Visualize web thread thread pool

(for spring project based on tomcat server)

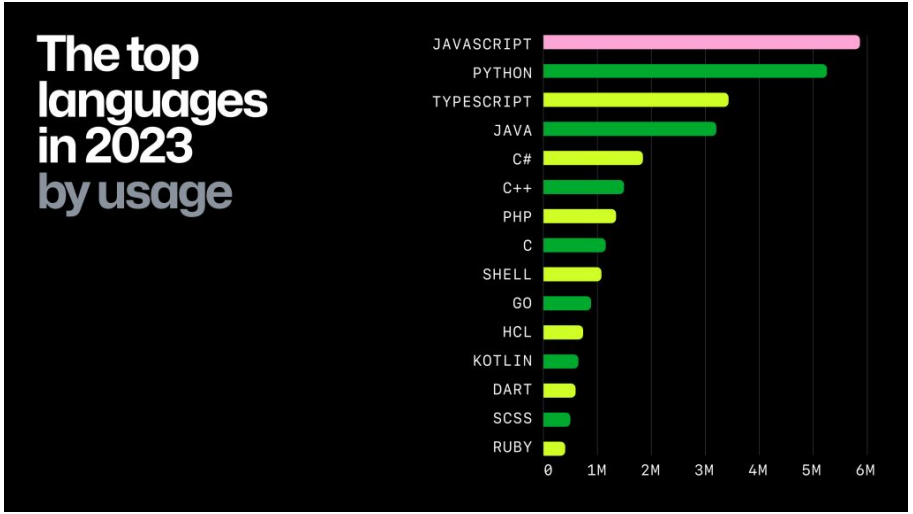
2019314700 한용준 2019310655 윤재환 2019312756
김준영 2018311338 김경현

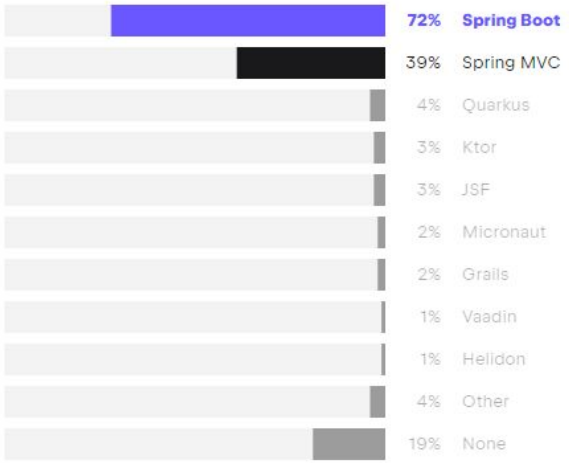
성균관대학교
SUNG KYUN KWAN UNIVERSITY(SKKU)

1. Objective

1. Java and Spring
2. Tomcat Server in Spring
3. Thread Pool
4. Issue with Existing Tools
5. Objective: Visualizing the Thread Pool

Java and Spring

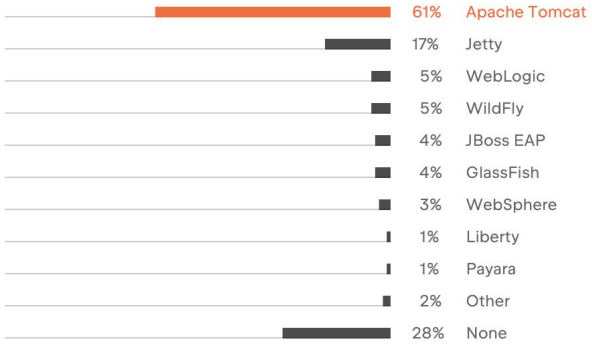




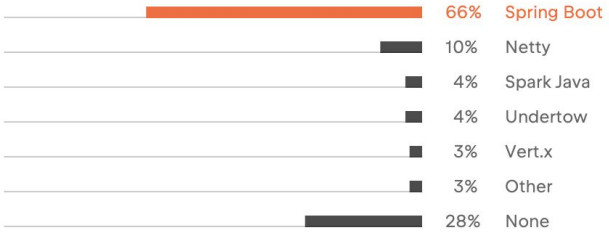
https://github.blog/news-insights/research/the-state-of-open-source-and-ai/
https://www.jetbrains.com/lp/devec

Java and Spring

What application servers do you regularly?



