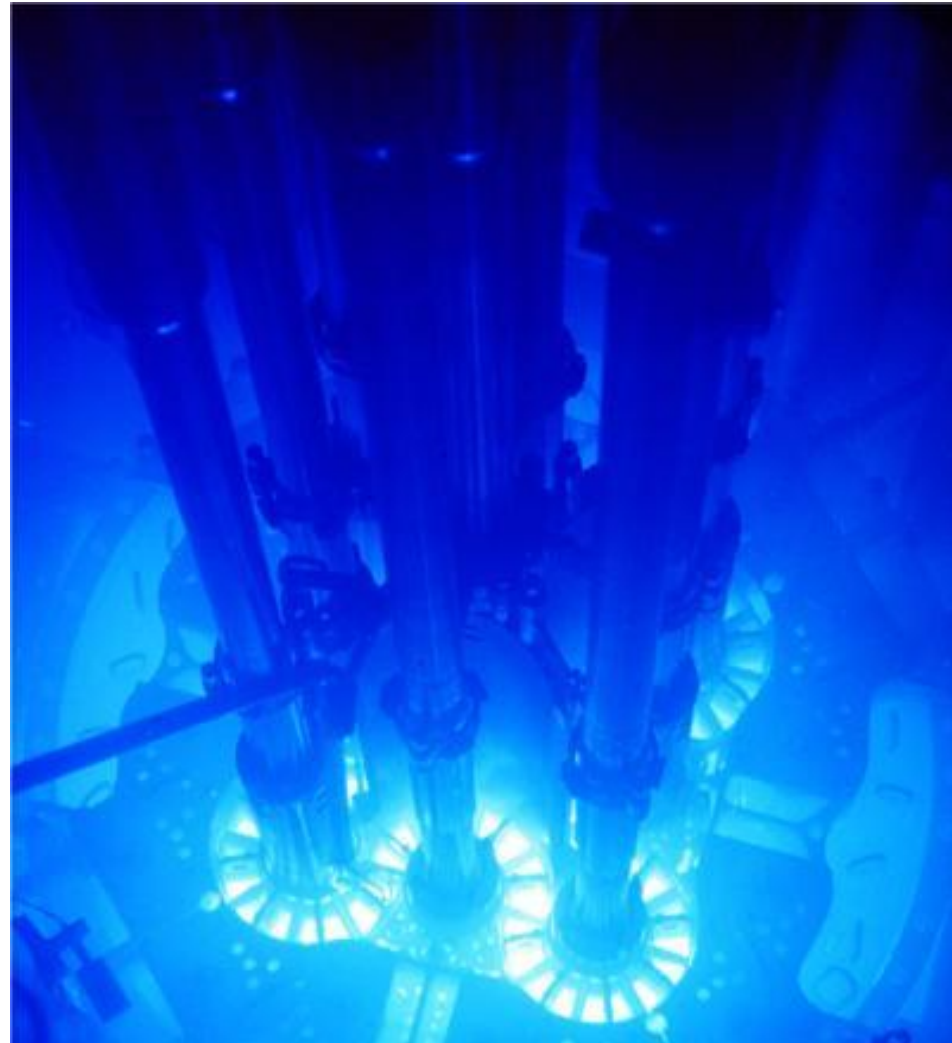
What we're looking for

1. Async/non-blocking
2. Event callbacks

What is IT going to take

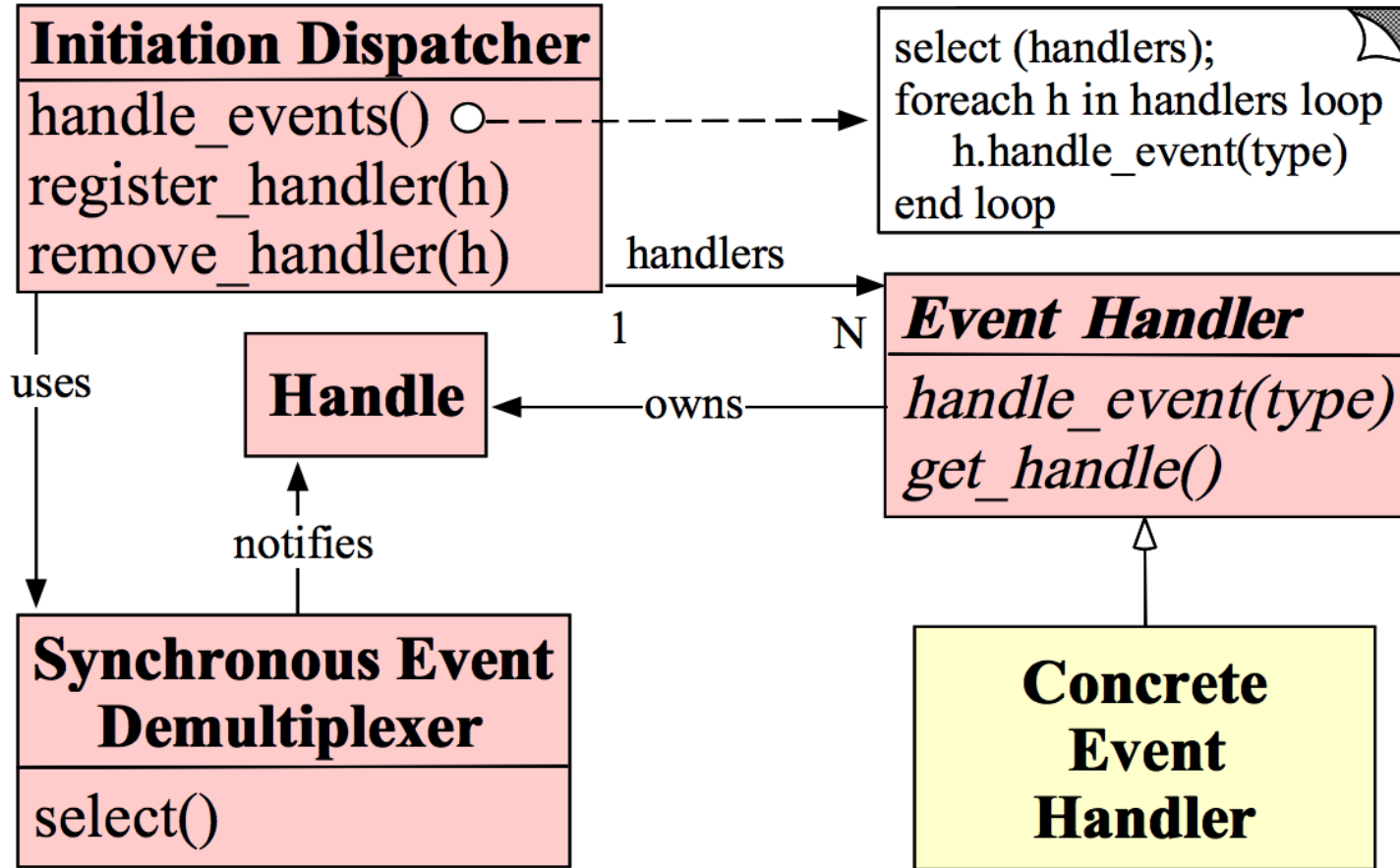
1. Reactor
2. nio

Welcome to the reactor

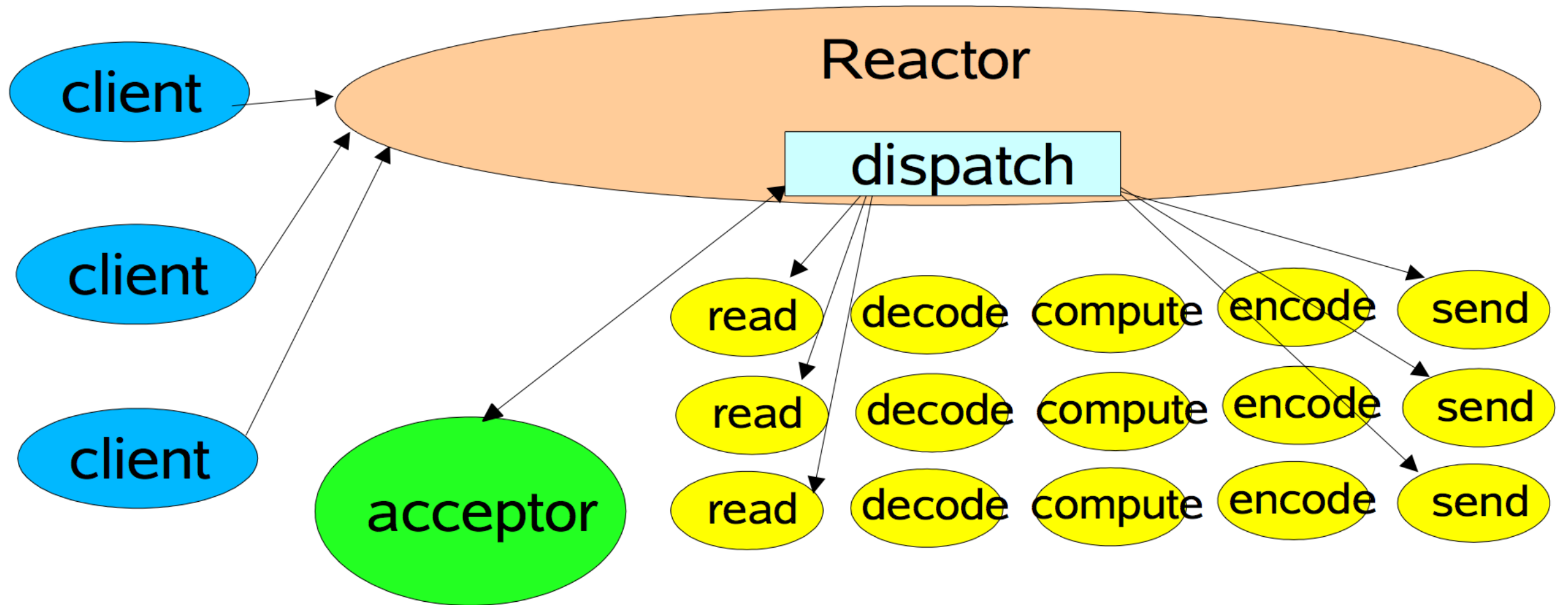


- pattern for lightweight concurrency
- Event driven
- Threads reuse
- Uses non-blocking io

