

Не всё, но многое о том, что вы
хотели знать о том, как
написать асинхронные
многопоточные HTTP
приложения, но боялись
спросить у Шипилёва."



Jpoint

Краткое содержание

- В основном такое:



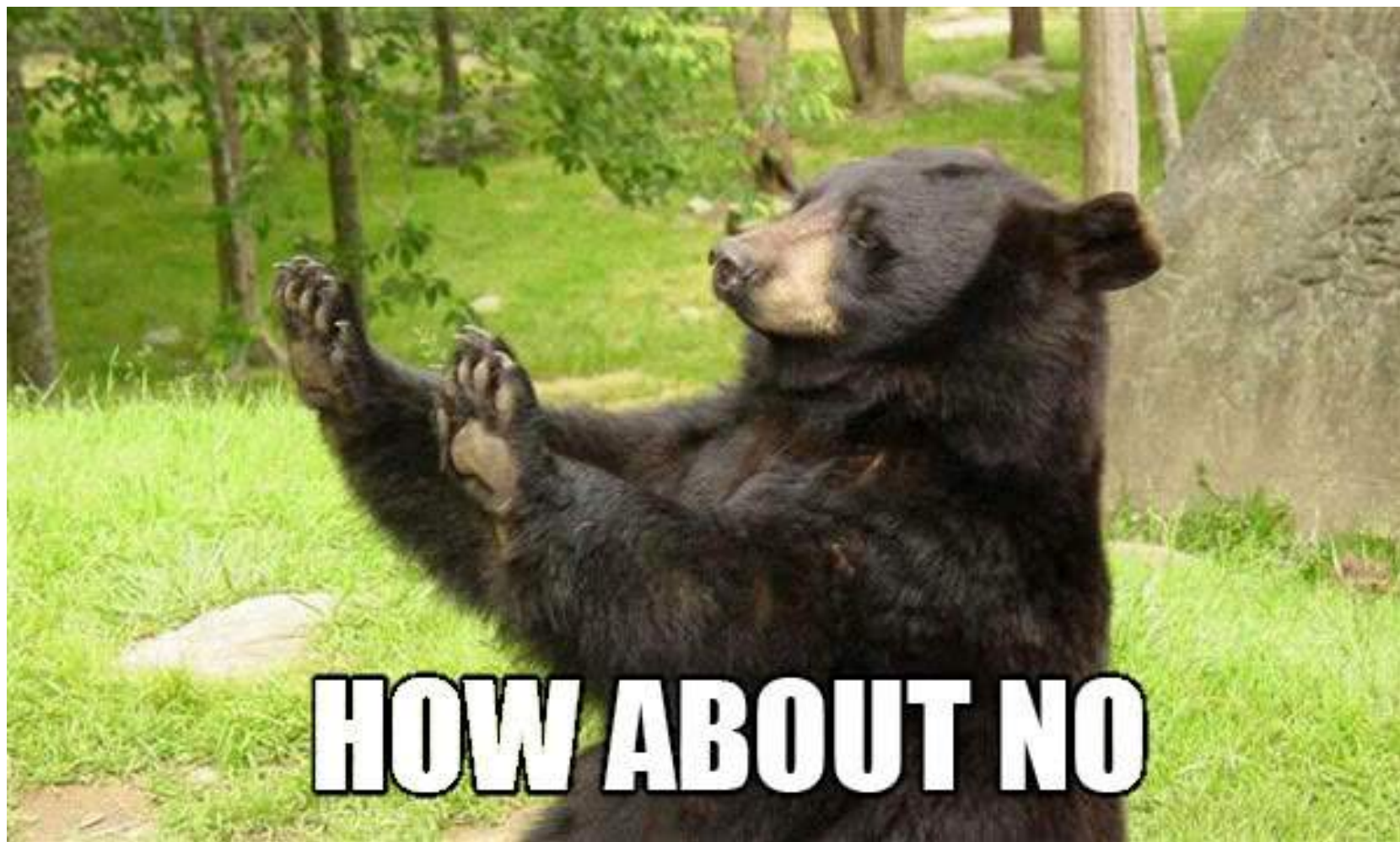
Краткое содержание

- И такое:



Краткое содержание

- И такое:



Барух это я.

@jbaruch

github.com/jbaruch

linkd.in/jbaruch



stackoverflow.com/users/402053/jbaruch



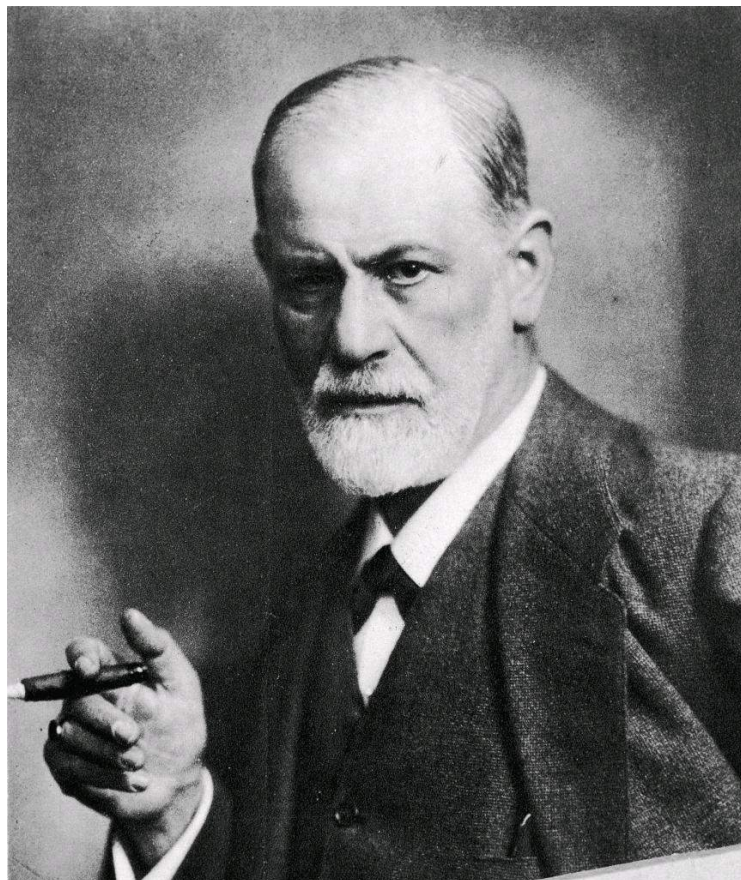
ПАРАЛЛЕЛЬНЫЕ ЗАГРУЗКИ



ПОЧЕМУ ВЫ ИХ НЕ УМЕЕТЕ?!

Требования

- Параллельные загрузки файлов
- Параллельная загрузка файла
- Прерывания и паузы
- Мониторинг процесса
- Кэширование по контрольной сумме



Игра в ассоциации: «Менеджер Загрузок»

**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)



О, нашли, написан на Java!

sourceforge[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

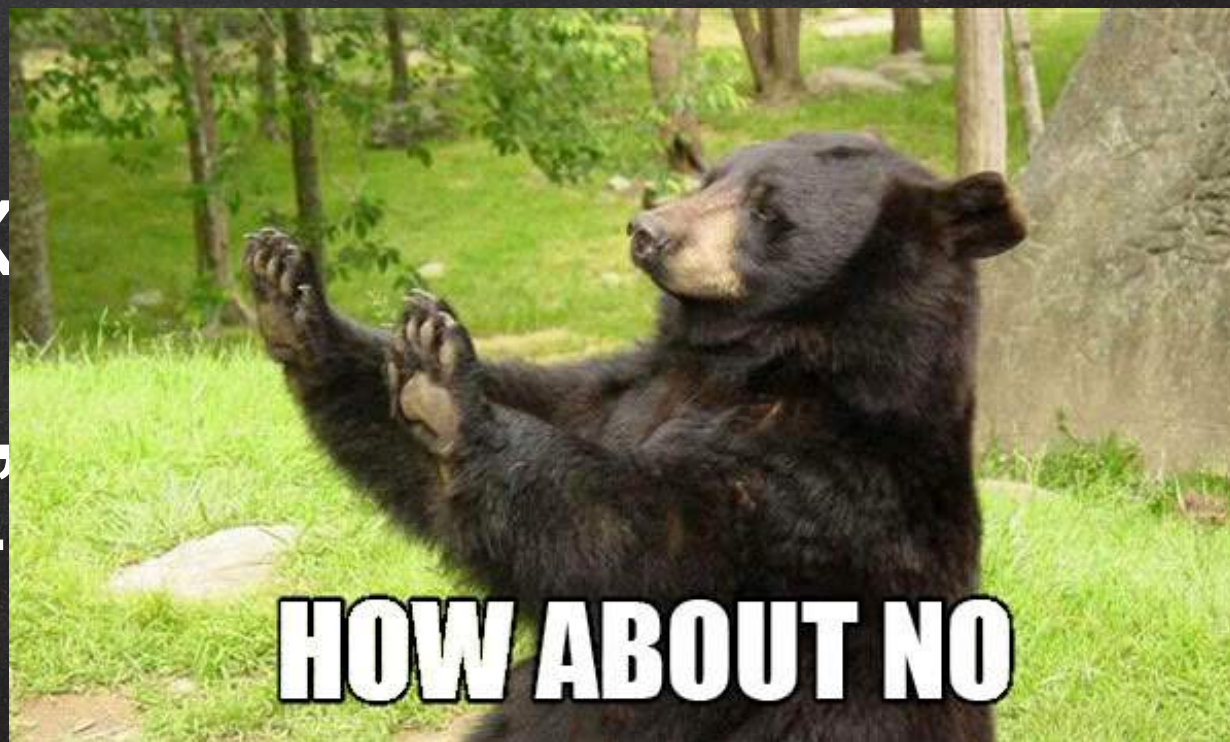
- Supports multiple download at a time
- bored of deleting old downloads one by one...delete all completed download at once
- looking for an old download.....just search it by name.
- 14 different look & feel
- Manage your downloads according to categories.
- System Tray Support
- Fast downloading speed.....
- 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.)

НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ

И ВЗЯТЬ JDM

Ну, берем, не?!

1. Ни следа лицензии
2. Ни сайта, ни
3. Ни следа исх
4. Да и вообще,
а не библиот

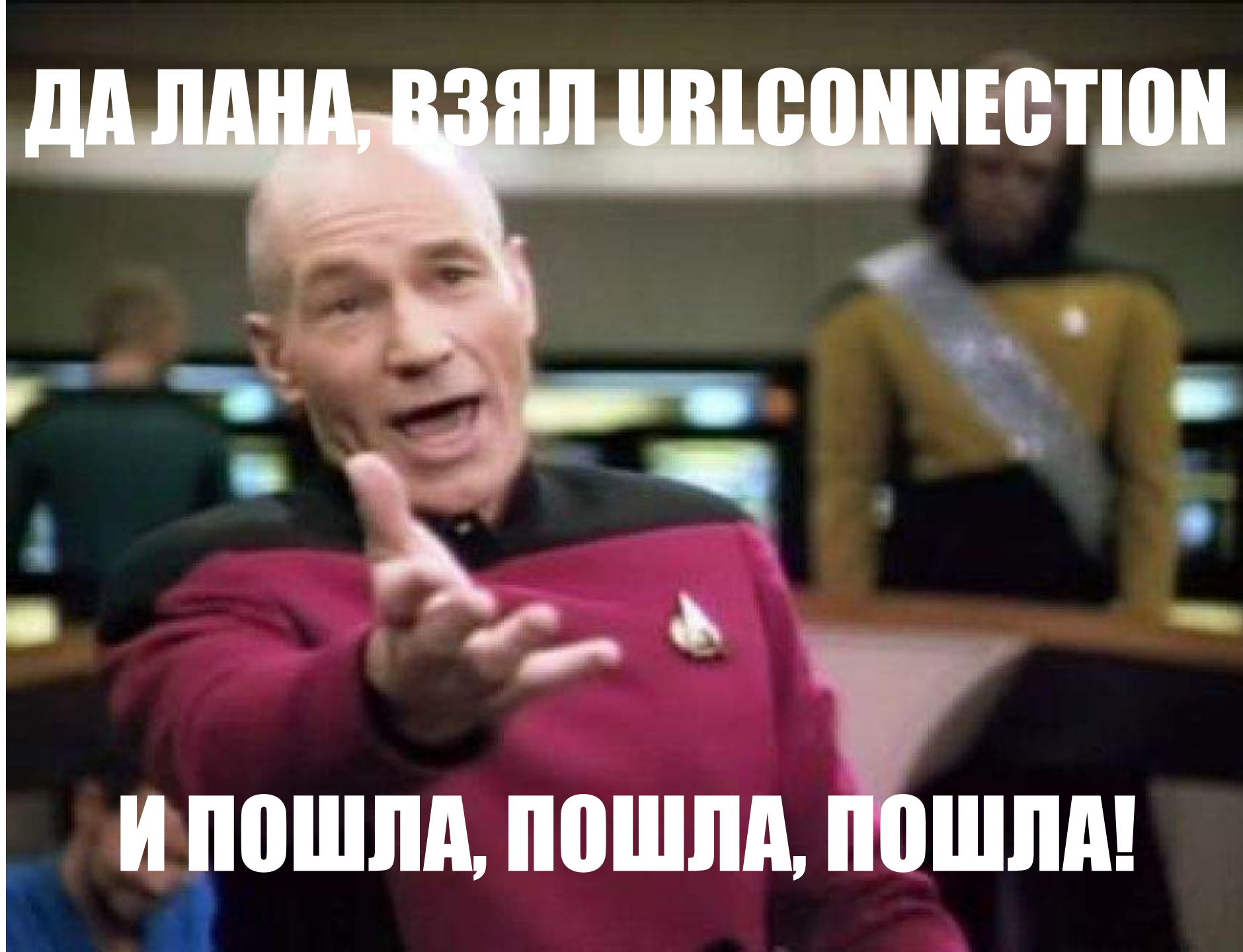


ХОЧЕШЬ ХОРОШИЙ МЕНЕДЖЕР ЗАГРУЗОК?



ПИШИ ЕГО САМ!

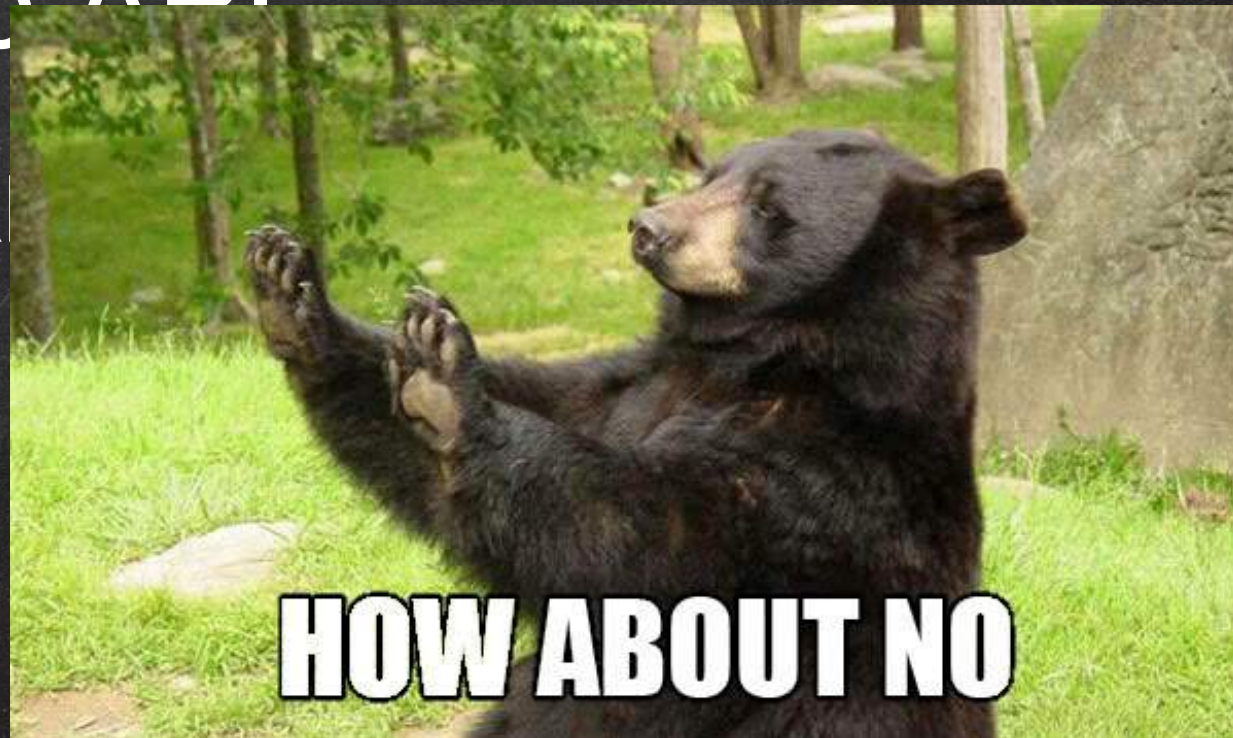
ДА ЛАНА, ВЗЯЛ URLCONNECTION



И ПОШЛА, ПОШЛА, ПОШЛА!

Java.net.urlconnection

1. Все проходит через память
2. нет нормального API
3. блокирует поток



Не блокируй!