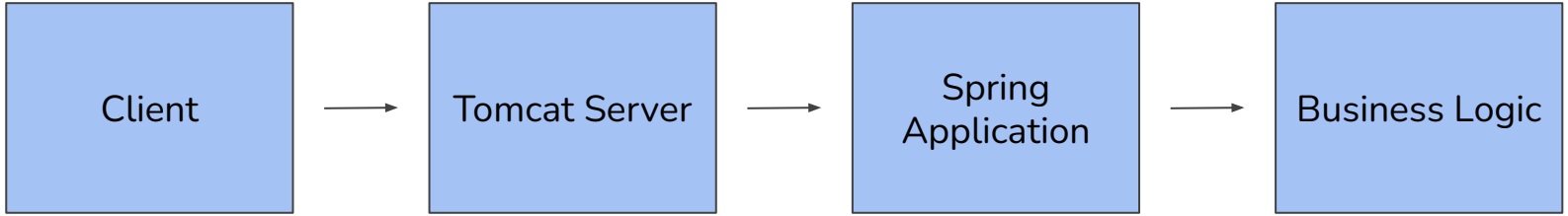
Which frameworks do you use as an alternative to an application server, if any?



https://www.jetbrains.com/lp/devecosystem-2021/java/

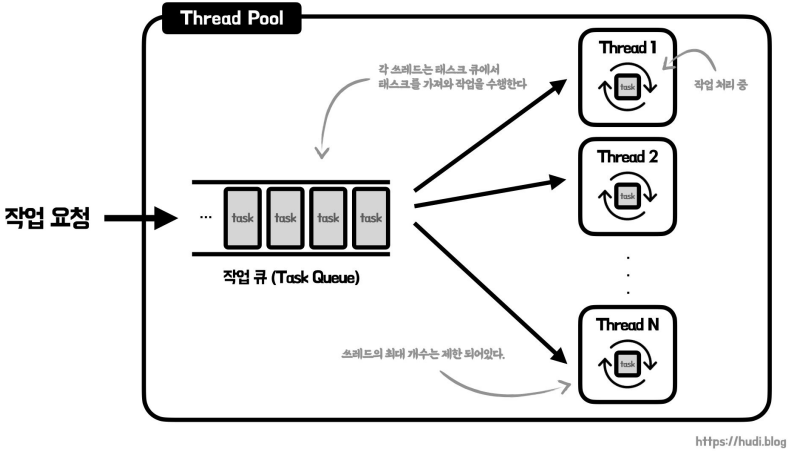
Tomcat Server in Spring

The process of handling HTTP requests in a Spring application :



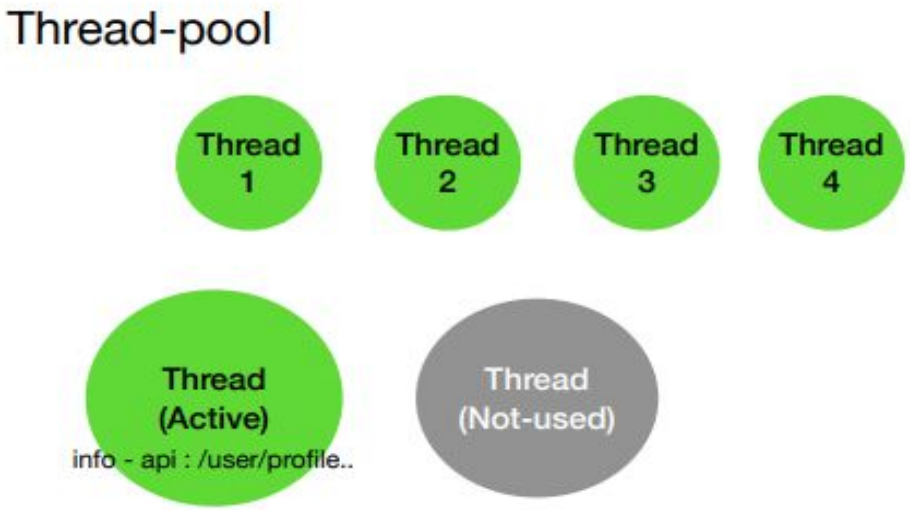
Thread Pool

- Reduces the overhead of creating new threads
- Prevents system overhead by limiting the number of concurrently running threads
- Ensures efficient resource management and faster task processing



Issues with Existing Tools

Objective: Visualizing the Thread Pool



Milestone

Weeks	2 3	4 5	6 7	8 9	10 11	12 13	14 15
Define Problem	O O	O					
Tech analysis and Study		O O	O				
Implement data collecting methods			O O				
deploy as library				O .			
develop web application				. O	O O	O	
UI/UX Design						O O	
Testing							O

2. Role of each member

Studying for implementation, previous works (how Spring works, how to implement) : All

