Spring Boot Microservices

Beginner to Guru

JSON with Spring Boot

JavaScript Object Notation

- JSON is a lightweight data-interchange format.
- Based on the JavaScript Programming Language ECMA-262 - December 1999
- JSON is now independent of JavaScript
- Current Standard ECMA-404 - December 2017
- JSON can be read and created by all major programming languages
- JSON is an ideal data-interchange format because of its wide-spread adoption

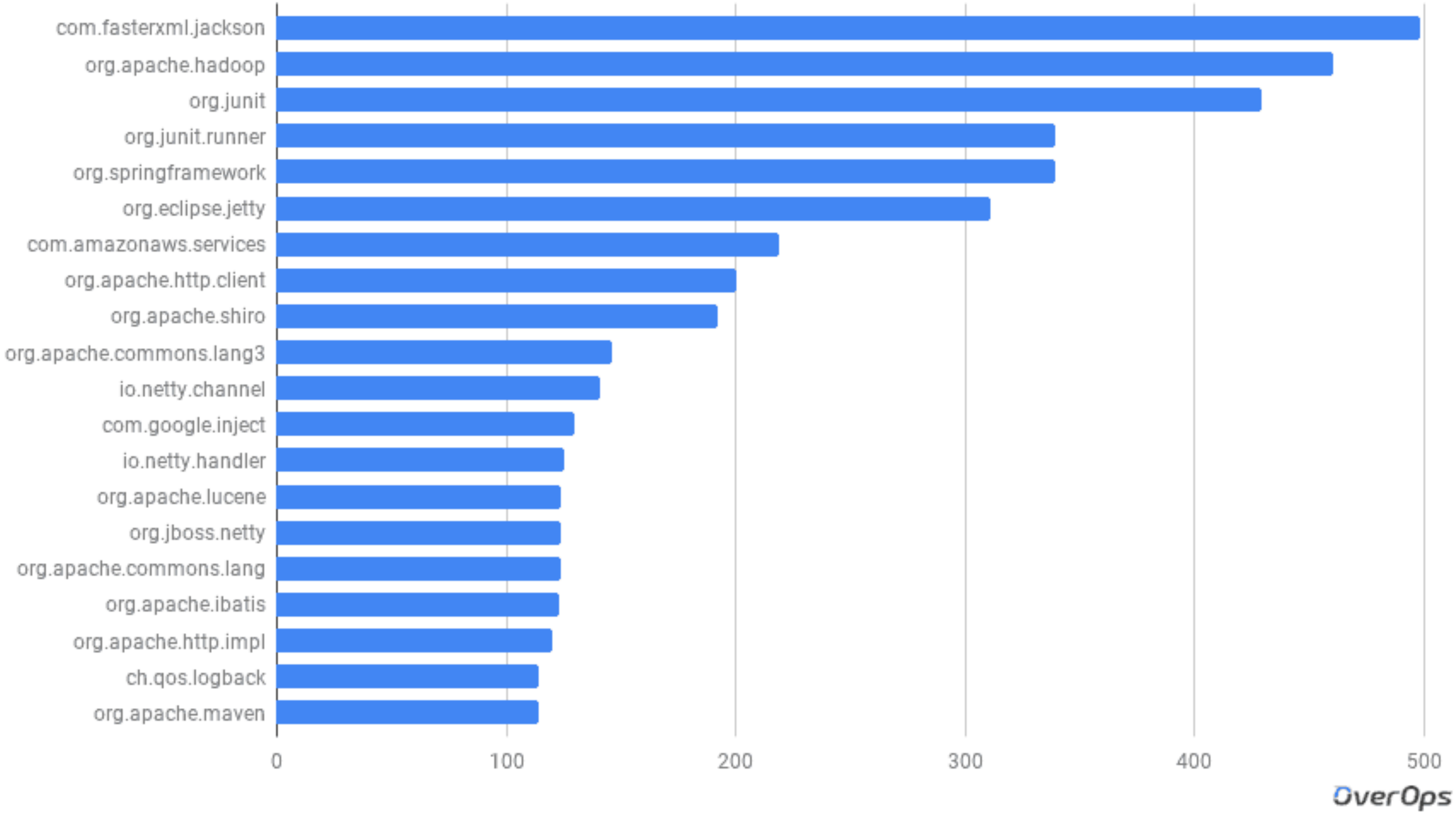
Processing JSON

- **Serialization** - is the process of converting a Java object to a JSON object
- **De-serialization** - is the process of converting a JSON object into a Java object
  - The JSON is typically a string, possibly a buffer (web request) or a file from the file system
- In an HTTP **GET** action, Spring will read data into POJOs, serialize the POJO into a JSON string, then deliver the JSON payload to the client.
- In an HTTP **POST** action, the client will be using HTTP to post, Spring will read the body of the request and De-serialize the JSON payload into a Java POJO
- Can also be used for messaging

Spring Boot JSON Libraries

- Spring Boot supports 3 JSON mapping libraries
  - **Jackson** - ***Spring Boot Default*** - most popular
  - **GSON** - Google's library, very popular
  - **JSON-B** - JEE API Standard (JSR 367)
    - Providing an API standard for JSON binding
    - Adoption is unclear, momentum seems weak in the Java community
  - Spring Boot will auto configure any of these 3 libraries

Top 20 Java Libraries - 2018



Performance and Security

- **Performance**
  - Jackson tends to benchmark very well
  - GSON benchmarks well also, beating Jackson in some aspects
- **Security**
  - Keep an eye on security alerts, can be problematic
- **Other**
  - Jackson is popular - there is a lot of free help on the internet

SPRING FRAMEWORK

GURU