Что мы ищем:

1. Асинхронный/не блокирующий
2. С коллбэками на эвенты

Что для этого понадобится:

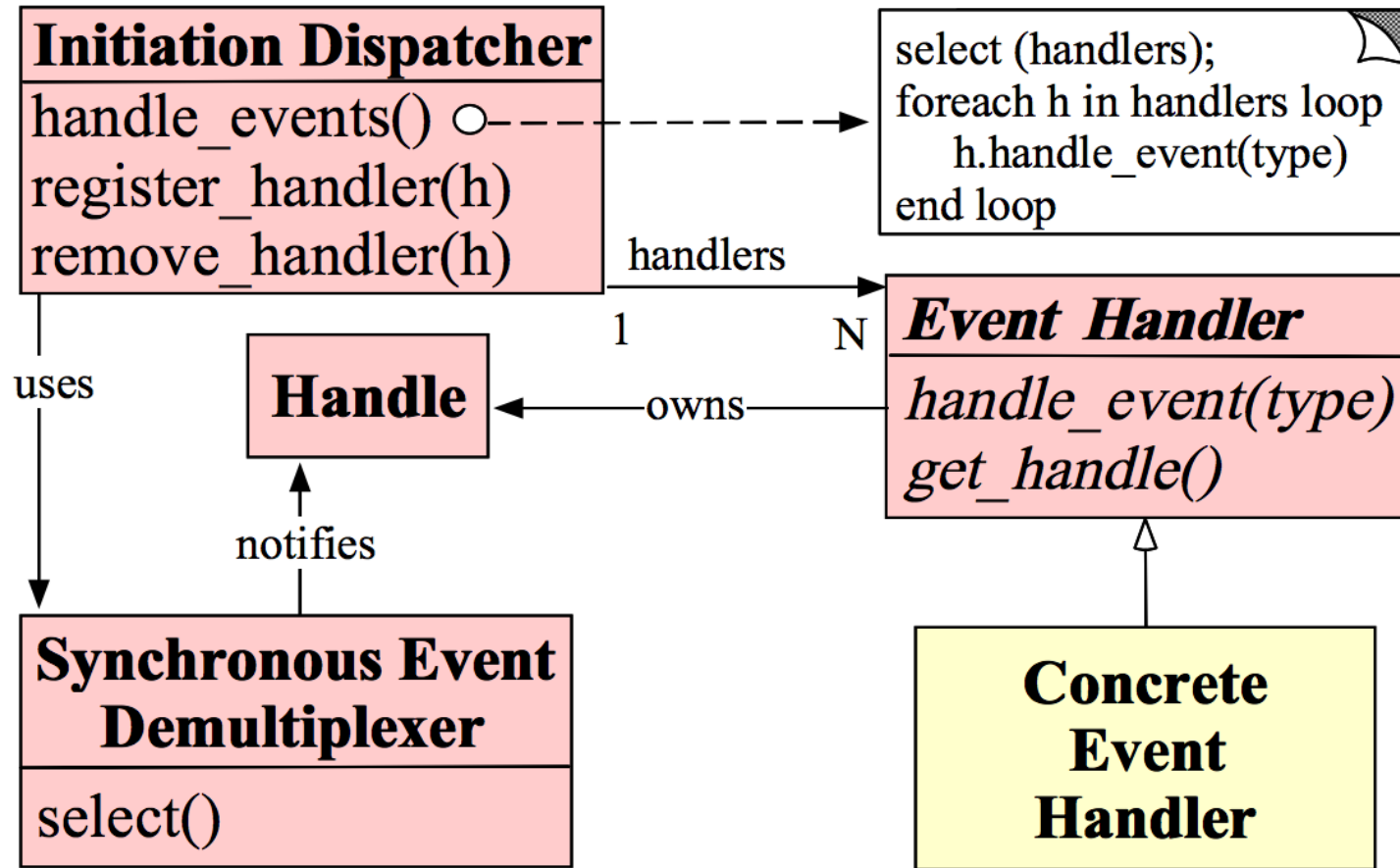
1. реактор
2. nio

Добро пожаловать в Реактор

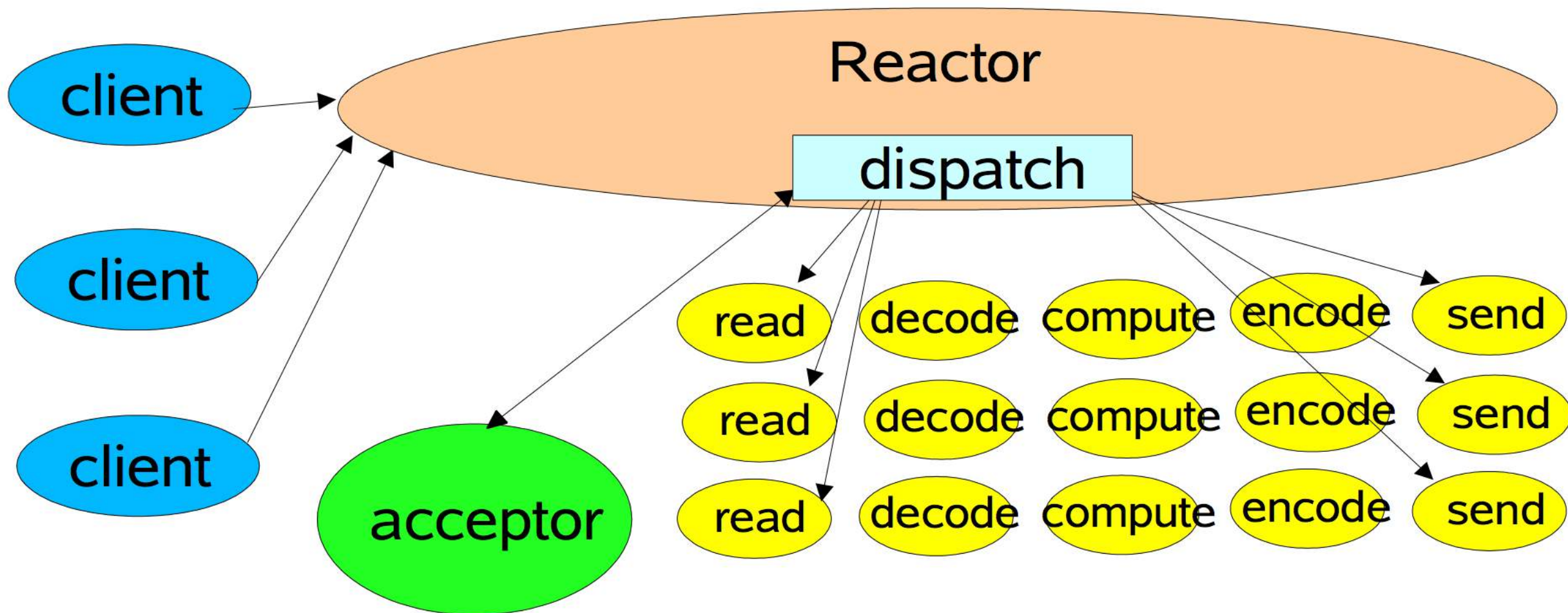


- Паттерн для легковесной многопоточности
- событийный
- Повторное использование потоков
- Не блокирующий IO

BOT как-ТО так:



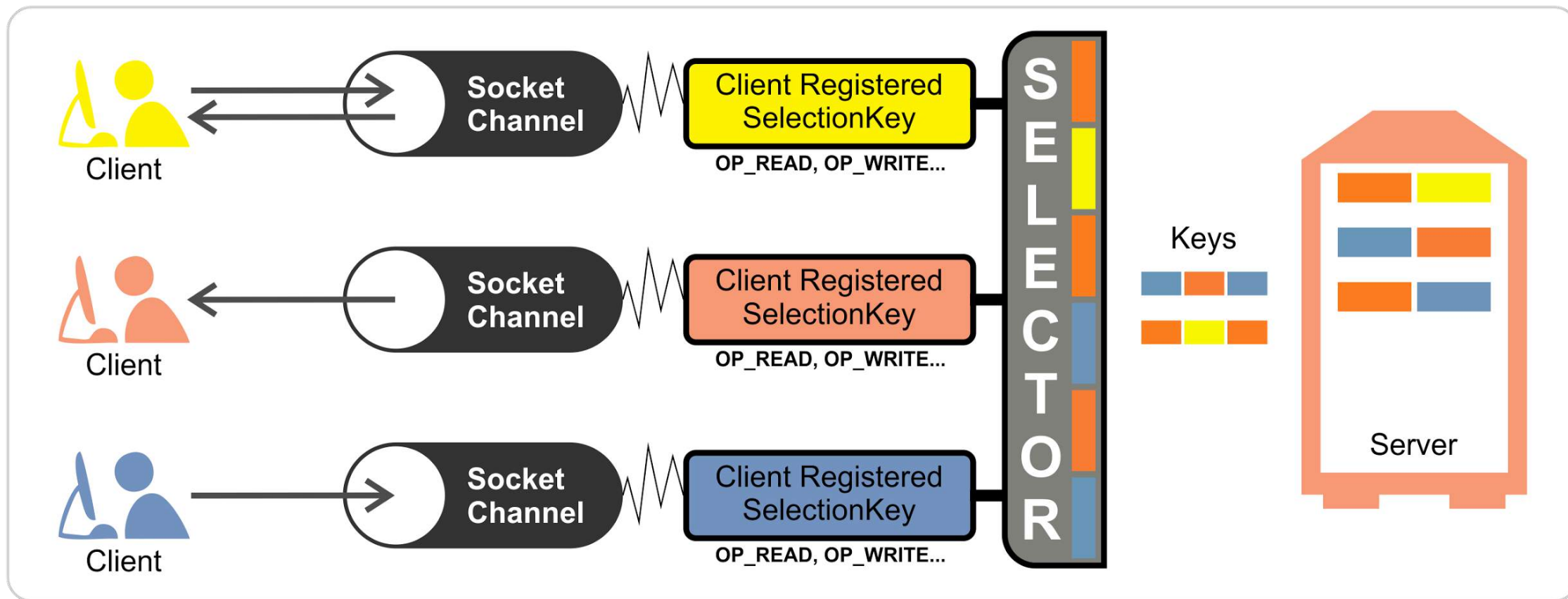
Угадайте, кто рисовал



В Джаве говорим
«реактор»,
подразумеваем
NIO.