UI/UX : 김준영

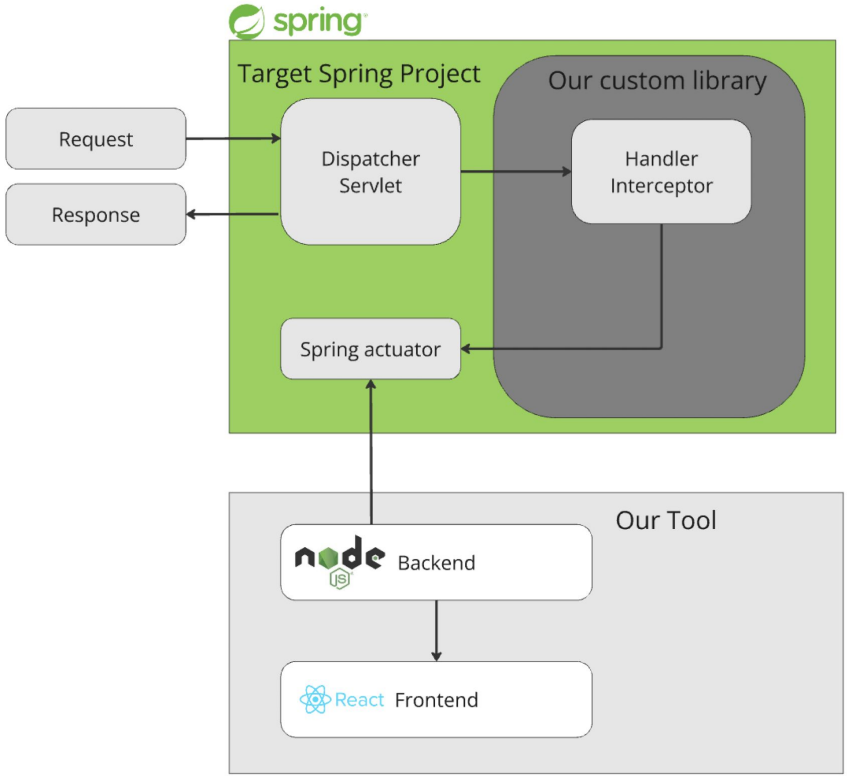
Backend : 김준영

Frontend : 윤재환, 김경현, 한용준

Custom Library : 한용준, 윤재환, 김준영

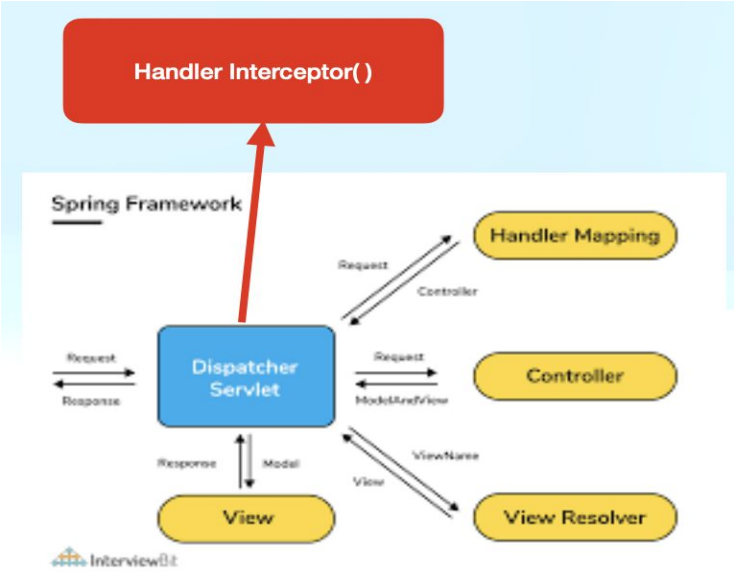
Implementation(한용준)

1. How to get data of each threads? (Library)
2. How to visualize(Backend, Frontend & UI/UX)



1. How to get data for each threads?

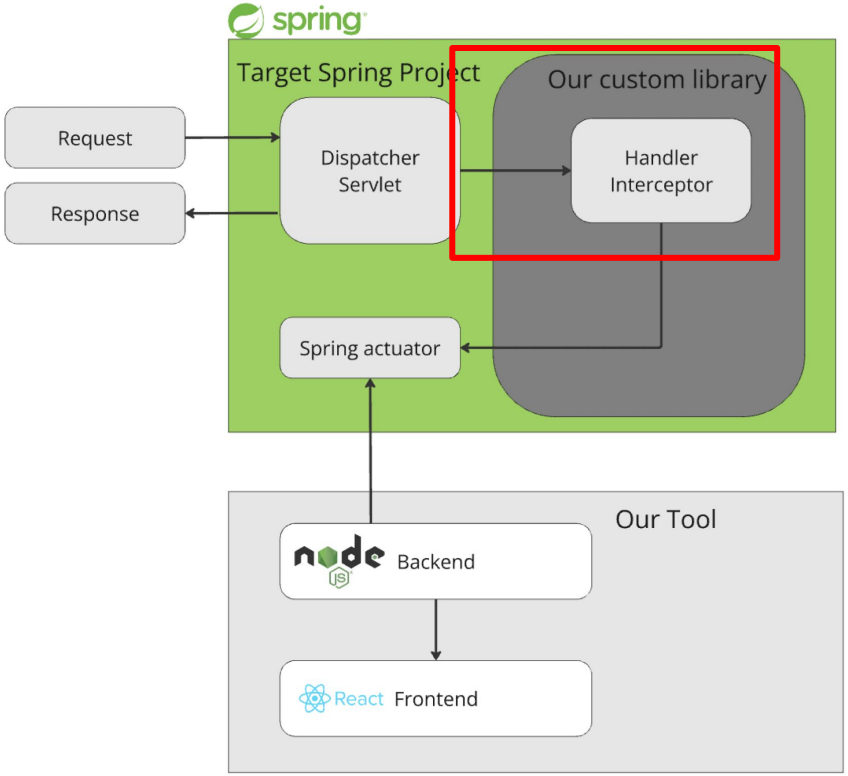
Dispatcher Servlet and Handler Interceptor()



