Java & PostgreSQL

The Past,
The Present
And The Future

Pgconf.RU '16

Álvaro Hernández <*aht@8kdata.com*>

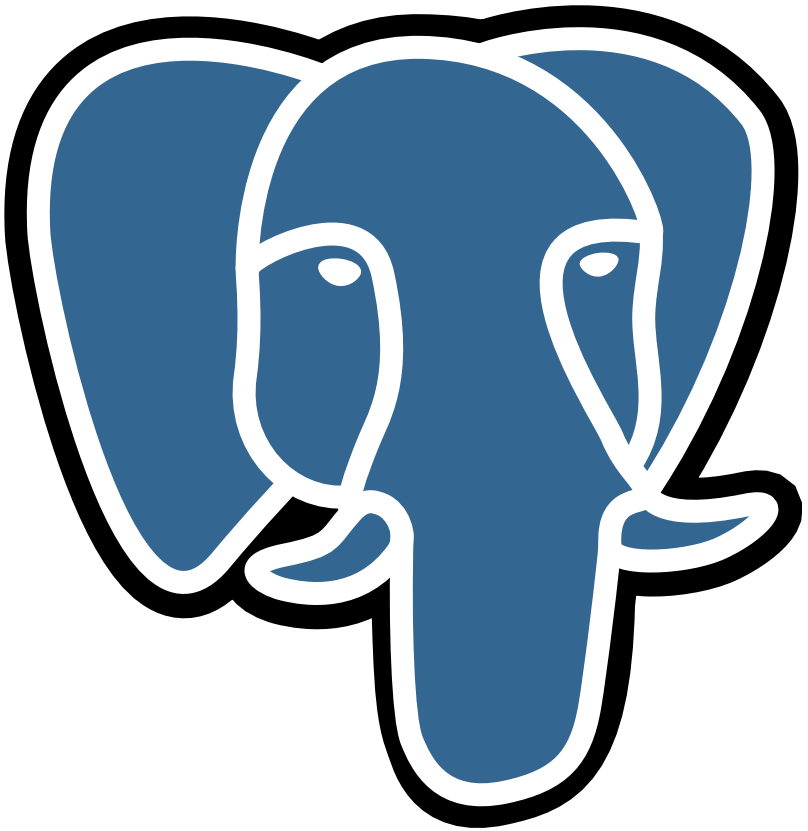
About 8K data

www.8kdata.com

- Research & Development in databases
- Consulting, Training and Support in PostgreSQL
- Founders of PostgreSQL España, 5th largest PUG in the world (>500 members as of today)
- About myself: CTO at 8Kdata:

@ahachete
http://linkd.in/1jhvzQ3

PostgreSQL & Java



https://www.flickr.com/photos/trevi55/296946221/

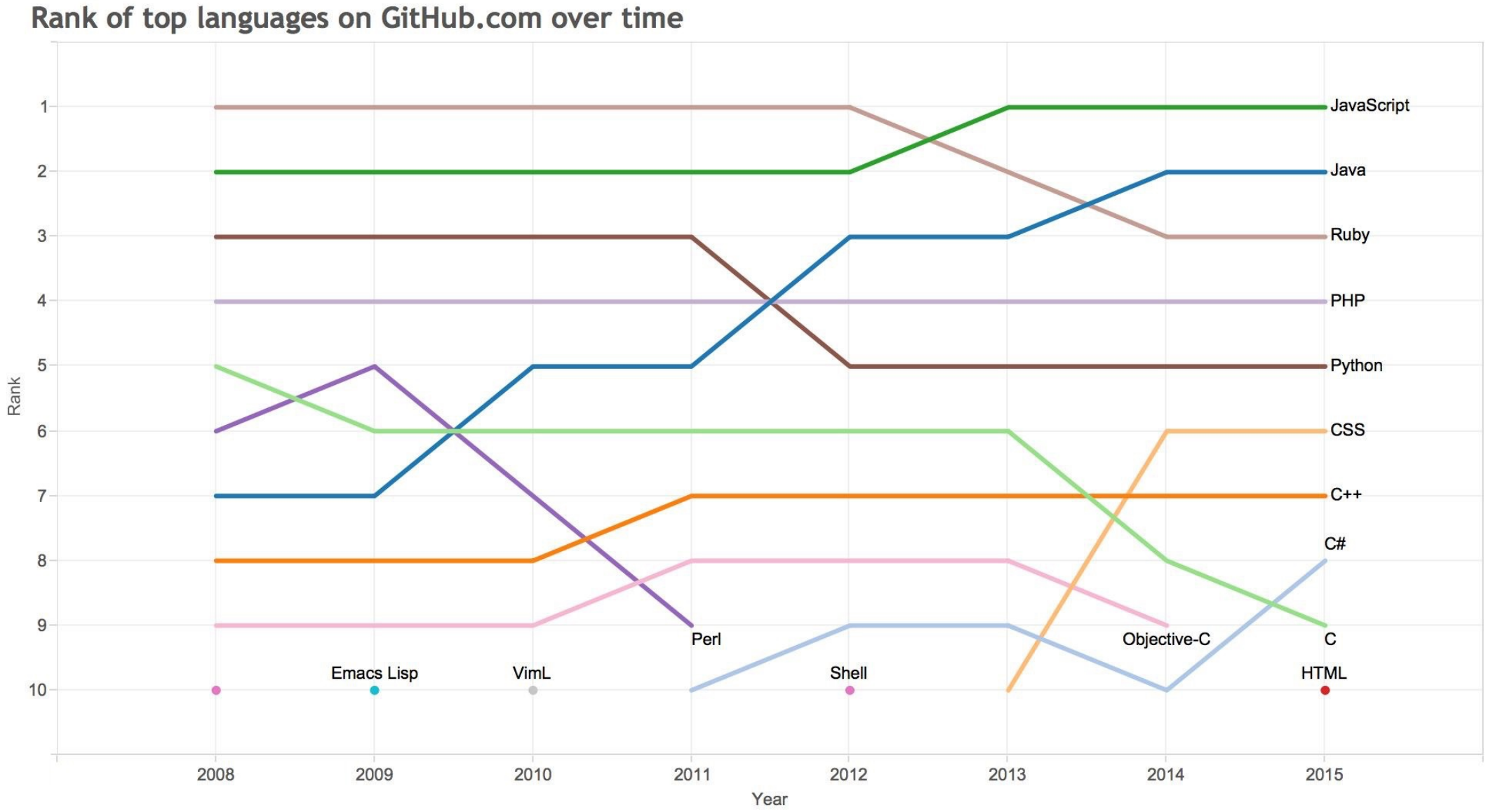
PostgreSQL & Java

- Java IS the enterprise language
- Arguably, there is more Java code accessing PostgreSQL than from any other programming language
- Both Java and PostgreSQL are mature, reliable and trusted

Java: TIOBE index

Feb 2016	Feb 2015	Change	Programming Language	Ratings	Change
1	**2**	↑	**Java**	**21.145%**	**+5.80%**
2	1	↓	C	15.594%	-0.89%
3	3		C++	6.907%	+0.29%
4	5	↑	C#	4.400%	-1.34%
5	8	↑	Python	4.180%	+1.30%
6	7	↑	PHP	2.770%	-0.40%
7	9	↑	Visual Basic .NET	2.454%	+0.43%
8	12	⇈	Perl	2.251%	+0.86%
9	6	↓	JavaScript	2.201%	-1.31%
10	11	↑	Delphi/Object Pascal	2.163%	+0.59%

Java: GitHub popularity



PYPL: Popularity Programming Languages

Rank	Change	Language	Share	Trend
1		Java	24.2 %	+0.3 %
2	↑	Python	11.9 %	+1.2 %
3	↓	PHP	10.7 %	-0.8 %
4		C#	8.9 %	+0.1 %
5		C++	7.6 %	-0.5 %
6		C	7.5 %	+0.1 %
7		Javascript	7.3 %	+0.3 %
8		Objective-C	5.0 %	-0.9 %
9	↑↑	Swift	3.0 %	+0.4 %
10		R	2.9 %	+0.3 %

```
thePast =
new List();
```

PostgreSQL and Java in the past

Java and PostgreSQL haven't mixed well:

- Managed memory vs unmanaged
- Java is (still!) perceived as slow and bloated
- Java requires a runtime (JVM)
- PostgreSQL is ANSI C
- Few Postgres developers like and/or are proficient in Java

JDBC Driver (pgjdbc)

