Globant

javamelody

monitoring of JavaEE applications

And for Springboot

What is it and what is this for?

Performance and Resource usage

- Indicators with summary charts:
  - Number of executions, mean execution times and percentage of errors of http requests, sql requests, jsf actions, struts actions, jsp pages or methods of business façades (EJB’s).
  - Java memory
  - Java CPU
  - Number of user sessions
  - Number of jdbc connections
- Notifications:
  - No alerts!
  - Notification mails (daily, weekly, etc.)
- External API:
  - An XML and JSON API to retrieve the info in HTML, PDF, PNG

How do I apply it to my application?

If you have `web.xml` version 3.0 or later, you xonly need to add this dependency in your `build.gradle` (or `pom.xml` if you prefer the complex way):

```
dependencies {

    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.postgresql:postgresql'

    providedCompile group: 'javax.servlet', name: 'javax.servlet-api', version: '3.0.1'
    implementation group: 'net.bull.javamelody', name: 'javamelody-spring-boot-starter', version: '1.83.0'
    testCompile group: 'junit', name: 'junit', version: '4.12'

}
```

And for Springboot?

What does it does?

- JavaMelody monitors Spring beans having an annotation
  - `@Controller`, `@Service`, `@Repository`,
  - `@RestController`,
  - `@Async`,
  - `@FeignClient`,
- and methods having an annotation:
  - `@Async`,
  - `@Scheduled` or `@Schedules` are automatically monitored.
- If you want to monitor method calls on some Spring beans (`@Beans`), you can add `@MonitoredWithSpring` on those classes or methods.

@Beans no son monitoreados por defecto.

Why is Melody lightweight?

- **No profiling**: no classes are instrumented, instead, it uses `Interceptors`.
- **No persistence**:
  - no persistence storage is used, so no overhead on I/O.
  - From time to time the statistics are backed up on a file.

- Tasks that make overhead:
  - Launching
  - Garbage collection
  - And Head dump.

What next?

- Should we turn it off on prod?
  - *I don't think it is good idea to switch off javamelody as most useful are historical data measured before you run into any problem and how would you know without monitoring tool there is problem?*
- There are some examples already available to copy/paste, or to learn from them in this @github_repository.
- There are currently some planes to add persintency to the framework.

Practical Links & References

Links

- Java Melody Official Wiki.
- Java Melody Official user guide.
- About alerts and API of Java Melody.
- Java Melody demo.

And that's all!