1. How to get data for each threads?

by overriding Handler Interceptor(),

we will calculate memory usage of each threads, and execution Time (Response Time - Request Time) + record uri etc.



1. How to get data for each threads?

Modifier and Type	Method	Description
default void	afterCompletion()	call back after completion of request processing
default void	postHandle()	Interception point after successful execution of a handler
default boolean	preHandle()	Interception point before the execution of a handler(or buisness logic)

end time record, memory usage, uri

start time record

https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/servlet/HandlerInterceptor.html

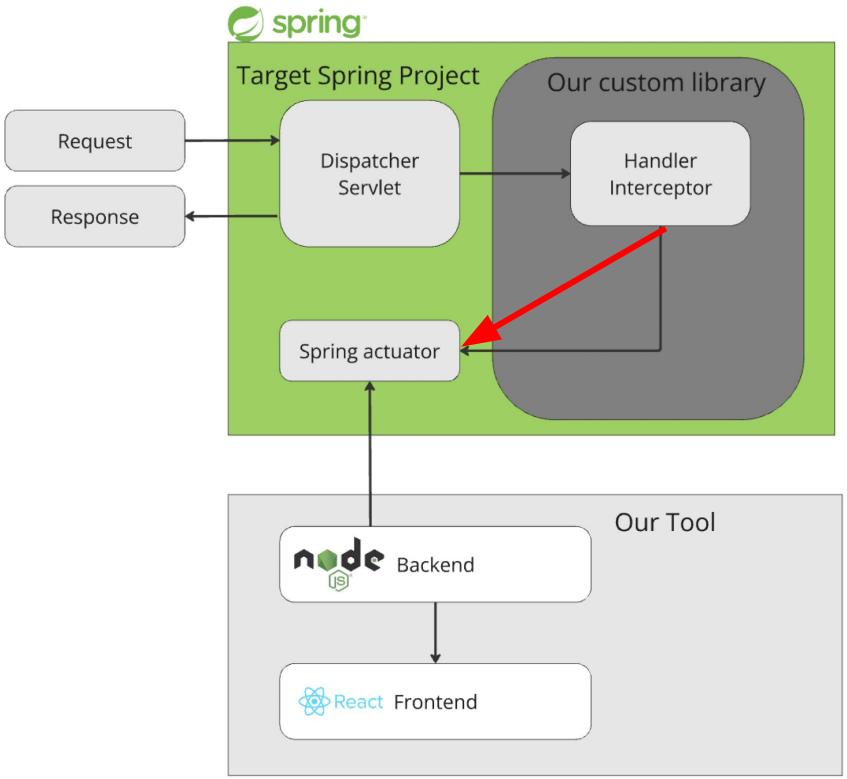
1. How to get data for each threads?

Spring Actuator is a library to help

monitor project built with Spring

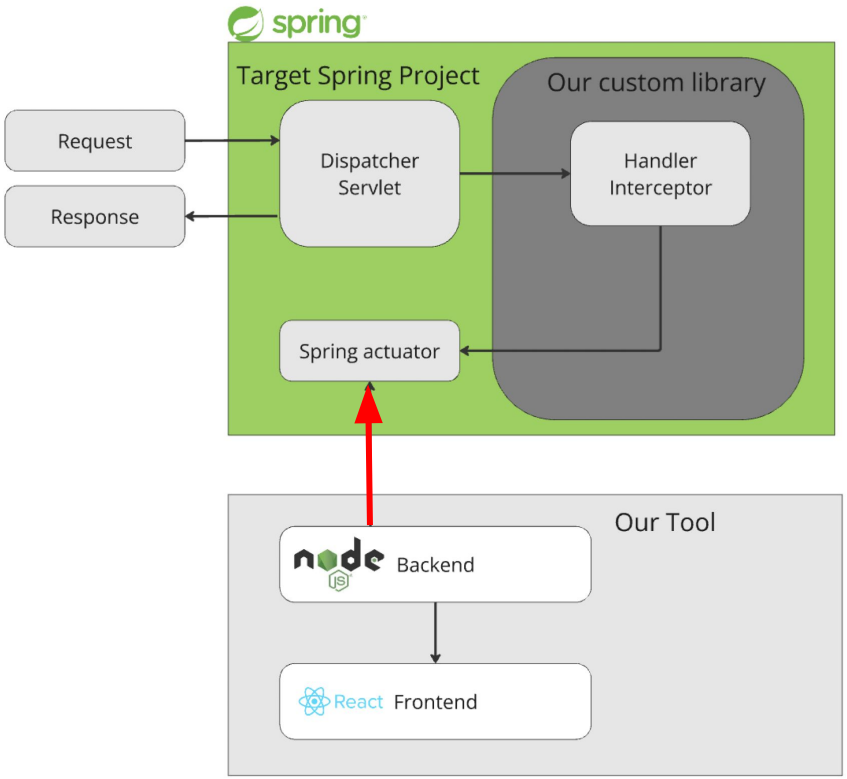
Spring Actuator expose metrics to certain endpoint