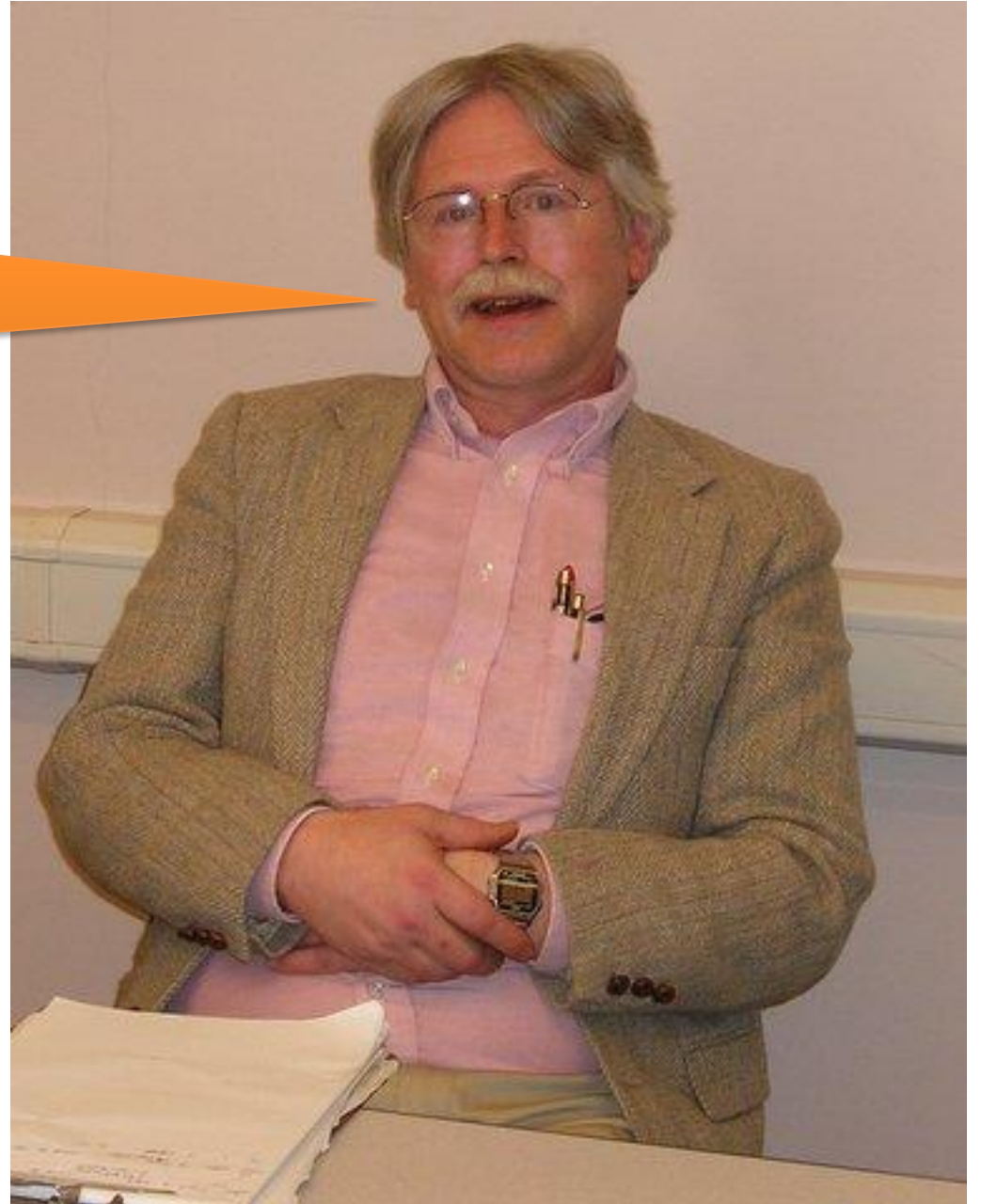
Original pattern



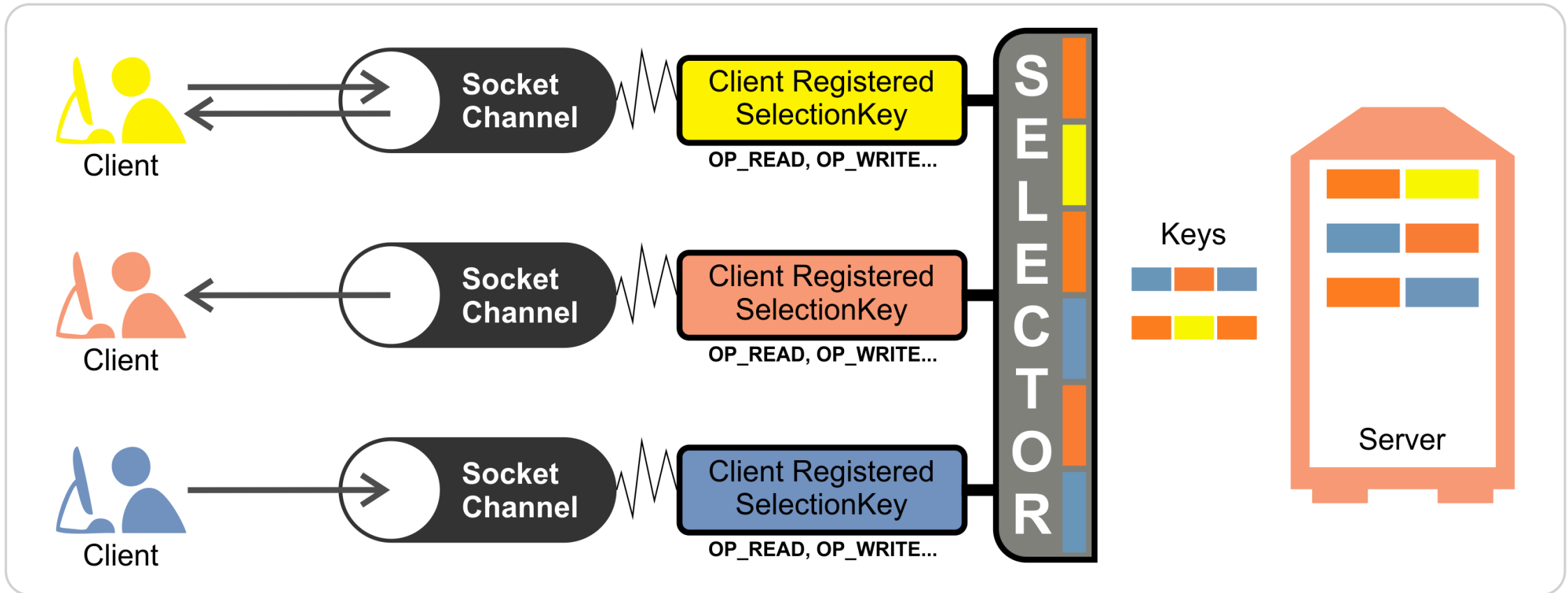
Guess the author by the diagram



In Java, Reactor
means NIO



Selector as a multiplexer



Java version - Registering

```
SocketChannel channel= SocketChannel.open();  
socketChannel.connect(new  
    InetSocketAddress("http://remote.com", 80));
```