Селектор проводит селекцию



Код! На Java! Регистрация

```
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);
```

Код! На Java! Распределение

```
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();  
}
```


Обработка событий в реакторе – сложная штука

- Надо помнить состояние
- Буфферизация – надо собирать куски файлов
- надо координировать события

ЭТО НИО – ТЕЛОХРАНИТЕЛЬ БУДДЫ



ОН СИЛЁН

Библиотеки Nio

- В основном серверные
 - Netty, grizzly, etc.
- Apache Mina
- Apache HTTP components asyncclient
- Ning http client

Библиотеки Nio

- ~~В основном серверные~~
 - ~~Netty, grizzly, etc.~~
- Apache Mina
- Apache HTTP components asyncclient
- Ning http client



- Серверная и клиентская `nio` библиотека
- Отпочковалась от `netty`
- Последний релиз: октябрь 2012

Contributors

Commits

Code frequency

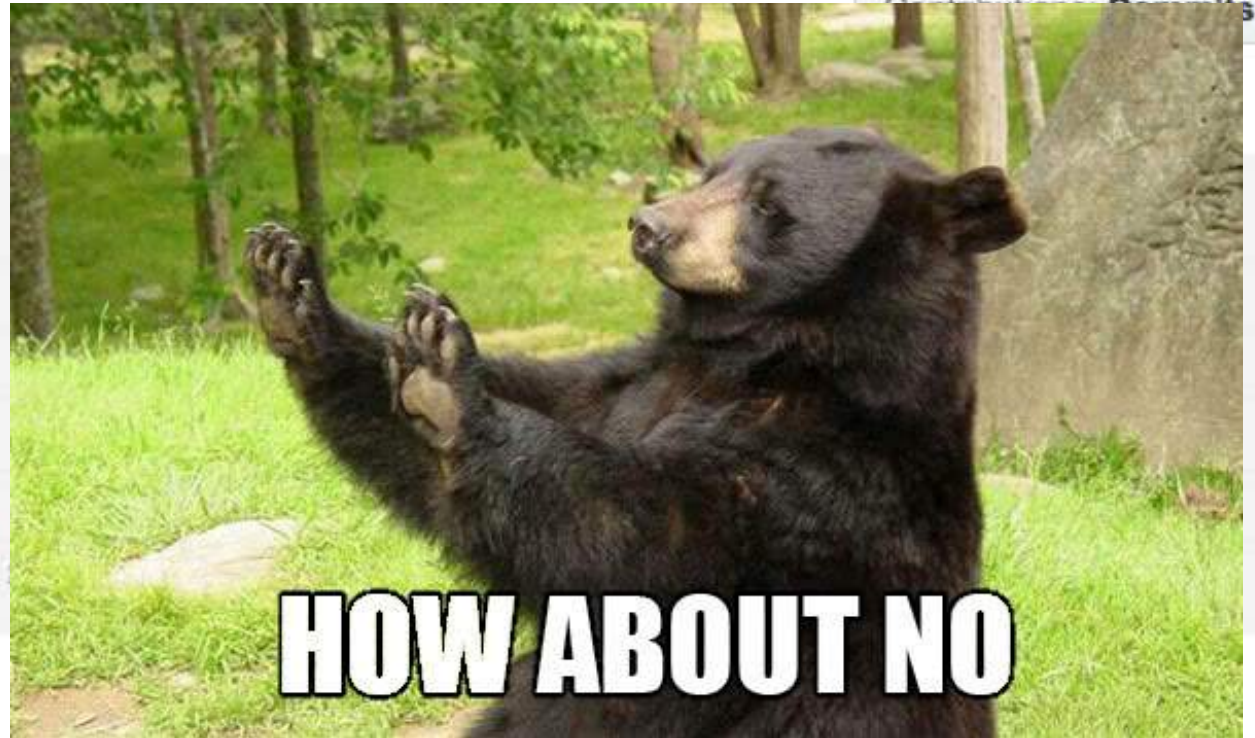
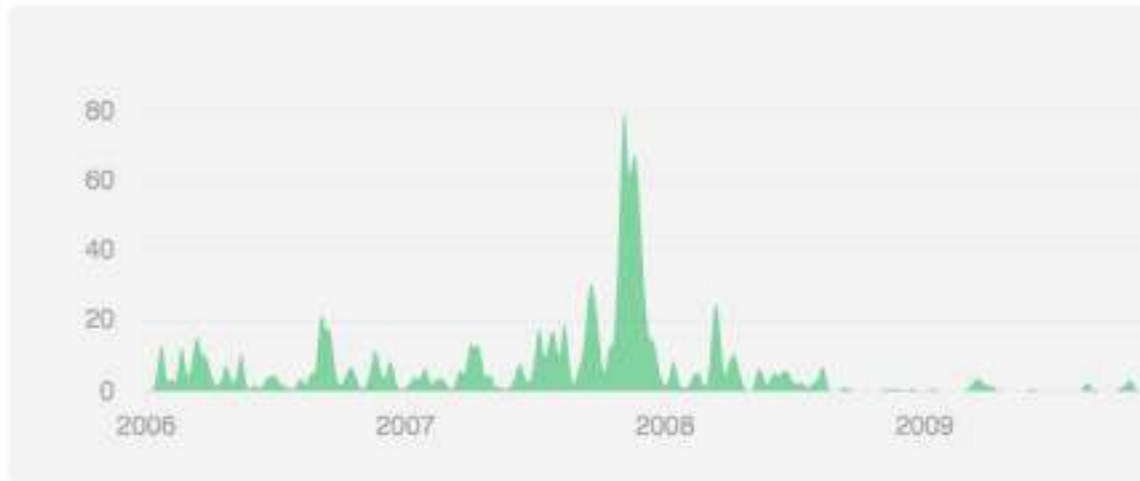
Punch card

Network

Members

Dec 25, 2005 – Sep 21, 2014

Contributions to trunk, excluding merge commits



Библиотеки Nio

- ~~В основном серверные~~
 - ~~Netty, grizzly, etc~~
- ~~Apache Mina~~
- Apache HTTP components asyncclient
- Ning http client

WHAT IS
NING?

NING
MODE MEDIA

CULTIVATING
COMMUNITY

[Pricing](#)

[Login](#) ▶

NING

Your brand. Your members.
Your control.



01:01

||||| **HD** ✕

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.



А вот она!



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

► [HttpClient 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.

Концепции HAC

- Request producer
- Response consumer


```
try (CloseableHttpClient asyncHttpClient = HttpClientClients.createDefault()) {
    asyncHttpClient.start();
    Future<HttpResponse> future = asyncHttpClient.execute(
        HttpClientMethods.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");
            }

        }, null);
    HttpResponse response = future.get();
}
```



ОПАЧКИ...

ОНИ ЖЕ ОДИНАКОВЫЕ!