We load our data collected by our custom HandlerInterceptor

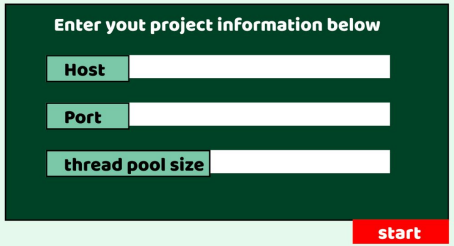


1. How to get data for each threads?

get data from Spring actuator endpoint through polling



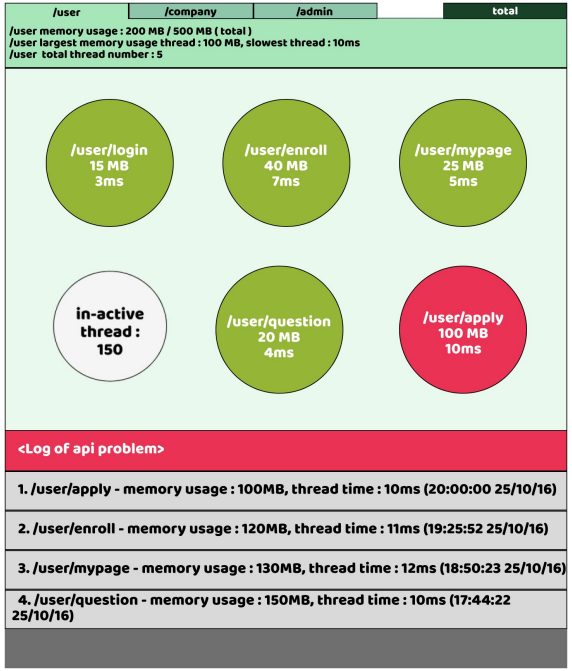
2. How to visualize?

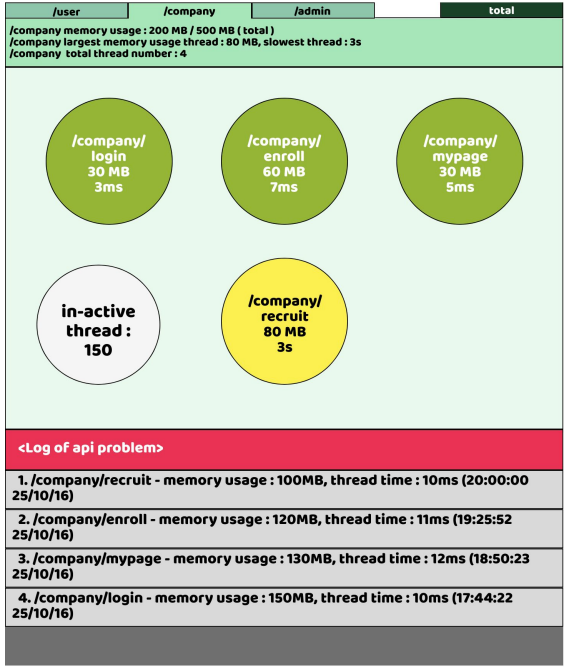


- host input: Insert your host name.
- port input: Insert your port number.
- thread pool size input: Decide project's thread pool size.

- start button: If you put all of the inputs, you can go inside our visualization program.