- "Official" driver. Type 4 driver

- Lessons learned:
  - ➔ We were not involved in the JDBC specification
  - ➔ Lack of a rowid is painful
  - ➔ Choosing "?" for PreparedStatement bind variables was a bad choice

pl/java

- Started strong, faded away
- Offers a JDBC API that wraps SPI calls with JNI
- There's no constantly running JVM. No support for saving state, more overhead
- At one point offered support for gcj

pl/java

- Started strong, faded away
- Offers a JDBC API that wraps SPI calls with JNI
- There's no constantly running JVM. No support for saving state, more overhead
- At one point offered support for gcj

pl/j

- Alternative implementation of server-side Java in PostgreSQL
- Followed the approach of a Java server running permanently, and offered JDBC support
- Heated debate vs pl/java regarding inclusion in core

```
thePresent =
new List<JavaTech>();
```

JDBC Driver (pgjdbc)

- Developer base and activity has surged in the last year
- Mavenized!
- Latest versions have significantly improved performance
- Solid, reliable choice

Other drivers

pgjdbc-ng

- Modern driver, requieres Java 7
- Uses Netty for network I/O
- Favors binary over text mode
- Goal of being really fast
- Not on par in terms of features with pgjdbc (notably, lacks COPY)
- Latest release: 0.6 (oct 2015)
- https://github.com/impossibl/pgjdbc-ng/releases

Other drivers

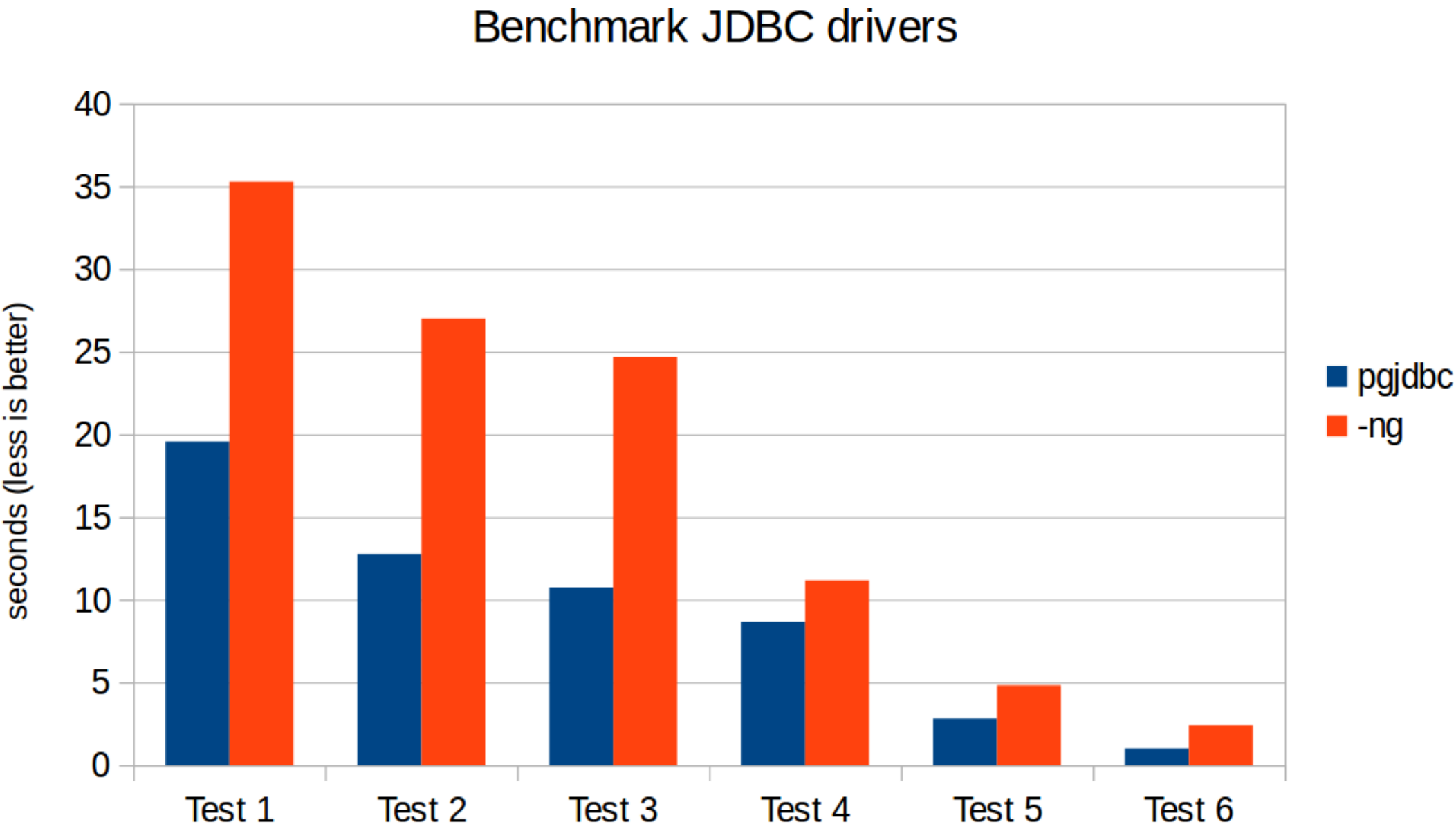
- Progress Type 5 driver
  https://www.progress.com/jdbc/postgresql
  Commercial driver, barely known by community