...

```
Selector selector = Selector.open();  
channel.configureBlocking(false);  
SelectionKey k = channel.register(selector,  
SelectionKey.OP_READ);  
k.attach(handler);
```

Java version - Dispatcher

```
while (!Thread.interrupted()) {  
    selector.select();  
    Set selected = selector.selectedKeys();  
    Iterator it = selected.iterator();  
    while (it.hasNext())  
        SelectionKey k = (SelectionKey)(it.next());  
        ((Runnable)(k.attachment())).run();  
    selected.clear();  
}
```

Handling reactor events is complex

- Need to maintain state
- Buffering – assembling chunks
- Coordinating async events

HTTP NIO FRAMEWORKS

DO THE HEAVY LIFTING



Nio libraries

- Most of them are servers
 - Netty, grizzly, etc.
- Apache Mina
- Apache HTTP components asyncclient
- Ning http client

Nio libraries

~~— Most of them are servers~~

~~— Netty, grizzly, etc.~~

– Apache Mina

– Apache HTTP components asyncclient

– Ning http client

The logo for Apache MINA, featuring the word "Apache" in a smaller font above the word "MINA" in a larger, bold font. The text is white and set against a black rounded rectangular background. Behind the text are several overlapping, glowing red and purple rings, resembling an atomic structure or a network diagram. A small "TM" trademark symbol is located at the bottom right of the logo.

Apache **MINA**

- Client and server nio library
- Evolved from netty
- Latest release October 2012

Contributors

Commits

Code frequency

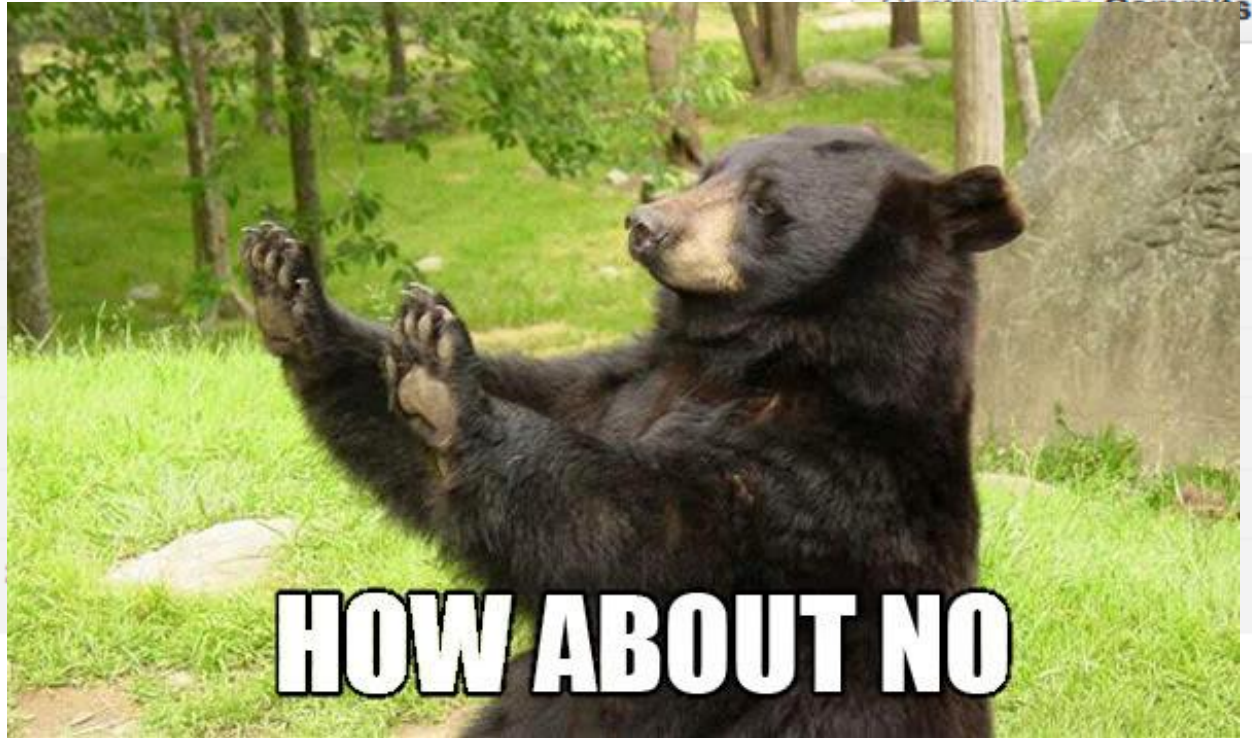
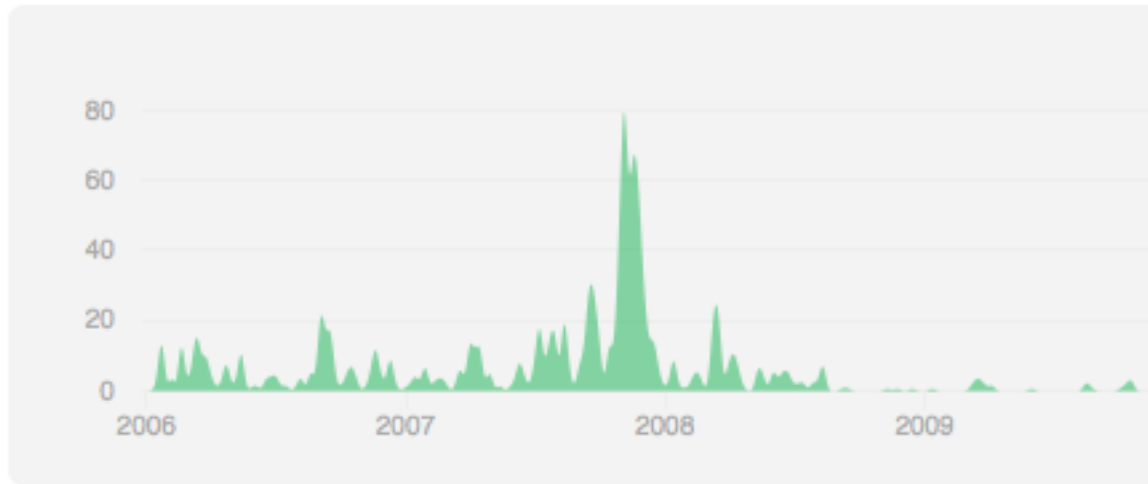
Punch card

Network

Members

Dec 25, 2005 – Sep 21, 2014

Contributions to trunk, excluding merge commits



Nio libraries

~~— Most of them are servers~~

~~— Netty, grizzly, etc~~

~~— Apache Mina~~

– Apache HTTP components asyncclient

– Ning http client

WHAT IS
NING?

NING
MODE MEDIA

CULTIVATING
COMMUNITY

[Pricing](#)

[Login](#) ▶



Build and cultivate your own community of **FOLLOWERS**

Built from the ground up for social, Ning's scalable hosted platform gives you the tools and expertise you need to *publish* and *connect* with your community - all in one place. Easy. Powerful. Affordable.

Ning. Your Brand. Your Members. Your Control.

[get started for](#)

Ning's async http client

INTRODUCING NING'S ASYNCHRONOUS HTTP CLIENT LIBRARY.

Posted by [Jeanfrancois Arcand](#) on March 4, 2010 – 12:42 PM

On **March 3, 2010**, Ning released a new Asynchronous HTTP Client library as open source. Its purpose is to allow Java applications to easily execute HTTP requests and asynchronously process the HTTP responses. You can get it at

<http://github.com/ning/async-http-client>



branch: **master** ▾

async-http-client / +



Merge pull request [#47](#) from markchadwick/patch-1 ⋮



brianm authored on 9 Dec 2011

latest commit [8dd6cf9cff](#)



README

The Google juice on this page is still strong. Here's one less hop.

3 years ago

README

Development has moved to

<https://github.com/sonatype/async-http-client>

Please use that repo, not this one :-)

-Brian

404

This is not the web page you are looking for.



Here it is!