2. How to visualize?

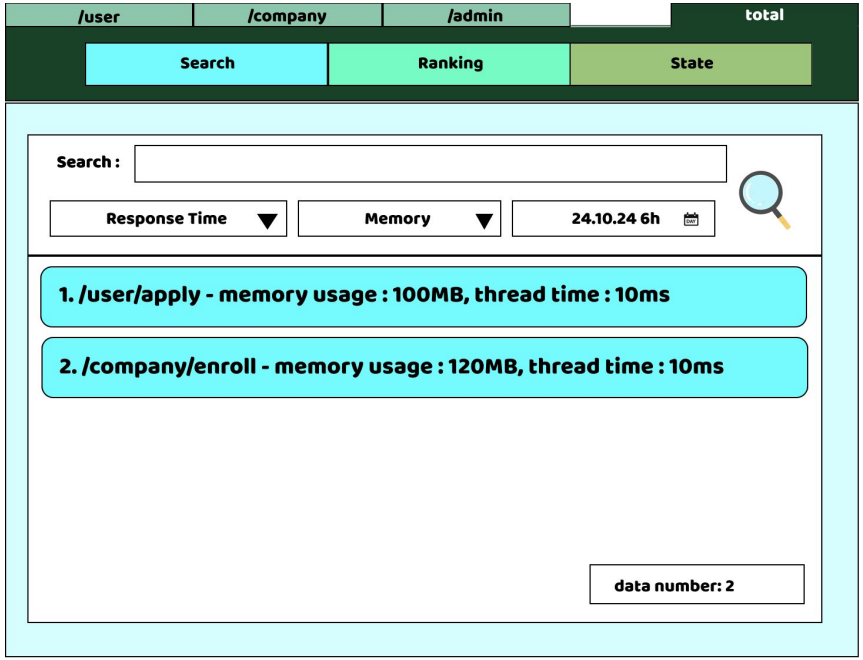




- /user & /company & /admin tab: Separate api visualization into endpoint
- api information: memory usage, total number, response time

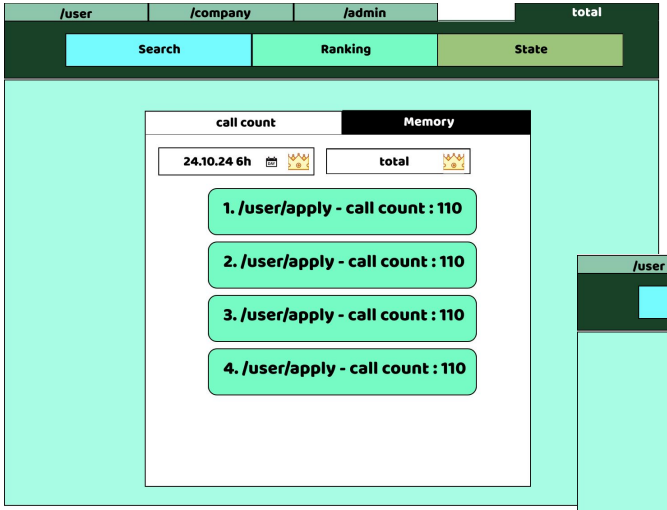
- log of api problems: leave log of api alerted by hour

2. How to visualize?

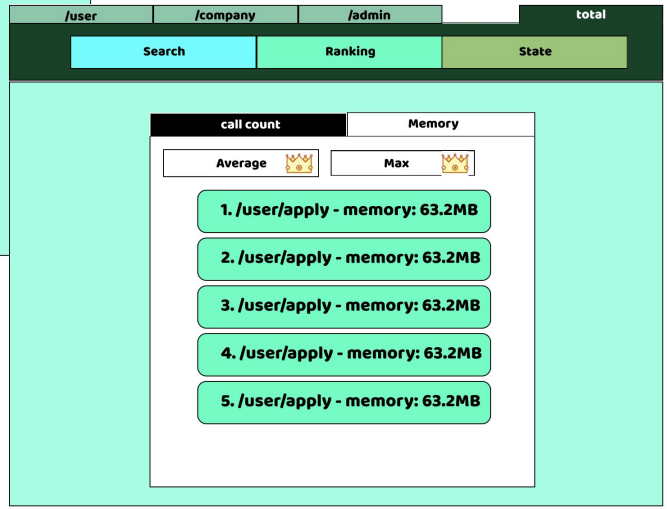


- total tab
  - Search tab (search uri)
  - Ranking tab
  - error tab
- response time toggle
  - Reaction time in large order or sorting in small order
- memory toggle
  - Memory usage in large order or sorted in small order
- hour toggle
  - decide running time
- magnifier button: search function
- data number: number of search results