Это мы, выбираем
между ning и http asyncclient



«Каждая компьютерная проблема
решается еще одним уровнем
абстракции»

Дэвид Уиллер

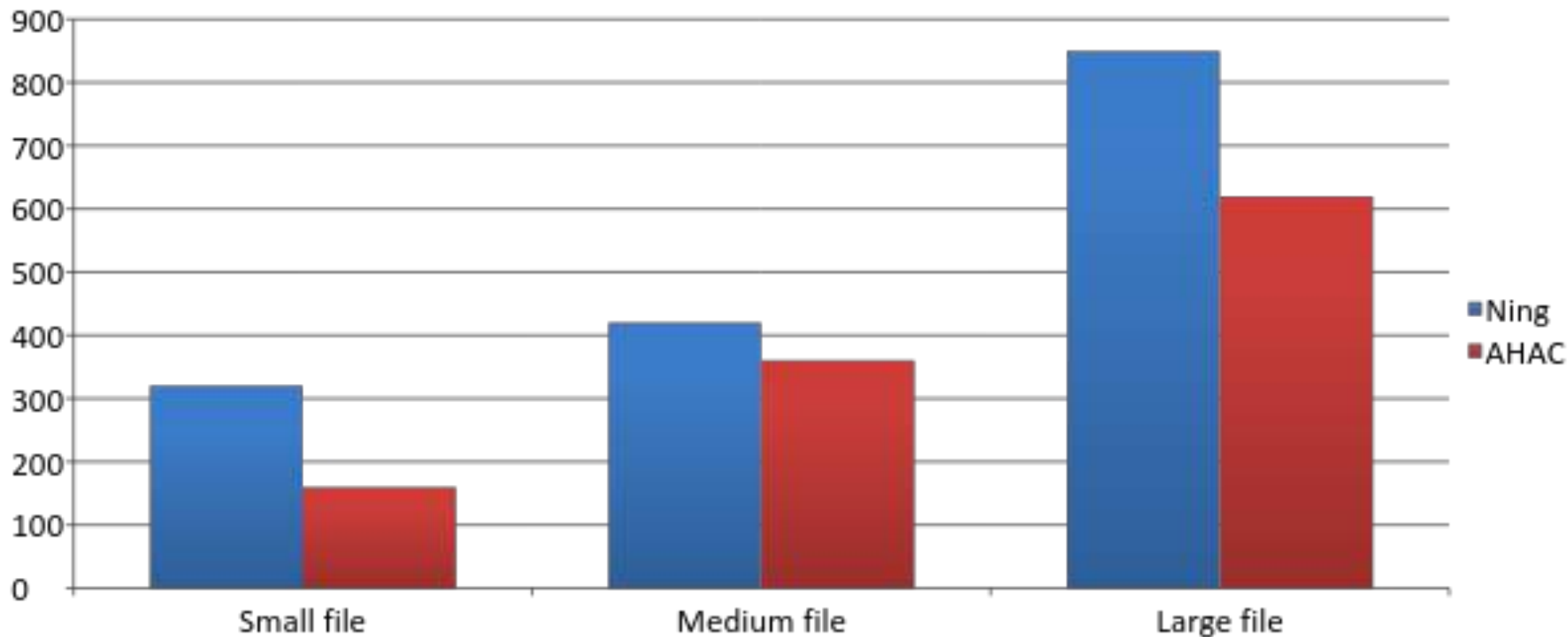


```
public interface HttpProviderDownloadHandler {  
    void onResponseReceived(int statusCode, Map<String, List<String>> headers);  
    boolean onBytesReceived(ByteBuffer buf);  
    void onFailed(Throwable error);  
    void onCancel();  
    void onCompleted();  
}
```


Но всё таки надо выбирать

Фича/Библиотека	Ning client	Http Async Client
-----------------	-------------	-------------------

Как с перфромансом?





Rfc2616: Целая вселенная

Fielding, et al
□
RFC 2616

[Page 6]

June 1999

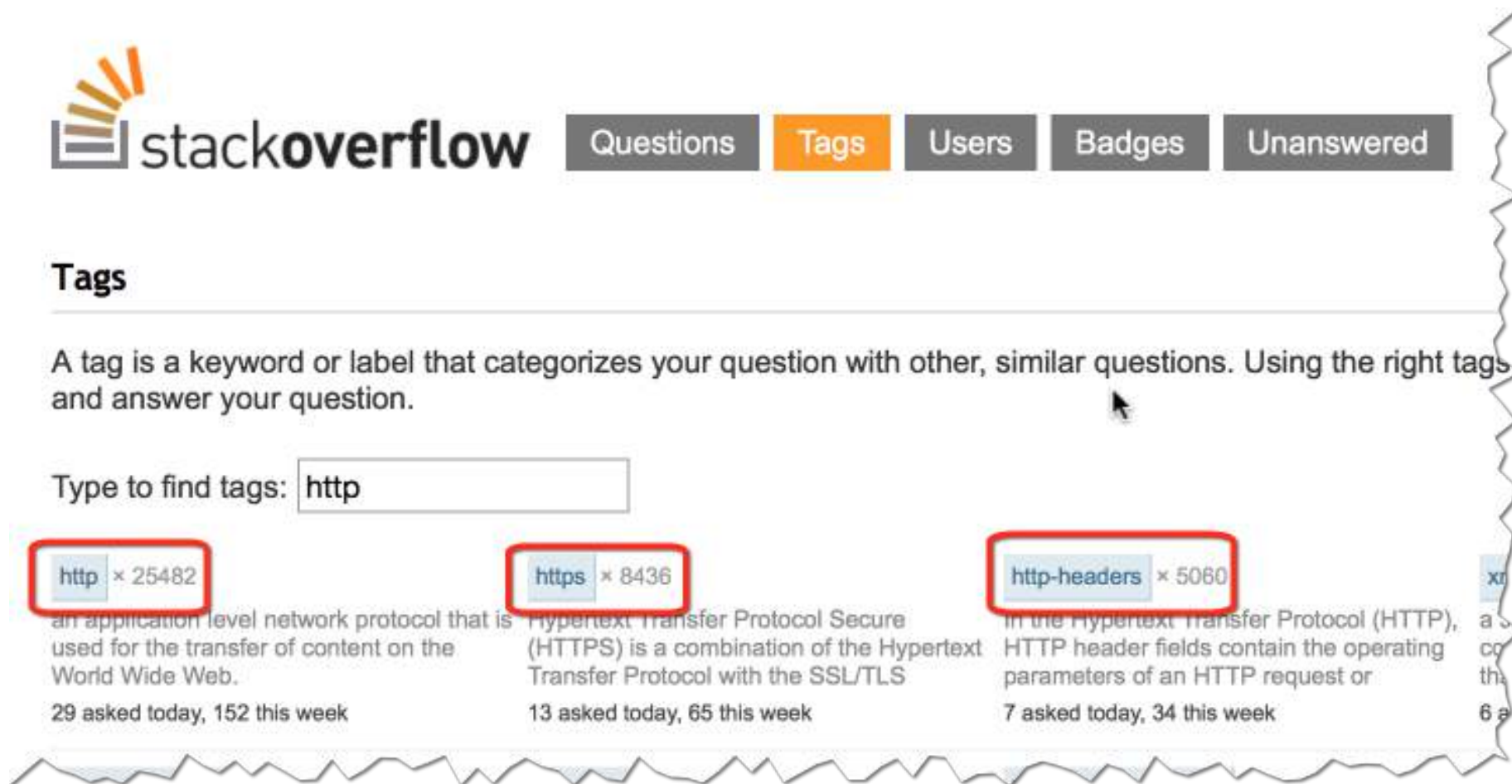
19.4.1	MI		167
19.4.2	Co		167
19.4.3	Co		168
19.4.4	In		168
19.4.5	No		168
19.4.6	In		169
19.4.7	MH		169
19.5	Addit		169
19.5.1	Co		170
19.6	Compa		170
19.6.1	Ch		171
19.6.2	Co	ons	172
19.6.3	Ch		172
20	Index		175
21	Full C		176



Совсем запутались?



Вперед, на StackOverflow. Заодно и репутацию прокачаете.



The screenshot shows the 'Tags' section of the Stack Overflow website. At the top is the Stack Overflow logo and a navigation bar with links for Questions, Tags (highlighted in orange), Users, Badges, and Unanswered. Below the navigation bar is the 'Tags' heading. A descriptive paragraph explains that a tag is a keyword or label used to categorize questions. Below this is a search input field with the text 'Type to find tags:' and a text box containing 'http'. Three tag suggestions are displayed below the input field, each enclosed in a red rectangular box. Each suggestion includes the tag name, a count of questions (indicated by 'x'), a brief description, and the number of questions asked today and this week. The tags shown are 'http' (25482 questions), 'https' (8436 questions), and 'http-headers' (5060 questions). The entire screenshot is framed by a decorative torn paper border on the right and bottom edges.

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:

Tag	Count	Description	Asked Today	Asked This Week
http	25482	an application level network protocol that is used for the transfer of content on the World Wide Web.	29	152
https	8436	Hypertext Transfer Protocol Secure (HTTPS) is a combination of the Hypertext Transfer Protocol with the SSL/TLS	13	65
http-headers	5060	in the Hypertext Transfer Protocol (HTTP), HTTP header fields contain the operating parameters of an HTTP request or	7	34



А эту, когда
понял, что range
header теряется
при редиректе

ИЗ ОКОПОВ HTTP



У меня вопрос!

Каким должен
быть content-
length если
используется
сжатие?



content-length



9



2

The client is
Encoding: s

What would
data.

Thanks,

[http](#)[share](#) | [edit](#)[add a comment](#)[start a bounty](#)

with Accept-

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



Closed

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.



grmoe commented on 18 Mar 2013

Collaborator

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

...

У меня вопрос!



Почему при редиректе на CDN
файл начинает качаться с нуля?


```
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;
}});
```

У меня вопрос!