 AsyncHttpClient / **async-http-client**

 Watch ▾ 186

 ★ S

Asynchronous Http and WebSocket Client library for Java


 2,418 commits

 6 branches

 71 releases

 76 contributors




 branch: **master** ▾


async-http-client / +



Revert enforcing compression

 **slandelle** authored 2 days ago

latest commit **b5680187ad** 

 **api**

Revert enforcing compression

2 days ago

 **extras**

Added the library typesafe config to read default config values. The ...

15 days ago

```
try (AsyncHttpClient asyncHttpClient = new AsyncHttpClient()) {  
    ListenableFuture<Response> future = asyncHttpClient.prepareGet(  
        "http://oss.jfrog.org/api/system/ping").execute(  
        new AsyncCompletionHandler<Response>() {  
            @Override  
            public Response onCompleted(Response response) {  
                System.out.println(response.getResponseBody());  
                return response;  
            }  
  
            @Override  
            public void onThrowable(Throwable t) {  
                t.printStackTrace();  
            }  
        });  
    Response response = future.get();  
}
```



HttpComponents

- [Home](#)
- [License](#)
- [Download](#)
- [Mailing Lists](#)
- [Developer documents](#)
- [Wiki \(external\)](#)
- [Security](#)

Overview

- [About](#)
- [News](#)
- [Powered by](#)
- [Get Involved](#)

Components

- [HttpCilent 4.4 \(alpha\)](#)

HttpAsyncClient Overview

The Hyper-Text Transfer Protocol (HTTP) is perhaps the most significant protocol used on the Internet today. Web services, network-enabled appliances and the growth of network computing continue to expand the role of the HTTP protocol beyond user-driven web browsers, while increasing the number of applications that require HTTP support.

Although the `java.net` package provides basic functionality for accessing resources via HTTP, it doesn't provide the full flexibility or functionality needed by many applications. `HttpAsyncClient` seeks to fill this void by providing an efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations.

Designed for extension while providing robust support for the base HTTP protocol, `HttpAsyncClient` may be of interest to anyone building HTTP-aware client applications based on asynchronous, event driven I/O model.

HAC Concepts

- Request producer
- Response consumer

```
try(CloseableHttpClient asyncHttpClient = HttpClientUtils.createDefault()) {
    asyncHttpClient.start();
    Future<HttpResponse> future = asyncHttpClient.execute(
        HttpClientUtils.createGet("http://oss.jfrog.org/api/system/ping"),
        new AsyncByteConsumer<HttpResponse>() {
            @Override
            protected void onResponseReceived(final HttpResponse response) {
                System.out.println(response.getStatusLine().getReasonPhrase());
            }
        }
    );
    @Override
    protected void onByteReceived(final CharBuffer buf, final IOControl ioctrl) {}
    @Override
    protected void releaseResources() {}
    @Override
    protected HttpResponse buildResult(final HttpContext context) {
        return (HttpResponse) context.getAttribute("http.response");
    }
    response = future.get();
}
```



WAIT A SECOND...

THEY ARE ALMOST THE SAME!

Choosing between ning and http asyncclient



"All problems in computer science can be solved by another level of indirection"

David
Wheeler

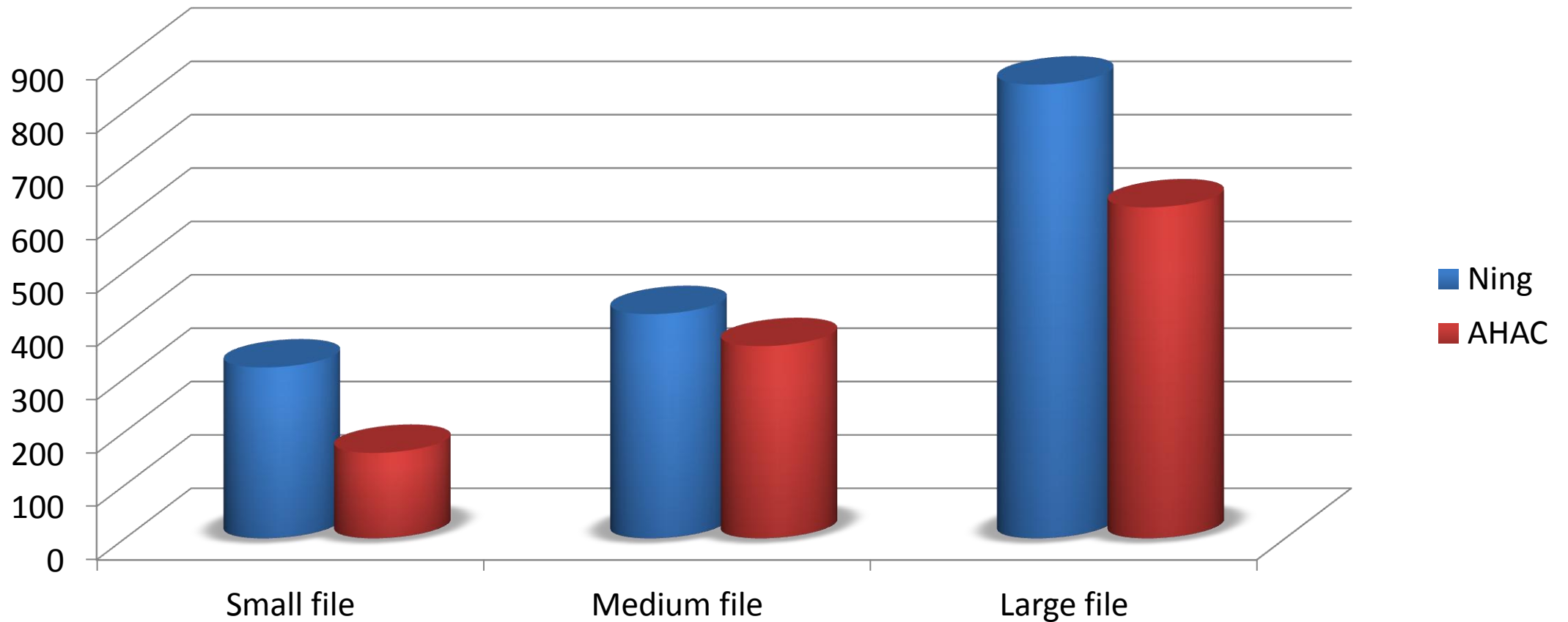


```
public interface HttpProviderDownloadHandler {  
    void onResponseReceived(int statusCode, Map<String, List<String>> headers);  
    boolean onBytesReceived(ByteBuffer buf);  
    void onFailed(Throwable error);  
    void onCancel();  
    void onComplete();  
}
```

Head to head

Feature/Library	Ning client	Http Async Client
-----------------	-------------	-------------------

Performance?





Rfc2616: a universe of its own

Fielding, et al
□
RFC 2616

[Page 6]

June 1999


19.4.1	MI	167
19.4.2	Co	167
19.4.3	Co	168
19.4.4	In	168
19.4.5	No	168
19.4.6	In	169
19.4.7	MH	169
19.5	Addit	169
19.5.1	Co	170
19.6	Compa	170
19.6.1	Ch	171
19.6.2	Co	ons	172
19.6.3	Ch	172
20	Index	175
21	Full C176



Confused?



Just read some stackoverflow (and improve your rep as you go)