2. How to visualize?



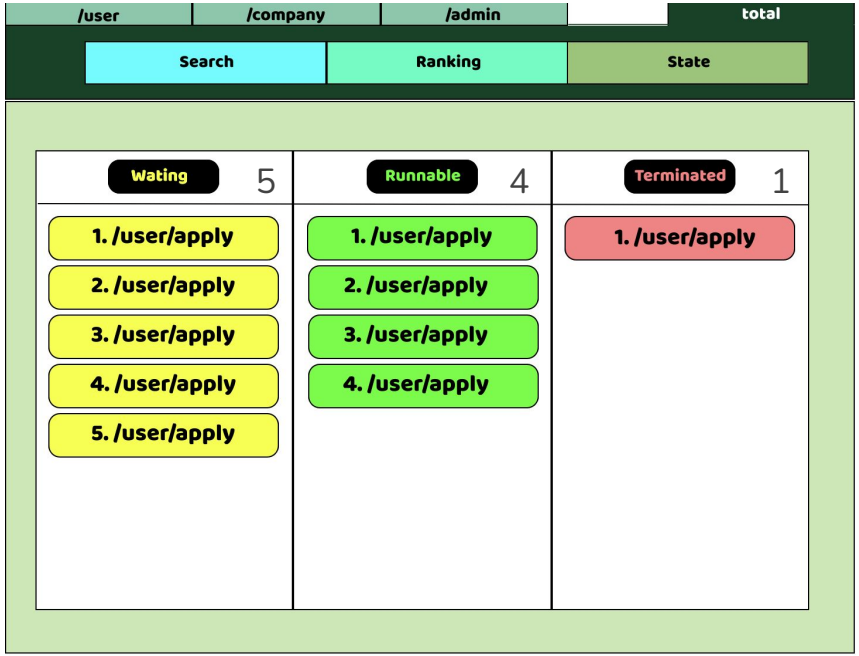
- call count tab
- memory tab



- call count of specific time or total time
- average memory usage or Max memory usage

- If you click the crown, you can see ranking.

2. How to visualize?



- Wating state
- Runnable state
- Terminated state
- You can recognize the number of api corresponding to the state.

Implementation - Custom Library

Our tool is targeting projects which are built with Java Spring framework.

Implementation - Backend

Based on single threaded nature , lightweight

cons of nodejs : CPU-heavy tasks (like complex calculations or image processing) can block other requests.

our backend just does routing against only one client

Implementation(한용준)

Component based architecture

virtual DOM

Our service requires :

lots of similar components

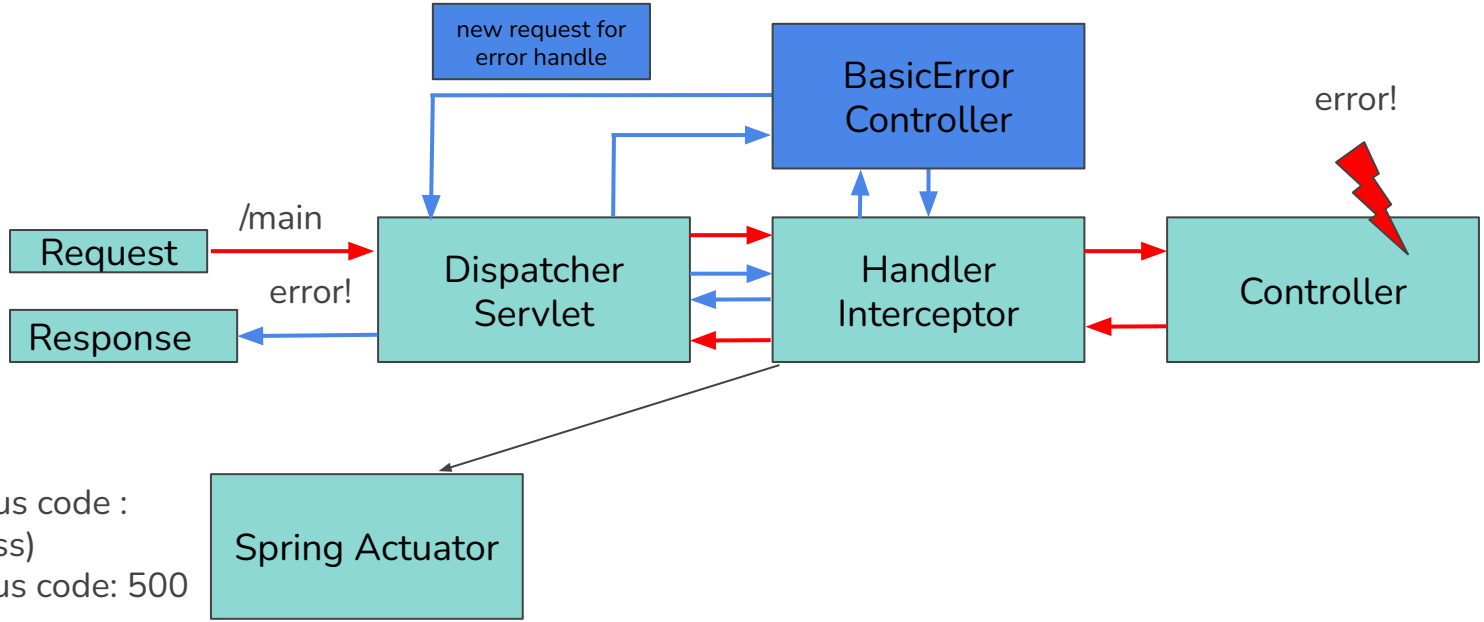
lots of update for web page in short term