Сколько одновременных
закачек можно ставить?



HTTP 1.1 ТАКОЙ:

**МОЖНО, ПОЖАЛУЙСТА,
НЕ БОЛЬШЕ 2 ПОДКЛЮЧЕНИЙ?**



БРАУЗЕРЫ ТАКИЕ:

[Summary](#)[Security](#)[Rich Text](#)[Selectors API](#)[Network](#)[Acid3](#)[JSKB](#)

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

URL ENCODING?

EASY.

У меня вопрос!



Что не так с
ЭТИМ КОДОМ?

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

Закодированное URLEncoder-ом не
раскодируешь обратно...

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

Нельзя раскодировать обратно после того,
как закодируешь:

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

Не пользуйтесь `java.net.URLEncoder`

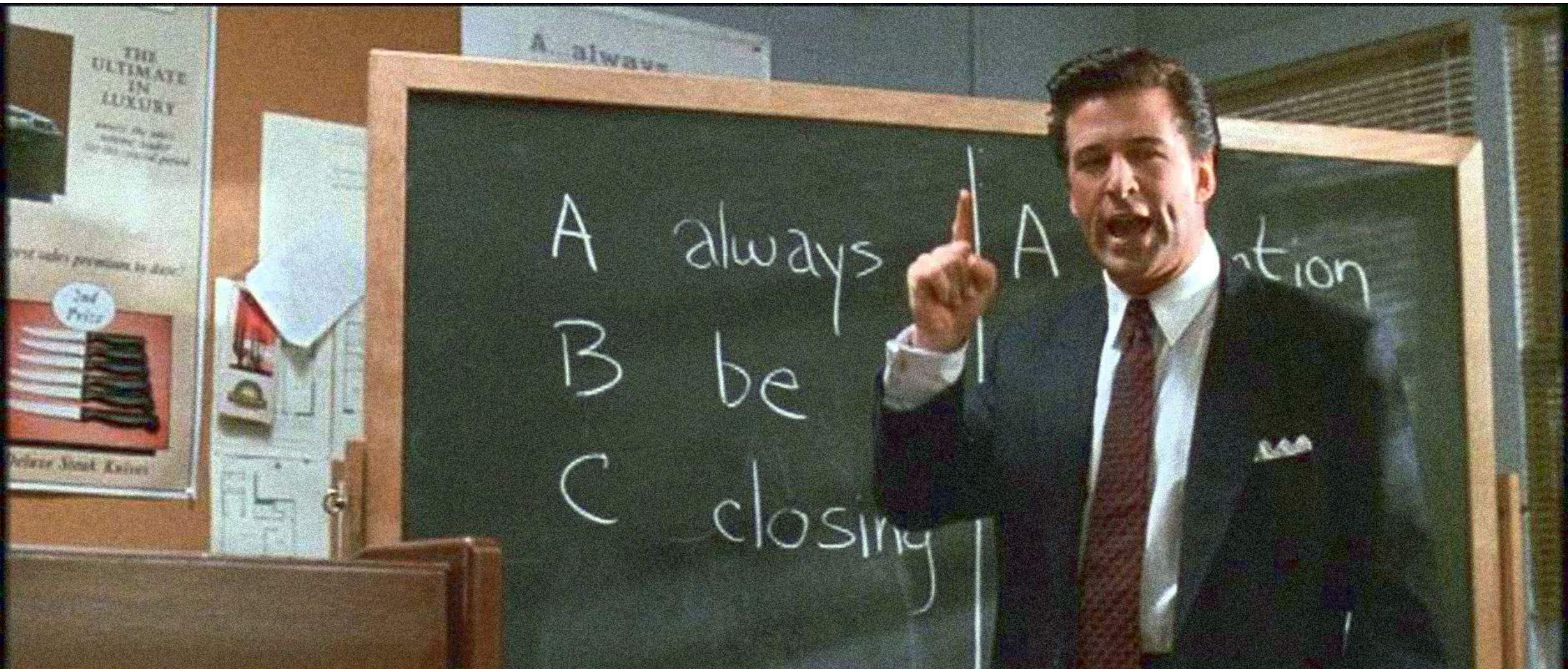
“Utility class for HTML form encoding. This class contains static methods for converting a String to the **`application/x-www-form-urlencoded`** MIME format.

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

У АНС есть способы лучше

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

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



A. always

A always

B be

C closing

A

ntion

THE
ULTIMATE
IN
LUXURY

...the only
...the only

2nd
Floor

...under previous to date



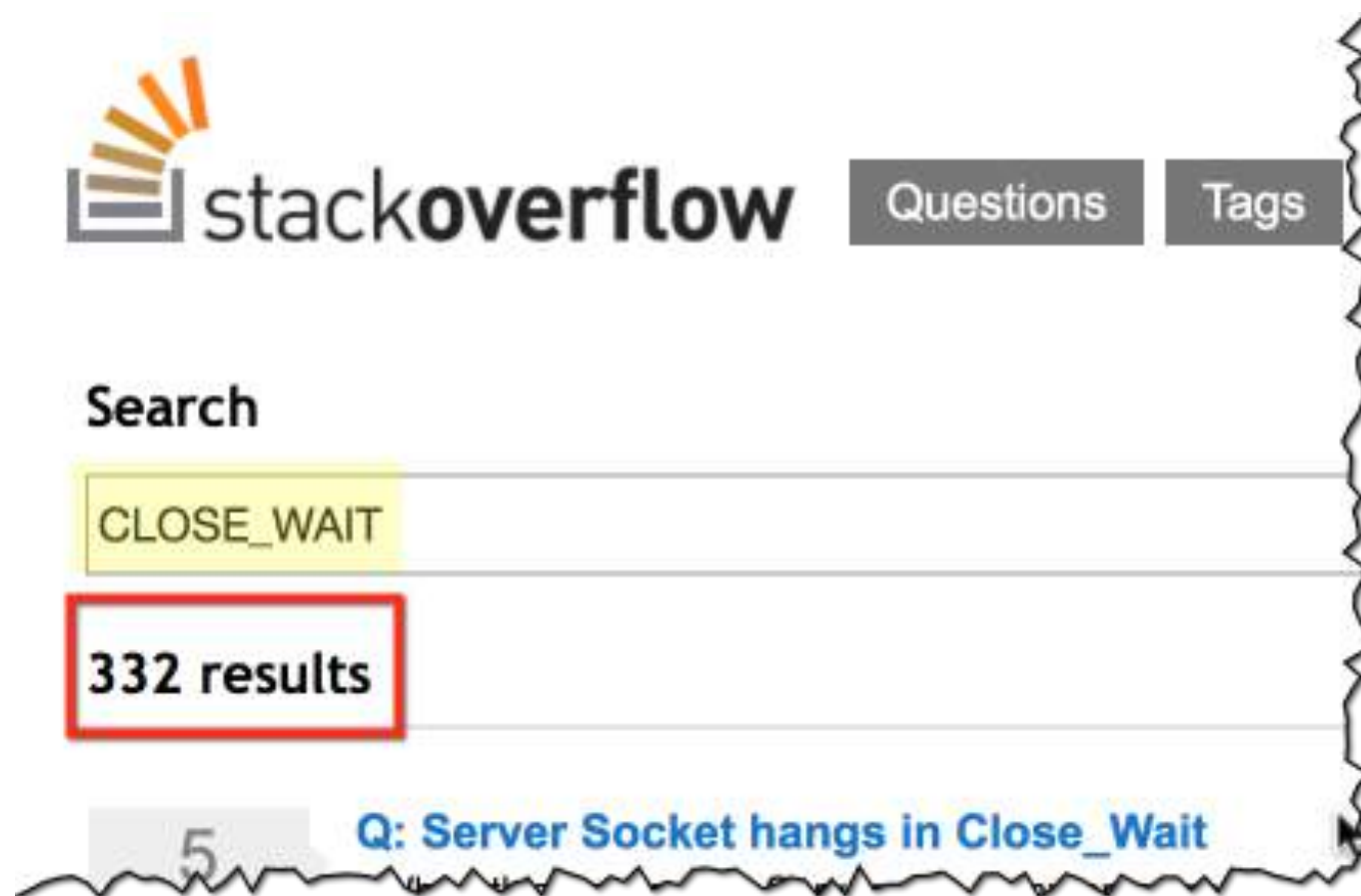
Deluxe Street Kitchen

У меня вопрос!