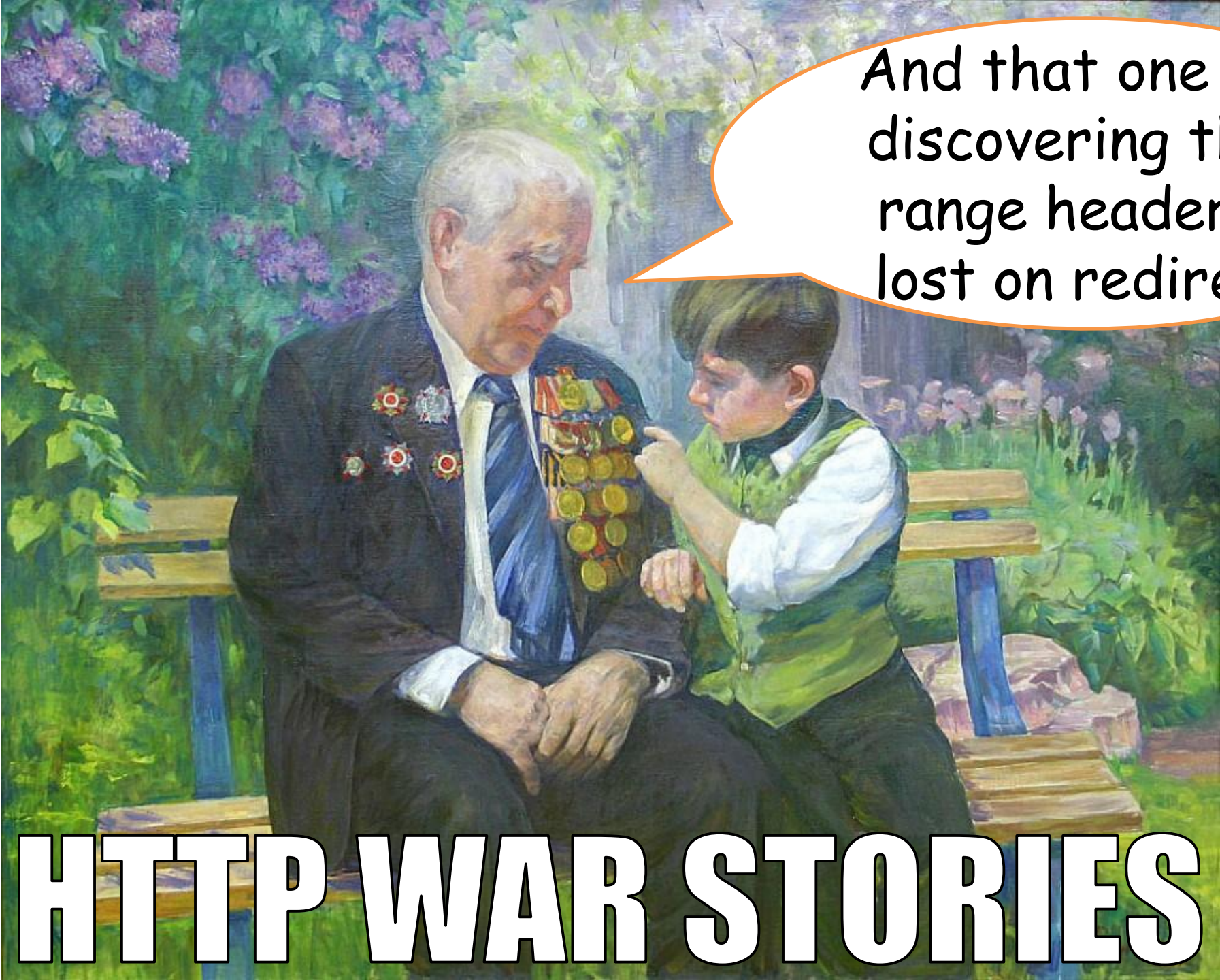
 **stackoverflow** Questions Tags Users Badges Unanswered

Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags and answer your question.

Type to find tags:

http × 25482	https × 8436	http-headers × 5060	xml × 4842
an application level network protocol that is used for the transfer of content on the World Wide Web. 29 asked today, 152 this week	Hypertext Transfer Protocol Secure (HTTPS) is a combination of the Hypertext Transfer Protocol with the SSL/TLS 13 asked today, 65 this week	in the Hypertext Transfer Protocol (HTTP), HTTP header fields contain the operating parameters of an HTTP request or 7 asked today, 34 this week	a s co th 6 a

A painting of an elderly man with white hair, wearing a dark suit and a blue striped tie, sitting on a wooden bench. He has several medals pinned to his lapel. A young boy in a white shirt and green vest is sitting next to him, pointing at the medals. The background is a lush garden with purple flowers and green foliage.

And that one for
discovering that
range header is
lost on redirect

HTTP WAR STORIES



Question!

What should be
content-length
when using
compression?



content-length



9



2

The client is
Encoding: s

What would
data.

Thanks,

http

share | edit

add a comment

start a bounty



with Accept-

e compressed

29 '10 at 6:35

[kwei](#)

2 ● 9

3 Answers

active

oldest

votes



It's the smaller of 1024 or the compressed size.

12

[RFC2616 section 14.11](#) vs:

Content-Length: before or after gzip #46



martinthomson opened this issue on 28 Feb 2013 · 16 comments



martinthomson commented on 28 Feb 2013

Collaborator

Content-Length is largely only needed as entity metadata in HTTP/2.0. It does provide a limited function in learning the complete size of a resource prior to receiving an entire message. (This is the behavior explicitly relied upon for POST, which is based on browser information only. For example, node.js always sends chunked encoding unless explicitly overridden.)

Since compression is applied by the framing layer, there's an ambiguity in the spec with respect to what value Content-Length is given. If the data frames are compressed at the framing layer, the pre-compression size is possibly, but not certainly, the size that is reported in Content-Length.



mcmanus commented on 18 Mar 2013

@mnot re content-length - the other use case is pure http2.. CL enables transfer progress meters (especially on downloads) which are useful ui elements.. so keeping the status quo of it reflecting transfer size is right imo.



grmocg commented on 18 Mar 2013

Collaborator

yes-- content-length's meaning should be unchanged, and still reflect the entity-body size when optionally present.

...

Question!



Why when redirected to CDN
all the chunks start from zero?

```
HttpAsyncClientBuilder builder = HttpAsyncClients.custom();
// add redirect strategy that copies "range" headers, if exist
builder.setRedirectStrategy(new DefaultRedirectStrategy() {
    @Override
    public HttpRequest getRedirect(HttpRequest request, HttpResponse response,
        HttpContext context)
    HttpRequest redirectRequest = super.getRedirect(request, response, context);
    // copy "Range" headers, if exist
    Header[] rangeHeaders = request.getHeaders(HttpHeaders.RANGE);
    if (rangeHeaders != null) {
        for (Header header : rangeHeaders) {
            redirectRequest.addHeader(header);
        }
    }
    return redirectRequest;
}});
```

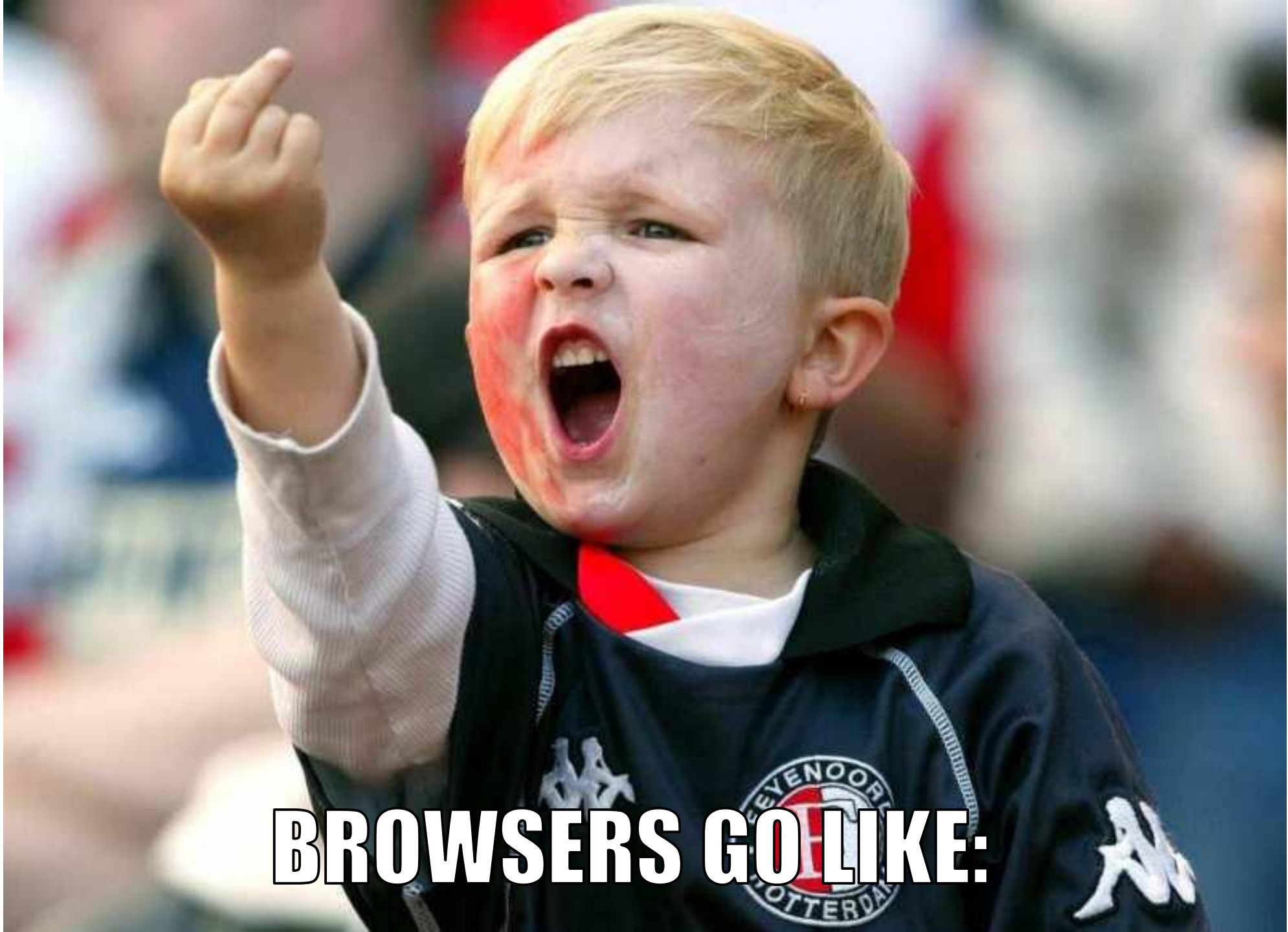

Question!

