# **Overview**

The “parser” application is an implementation of the Wallethub test task on a position of a Java developer. The application is written in plain Java with a help of the [Maven](https://maven.apache.org/) build tool.

# **Model**

The application’s model includes classes LogEntry and IpStatistics.

## **LogEntry**:

Defines the following properties of a line in a log:

* Date;
* IP Address;
* HTTP Request;
* Response code;
* Browser;

These fields form a line in a log with “|” as a delimiter.

## **IpStatistics**:

Defines statistics of IP addresses occurrences in a log filtered by the application’s logic:

* IP Address;
* Count;

# **Services**

The application consists of several service classes each implementing its role.

## **ApplicationProperties**

Reads application’s parameters defined in the resourses/application.properties file. This approach (instead of hard-coding them in the source code) lets a user modify settings without rebuilding the application. So far, only DB connection properties are used.

## **CommandLineProcessor**

Validates command line parameters – “accesslog”, “startDate”, “duration” and “threshold” – provided by a user to the main Parser class. Utilizes the [commons-cli](https://commons.apache.org/proper/commons-cli/) Apache library.

## **LogReader**

Reads an access log line by line and returns the next LogEntry object on demand.

## **LogEntryDao**

Provides DAO operations with LogEntity class objects to the “log\_entities” DB table

## **IpStatisticsDao**

Provides DAO operations with IpStatistics class objects to the “ip\_statistics” DB table

# **Main Application**

The class Parser is the application’s entry point containing the static void main() method. It works due to the following algorithm:

1. Read application properties.
2. Parse command line parameters
3. Establish a connection to the DB.
4. Clear up the “log\_entities” table.
5. Open the access log and read entries one by one. With each 1000 log entries, store them in bulk to the “log\_entities” table (much faster, than make a separate transaction for each record).
6. Perform an SQL query gathering IP address statistics according to the provided command line parameters. Print each IP address statistics result and store it the ‘ip\_statistics’ table in the database.

# **Database**

To prepare a database, one must run scripts from the docs/create\_db.sql file from a MySQL console. Also, MySQL connection properties must be provided in the resources/application.properties file.

# **Build and Deployment**

“parse” is a Maven based application. To build, one must run the “mvn clean package” command – it will clear up, compile and build a jar file in the target/ directory.

The application can be launched as described in the task document, i.e.:

java -cp target/parser.jar com.ef.Parser --accesslog=/d/programming/interviews/WalletHub/access.log --startDate=2017-01-01.15:00:00 --duration=hourly --threshold=200