- PostgreSQL async driver
  https://github.com/mauricio/postgresql-async
  Non-JDBC
  Written in Scala, also supports MySQL
  Netty based
  Active development

Other drivers

- RxJava-jdbc
  https://github.com/davidmoten/rxjava-jdbc
  JDBC generic (not postgres specific)
  All the RxJava goodness!
  Compose queries in serial or parallel
  Map results into tuples or own classes

Benchmark!


Benchmark JDBC drivers
seconds (less is better)
40
35
30
25
20
15
10
5
0
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
pgjdbc
-ng

pl/java

On behalf of Chapman Flack…
Announcing pl/java 1.5 beta!!!!

http://tada.github.io/pljava/releasenotes.html

- Coming back! First release since 2011
- Modernized, more active community
- Works with 9.5, Java 6-8 :)

Current best practices

- Beware of most online tutorials. Most are outdated and code contains errors:
  - ➔ Don't load the driver (Class.forName)
  - ➔ Use try-with-resources
  - ➔ Carefully check exceptions
  - ➔ Use prepared statements

https://www.pgcon.org/2014/schedule/events/713.en.html

(self-plug)

ORMs

Really, don't get me started on this...

jOOQ

```
  SELECT AUTHOR.FIRST_NAME, AUTHOR.LAST_NAME, COUNT(*)
    FROM AUTHOR
    JOIN BOOK ON AUTHOR.ID = BOOK.AUTHOR_ID
   WHERE BOOK.LANGUAGE = 'DE'
     AND BOOK.PUBLISHED > DATE '2008-01-01'
GROUP BY AUTHOR.FIRST_NAME, AUTHOR.LAST_NAME
  HAVING COUNT(*) > 5
ORDER BY AUTHOR.LAST_NAME ASC NULLS FIRST
   LIMIT 2
  OFFSET 1
```

```
create.select(AUTHOR.FIRST_NAME, AUTHOR.LAST_NAME, count())
      .from(AUTHOR)
      .join(BOOK).on(AUTHOR.ID.equal(BOOK.AUTHOR_ID))
      .where(BOOK.LANGUAGE.eq("DE"))
      .and(BOOK.PUBLISHED.gt(date("2008-01-01")))
      .groupBy(AUTHOR.FIRST_NAME, AUTHOR.LAST_NAME)
      .having(count().gt(5))
      .orderBy(AUTHOR.LAST_NAME.asc().nullsFirst())
      .limit(2)
      .offset(1)
```

```
theFuture =
new List<Future<?>>();
```

Some predictions

- Expect more new features and performance improvements from pgjdbc, pgjdbc-ng
- Binary support for jsonb in the protocol!
- pl/java renaissance
- pl/j comeback?

Phoebe (WIP)

- New PostgreSQL driver
- Async & Reactive by design. RxJava based
- Targets clusters, not only individual servers
- Netty-based, async off-heap I/O

Phoebe (WIP)

Expected features:

- Binary mode
- Unix Domain Sockets
- Logical decoding
- Query pipelining
- Fully asynchronous operation
- Execute query on rw or ro nodes
- Fluent-style API
- Compatible with Java >= 6

Phoebe (WIP)

Current API design:

```
RxPostgresClient client = RxPostgresClient
                .create()
                .tcpIp("::1", 5432)
                .tcpIp("localhost", 5433)
                .allHosts()
                .init();
client.onConnectedObservable().subscribe(
    c -> System.out.println(c)
  );
```


8K
data