How many simultaneous connections should I open?



HTTP 1.1 GOES LIKE:

**CAN YOU PLEASE LIMIT
TO 2 CONNECTIONS?**



BROWSERS GO LIKE:

Compare

name	score	PerfTiming	Connections per Hostname	M
<input type="checkbox"/> IE 11 →	12/16	yes		13
<input type="checkbox"/> IE 10 →	12/16	yes		8
<input type="checkbox"/> Android 2.3 →	8/16	no		8
<input type="checkbox"/> Blackberry 7 →	11/16	no		7
<input type="checkbox"/> Chrome 32 →	12/16	yes		6
<input type="checkbox"/> Firefox 26 →	11/16	yes		6
<input type="checkbox"/> IE 9 →	12/16	yes		6
<input type="checkbox"/> Safari 7.0.1 →	11/16	no		6
<input type="checkbox"/> Chrome 34 →	12/16	yes		6
<input type="checkbox"/> Firefox 27 →	11/16	yes		6
<input type="checkbox"/> Android 4 →	13/16	yes		6
<input type="checkbox"/> Chrome Mobile 18 →	12/16	yes		6
<input type="checkbox"/> IEMobile 9 →	11/16	yes		6

Compare Browsers

URL ENCODING?

EASY.

Question!

What's wrong with the following code?



```
public static String encodeUrl(String urlStr) {  
    URLEncoder.encode(urlStr, "UTF-8");  
    ...  
}
```

Decoded URLs cannot be
re-encoded to the same form

<http://example.com/?query=a&b==c>

Cannot be decoded back after it was
encoded:

<http://example.com/?query=a%26b==c>

Don't use `java.net.URLEncoder`

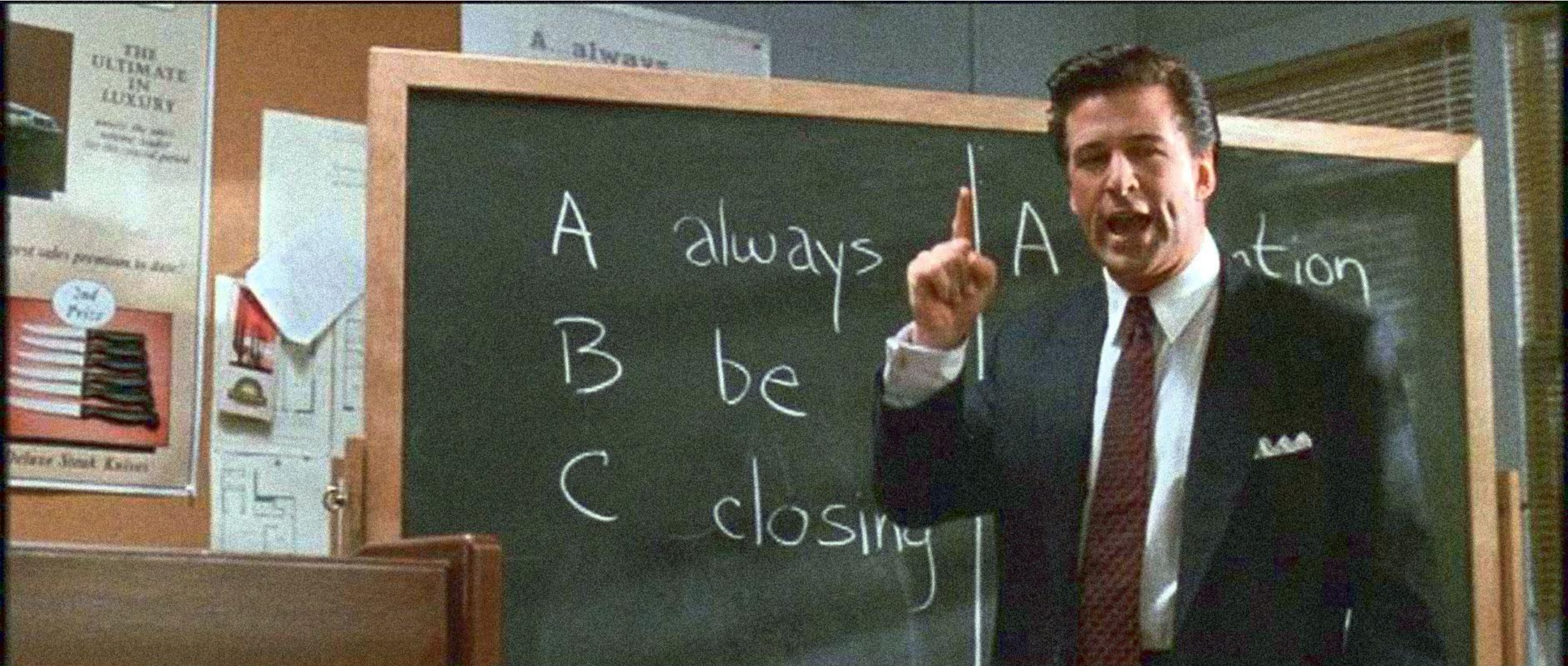
“Utility class for HTML form encoding. This class contains static methods for converting a `String` to the **`application/x-www-form-urlencoded`** MIME format.

For more information about HTML form encoding, consult the HTML specification.”