Challenges : if error occurs in controller or service layer (internal server error)

Situation : request 1 - uri : /main
request 2 - uri : /login

memory usage : [6080856,57552,1000]
uri : [/main, /login, /error]

Challenges : Redirected to uri:/error where the error occurred (creating a new requested object) and cannot be determined from which uri the error occurred./error

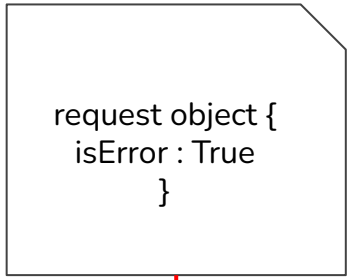


/main : status code : 200 (success)
/error : status code: 500 (error)

Solution : case1 : no error

preHandle()

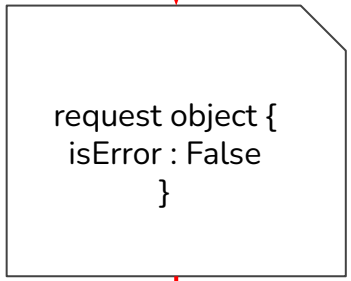
1. create Attribute(field)on request object : isError (Boolean)



this function is called only when request is completed without error

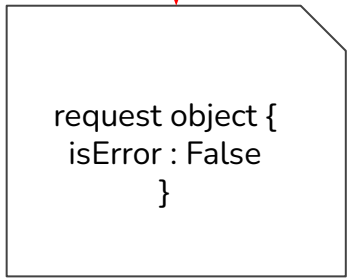
postHandle()

2. set isError to False



afterCompletion()

3. decide this request has no error



Solution : case2 : error

preHandle()

1. create Attribute(field)on request object : isError (Boolean)

request object {
isError : True
}

this function is called only when request is completed without error

postHandle()

2. set isError to False

afterCompletion()

2. decide this request has error

request object {
isError : True
}

Challenges :
result :

상황 : /main 요청 -> 에러 : actuator endpoint :

memory usage : [6080856,57552]
uri : [/main, /login]
isError : [“yes” , “no”]

Challenges: diverse functions

Need more than only per-thread visualization

What can we do with other data?

Time, total calls, memory usage, thread state, etc.

+ We aim for legible analysis → no tiring graphs, more shapes and colors

→ Search for active API at specific time, listed in ascending order regarding memory usage

→ Ranking of total calls over an interval

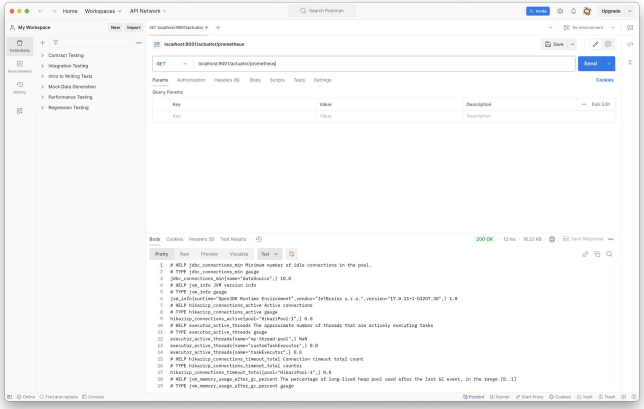
→ Difference on thread state ratio over time

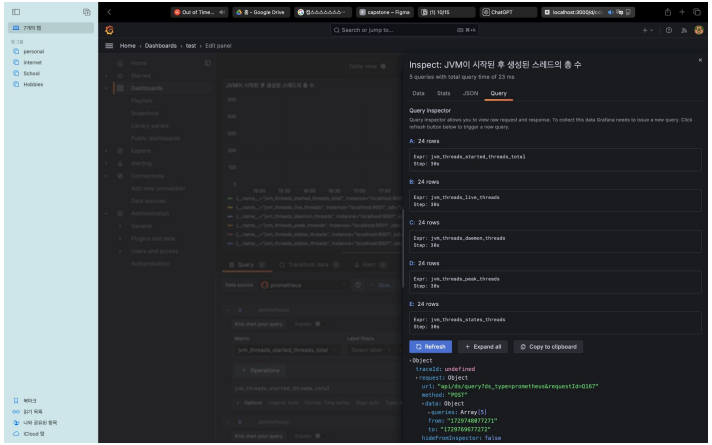
Challenges upcoming

How do we separate and rank uri as we want with data and differentiate new data from incremental data?

Limitation

1. Limited to monitoring : can’t actually change target projects through our tool.
2. Performance Degradation





Thank you