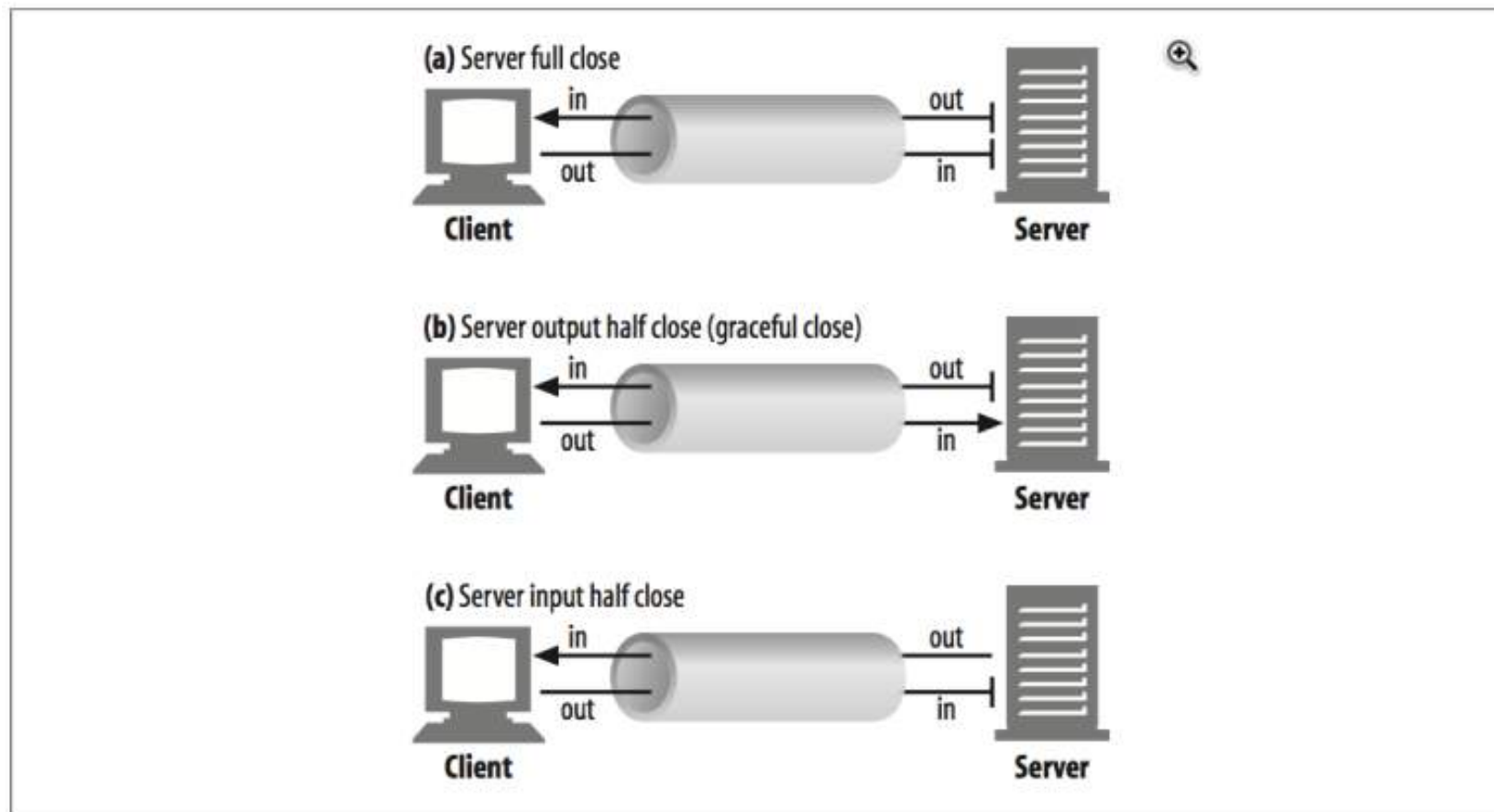
Как
правильно
закрывать
сокет?



Да ладно, чего там сложного?



Трактат о закрывании сокета



Полузакрото:
«магазин закрывается, проходите к кассам»



Не блокируйте в close()!

- Сервер ожидает, что вы закроете как надо
- В какой-то момент ему надоест
- А вы так и будете ждать

**Я ВСЕГДА ЗАКРЫВАЮ
ПОДКЛЮЧЕНИЯ**

**И КОГДА Я ЭТО ДЕЛАЮ,
Я ПОЛЬЗУЮСЬ
TRY-WITH-RESOURCES**

Вот, и Чак одобряэ.

```
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();  
}
```



У меня вопрос!

Как писать
куски файлов
параллельно?



– Писать в разные файлы, собирать в один в конце

– Писать в тот-же файл, начиная с правильного места

НУ ТАК ЕСТЬ ЖЕ ДЛЯ ТАКОГО



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

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

Random Access Files

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

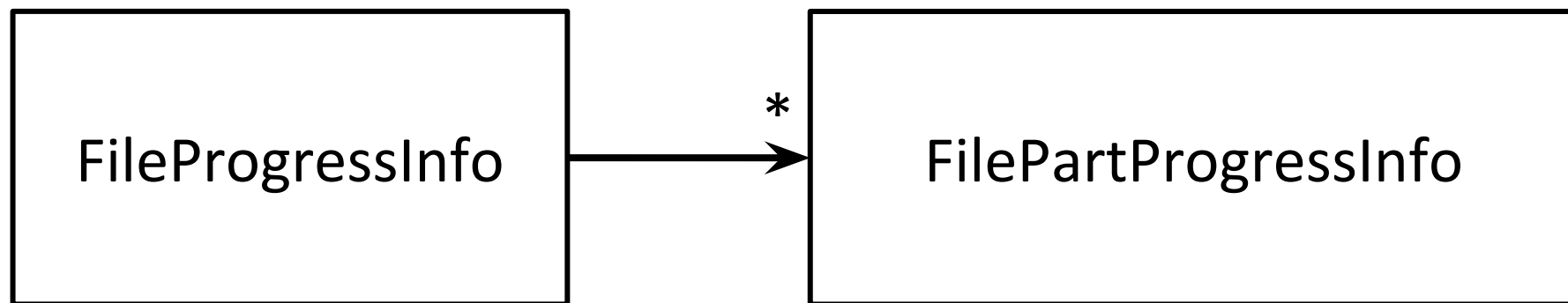
This functionality is possible with the `SeekableByteChannel` interface. This interface enables you to set or query the position, and you can then read the data from

Пользуемся FileChannel

- Реализует SeekableByteChannel

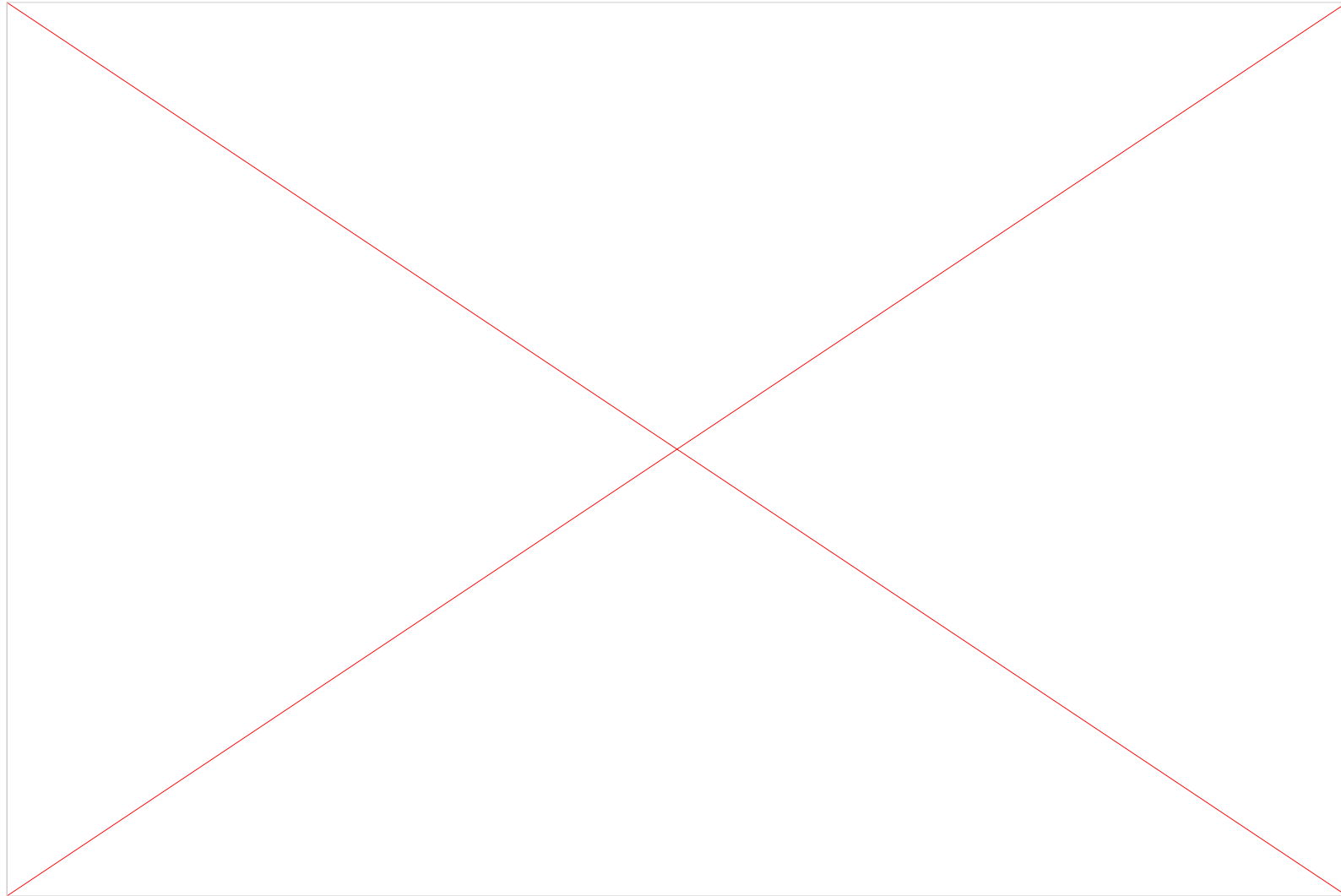
```
java.nio.channels.FileChannel#write(  
    java.nio.ByteBuffer src, long position)
```


Отслеживаем прогресс



- **PersistentFileProgressInfo**
 - Сохраняем размер, чексамы, количество частей
 - Состояние каждой части(отступ, размер, состояние...)