AHC Alternatives

`org.apache.http.client.utils.URIBuilder`

`org.apache.http.client.utils.URLEncodedUtils`



A always | A
B be
C closing

A ntion

THE ULTIMATE IN LUXURY

A. always

Best value premium to date

2nd Prize

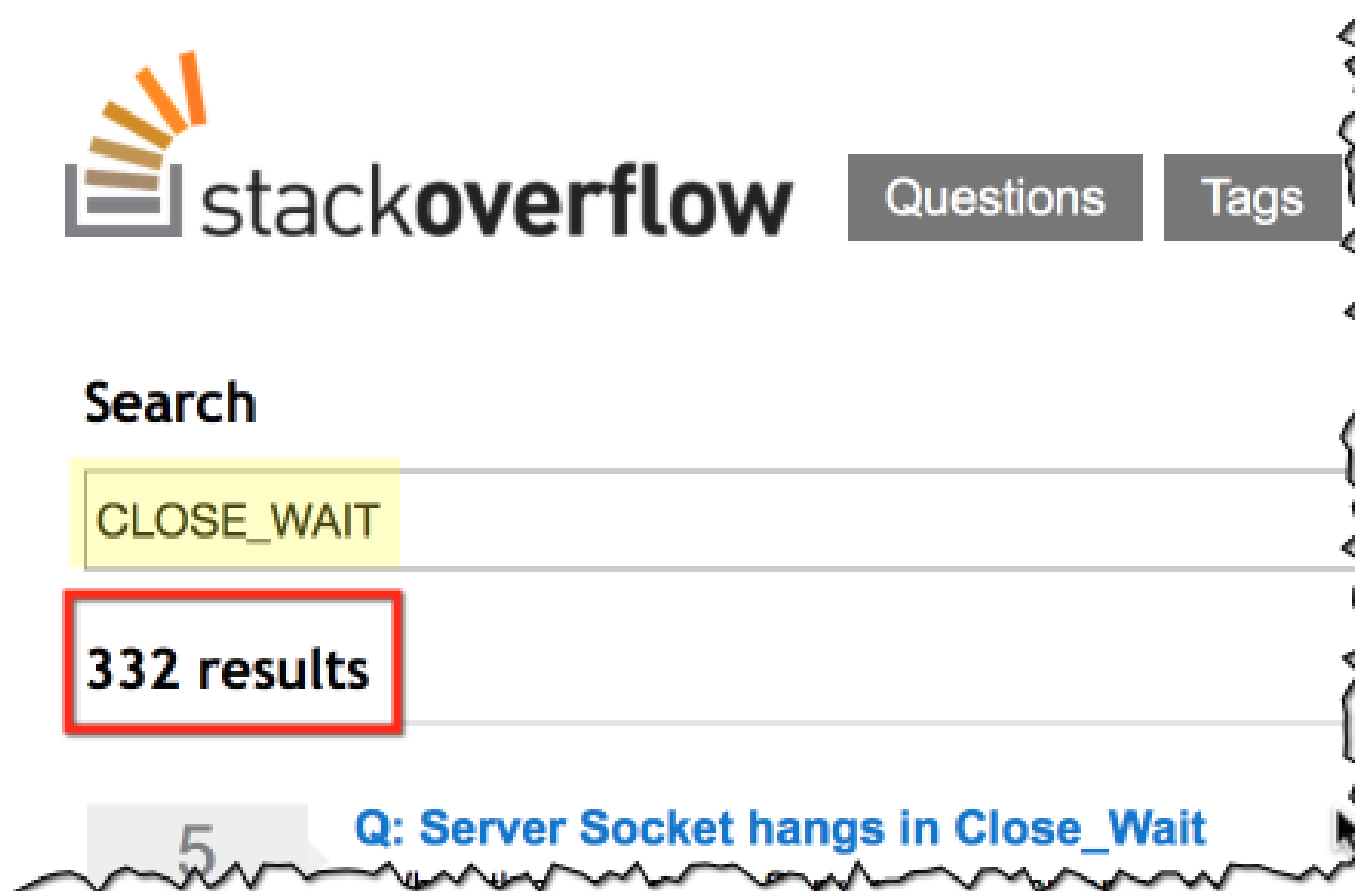
Please Visit Kates

Question!

How do I
close a
socket
correctly?



How hard can it be to close a socket?



The image shows a screenshot of the Stack Overflow website. At the top left is the Stack Overflow logo, followed by the text "stackoverflow". To the right of the logo are two buttons: "Questions" and "Tags". Below the logo is a search bar with the text "Search" above it. The search bar contains the text "CLOSE_WAIT". Below the search bar, the text "332 results" is displayed in a red-bordered box. At the bottom of the screenshot, there is a question title "Q: Server Socket hangs in Close_Wait" with a "5" in a grey box to its left. The entire screenshot is framed with a hand-drawn, jagged black border.

stackoverflow Questions Tags

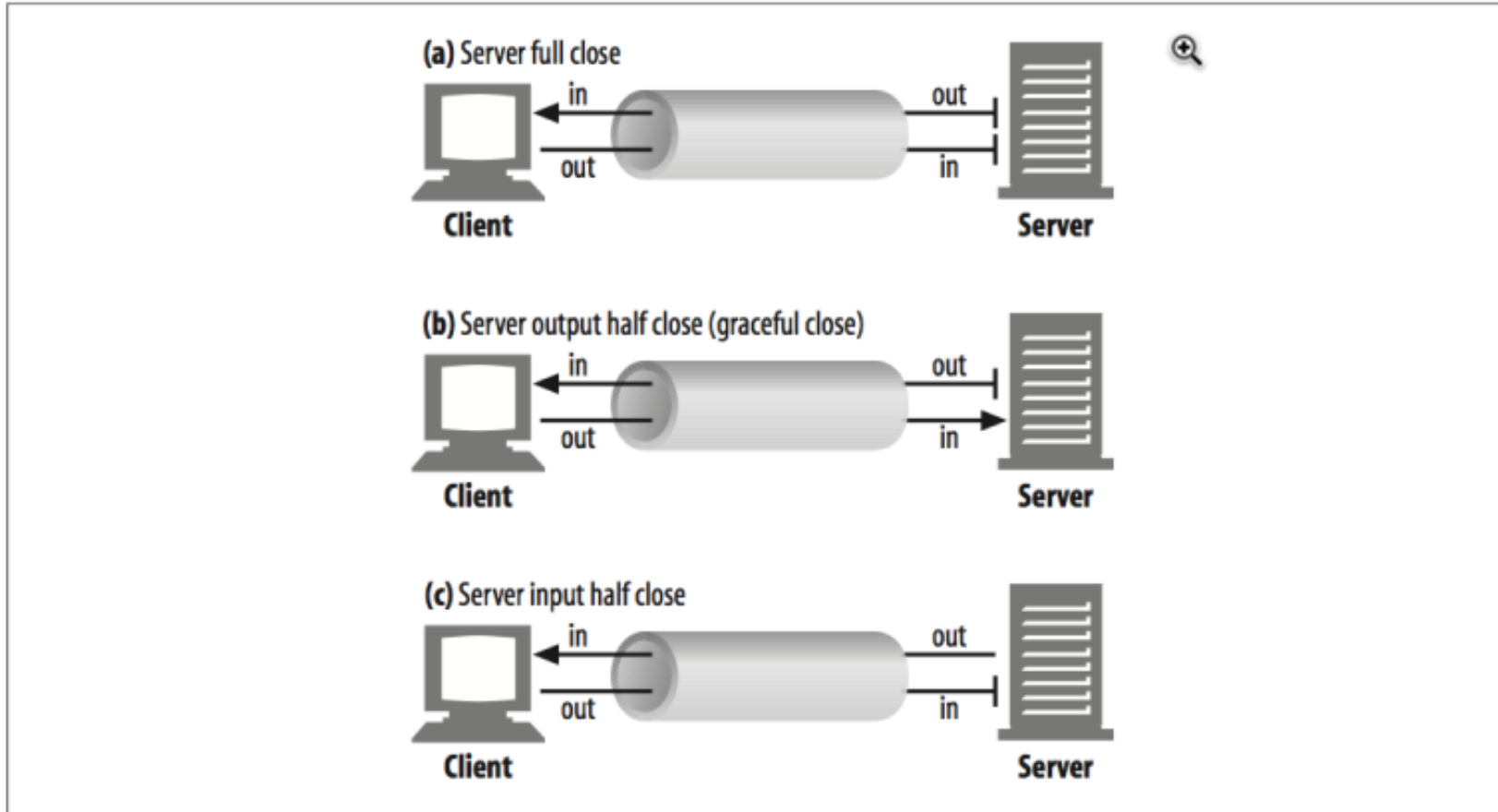
Search

CLOSE_WAIT

332 results

5 Q: Server Socket hangs in Close_Wait

The art of socket closing



Half-closed: no new customers



Never block in socket close()

- The other side expects you to clean up nicely
- It will give up on time out
- You will wait (forever)

**I ALWAYS CLOSE
CONNECTIONS**



**AND WHEN I DO IT, I USE
TRY-WITH-RESOURCES**

Remember?

```
try (AsyncHttpClient asyncHttpClient = new AsyncHttpClient()) {  
    ListenableFuture<Response> future = asyncHttpClient  
        .prepareGet("http://oss.jfrog.org/api/system/ping")  
        .execute(new AsyncCompletionHandler<Response>() {  
  
            @Override  
            public Response onCompleted(Response response) {  
                System.out.println(response.getResponseBody());  
                return response;  
            }  
  
            @Override  
            public void onThrowable(  
                t.printStackTrace();  
            }  
        });  
    future.get();  
}
```



Question!

How can I write
file parts
concurrently?



- Write to separate files, combine on finish
- Write to same file, seeking to the right position



WHY WON'T YOU JUST USE

JAVA.IO.RANDOMACCESSFILE?!?

The Java™ Tutorials

Basic I/O

I/O Streams

Byte Streams

Character Streams

Buffered Streams

Scanning and

Formatting

Scanning

Formatting

I/O from the Command

[File](#)

« [Previous](#) • [Trail](#) • [Next](#) »

Random Access Files

Random access files permit nonsequential, or random, access to a file's contents.

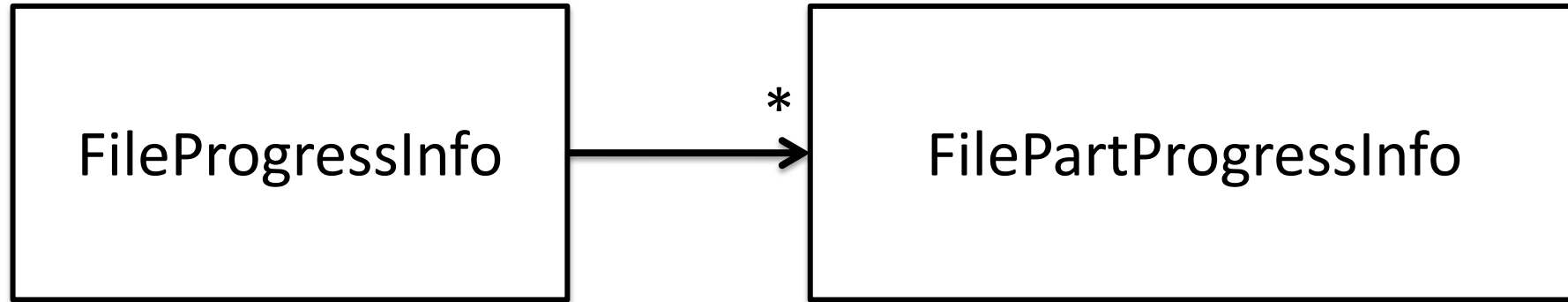
This functionality is possible with the `SeekableByteChannel` interface. They enable you to set or query the position, and you can then read the data from

Use FileChannel

- Implements SeekableByteChannel

```
java.nio.channels.FileChannel#write(  
    java.nio.ByteBuffer src, long position)
```

Download progress tracking



- **PersistentFileProgressInfo**
 - Save the total size, sha1, number of parts
 - State of each part (offset, size, completed...)

File Locking



File locking Levels

- VM level
- OS level

OS level File locking

- Multiple downloader instances writing to the same file
- Needed for writing:
 - Partial download file
 - Persistent download progress

OS Level File Locking - Exclusive

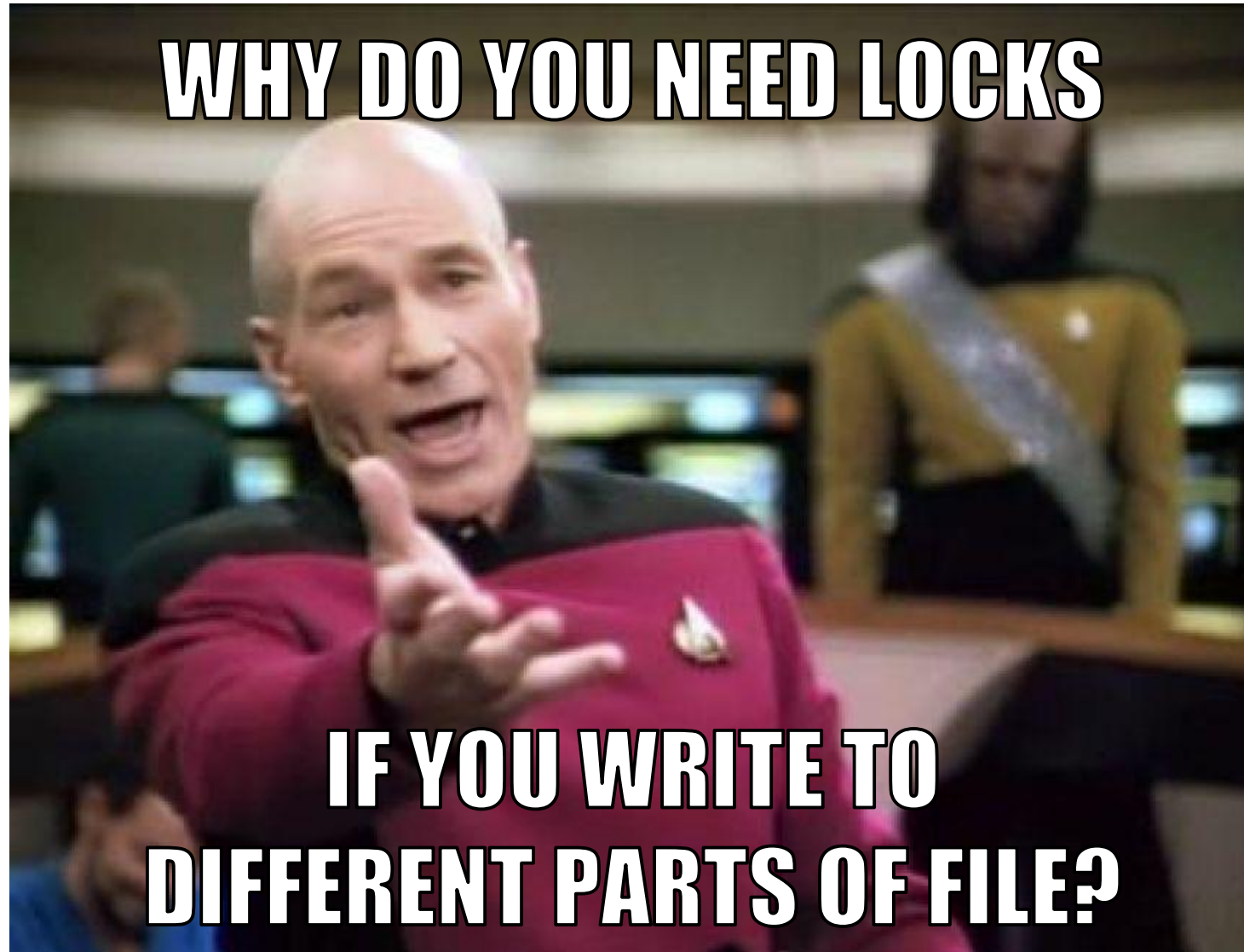
```
FileLock lock = fileChannel.tryLock();
                //Non-shared: (0L, Long.MAX_VALUE, false)
    if (lock == null) {
        throw new OverlappingFileLockException();
    }
    return lock;
}
```

OS Level File Locking – Advisory exclusive

```
private FileLock lock(FileChannel fileChannel) throws IOException {  
    FileLock lock = fileChannel.tryLock(Long.MAX_VALUE - 1, 1, false);  
    if (lock == null) {  
        throw new OverlappingFileLockException();  
    }  
    return lock;  
}
```



VM Level File Locking



VM Level File Locking

- Prevent same VM threads writing to the file when we started closing it
- Closing sequence:
 - Release file locks
 - Close channels
 - Rename a file to its final name (remove .part)
 - Erase progress info

VM Level File Locking

```
ReentrantReadWriteLock.ReadLock writeToFileLock = rwl.readLock();  
ReentrantReadWriteLock.WriteLock closeFileLock = rwl.writeLock();
```

```
public void close() throws IOException {  
    this.closeFileLock.lock();  
}
```

```
public int write(int partIndex, ByteBuffer buf) {  
    if (!this.writeToFileLock.tryLock()) {  
        throw new IllegalStateException("File is being closed");  
    }  
    ...  
}
```


What's next?

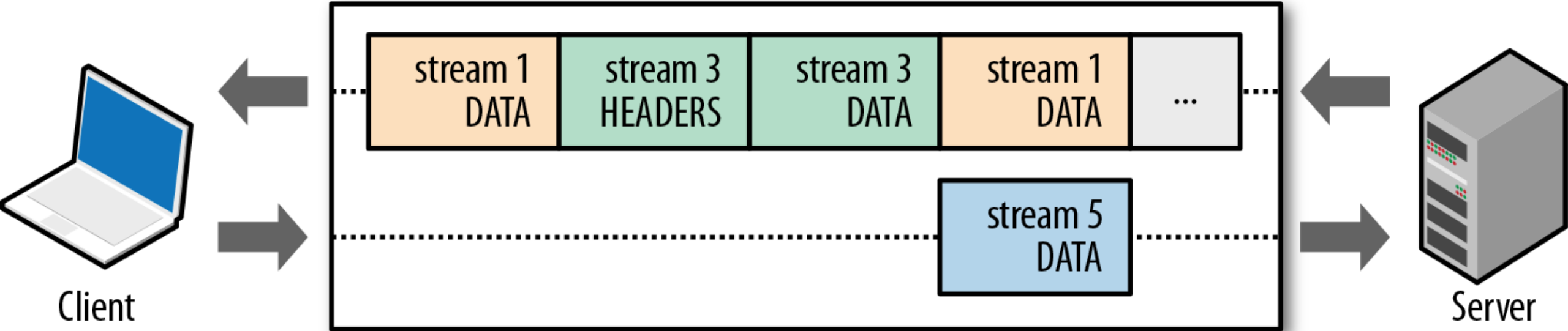


http/2

- Mostly standardizing Google's spdy
- Header compression
- multiplexing
- Prioritization
- Server push
- On the way clear some stuff
- E.g. compressed content length

Ease the load

HTTP 2.0 connection



Links!

- RTFM: [RFC 2616](#)
- Ultimate book: **HTTP: The Definitive Guide**
 - [Amazon](#)
 - [Safari](#)
- [Reactor pattern](#)
- [Doug Lea on NIO](#)

No, Thank you!