Локинг файлов



Уровни локинга

- Уровень JVM
- Уровень операционной системы

Уровень ОС

- Защита от дурака: Несколько приложений скачивают тот же файл в то же место
- Надо лочить:
 - Частично закаченный файл
 - Сохраненный на диск прогресс

Легко, нам нужен shared lock!

- Один пишет, остальные ~~смотрят~~ читают

tryLock

```
public abstract FileLock tryLock(long position,  
                                long size,  
                                boolean shared)  
                                throws IOException
```




java.io.IOException "The process cannot access the file because another process has locked a portion of the file"

Windowz...



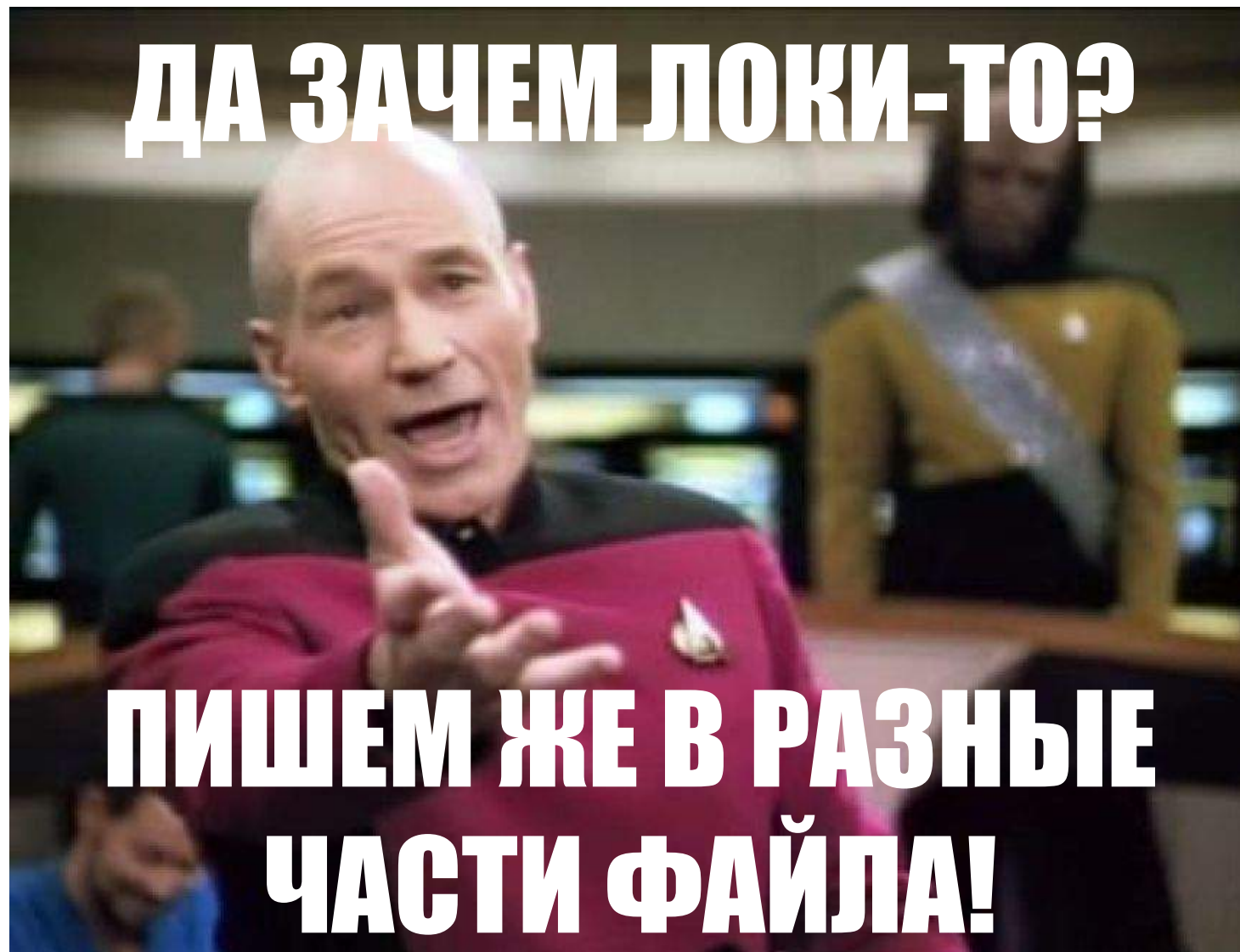
На каждую хитрую Винду свой StackOverflow с винтом

```
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;  
}
```



Щито?!

Локинг на уровне VM



Локинг на уровне VM

- Закрывать файл должен только один поток
- Процесс закрытия:
 - Снять локи ОС
 - Закреть channels
 - Переименовать файл в конечный вид (убрать . part)
 - Стереть файл прогресса

Локинг на уровне VM

```
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");  
    }  
    ...  
}
```

Будущее...



Будущее уже (почти) здесь!

Hypertext Transfer Protocol version 2 draft-ietf-httpbis-http2-latest

Abstract

This specification describes an optimized expression of the semantics of the Hypertext Transfer Protocol (HTTP). HTTP/2 enables a more efficient use of network resources and a reduced perception of latency by introducing header field compression and allowing multiple concurrent exchanges on the same connection. It also introduces unsolicited push of representations from servers to clients.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts

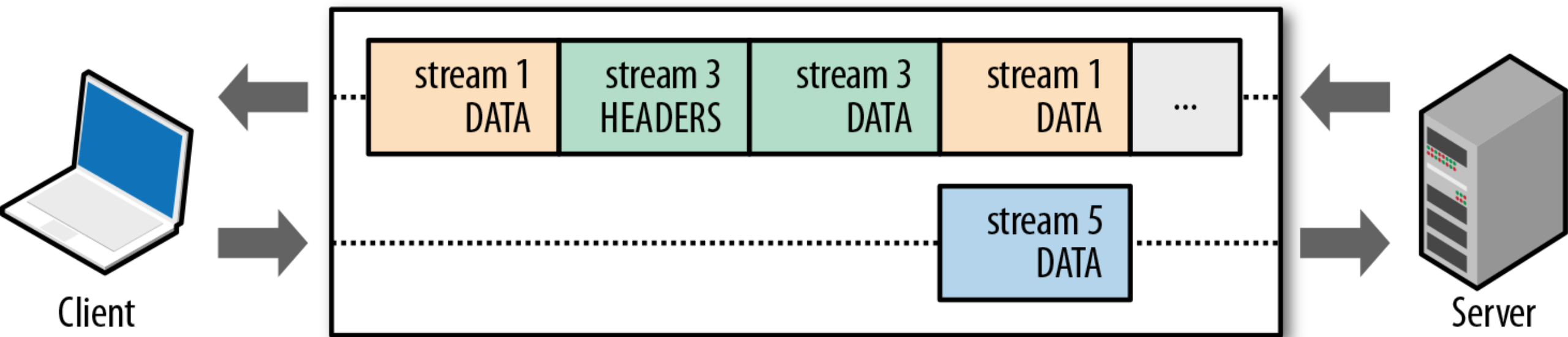


http/2

- В основном стандартизирует spdy от Google
- Компрессия хидеров
- мультиплексирование
- приоритизация
- Сервер пуш
- Ну и прояснить всякие моменты
 - Например длинна сжатого контента

Мультиплексинг, фреймы и сервер-пуш

HTTP 2.0 connection



Линки!

- RTFM: [RFC 2616](#)
- САМАЯ ВАЖНАЯ КНИГА (про 1.1):
HTTP: The Definitive Guide
 - [Amazon](#)
 - [Safari](#)
- [Шаблон «реактор»](#)
- [Даг Ли про NIO](#)

**"Расставашки -
всегда пичалька".**

(с) Сократик

