

# Visualize Web Threadpool

(for spring project based on tomcat server)

# What we have done for last two weeks

1. Consideration of exceptional situations
2. Performance degradation

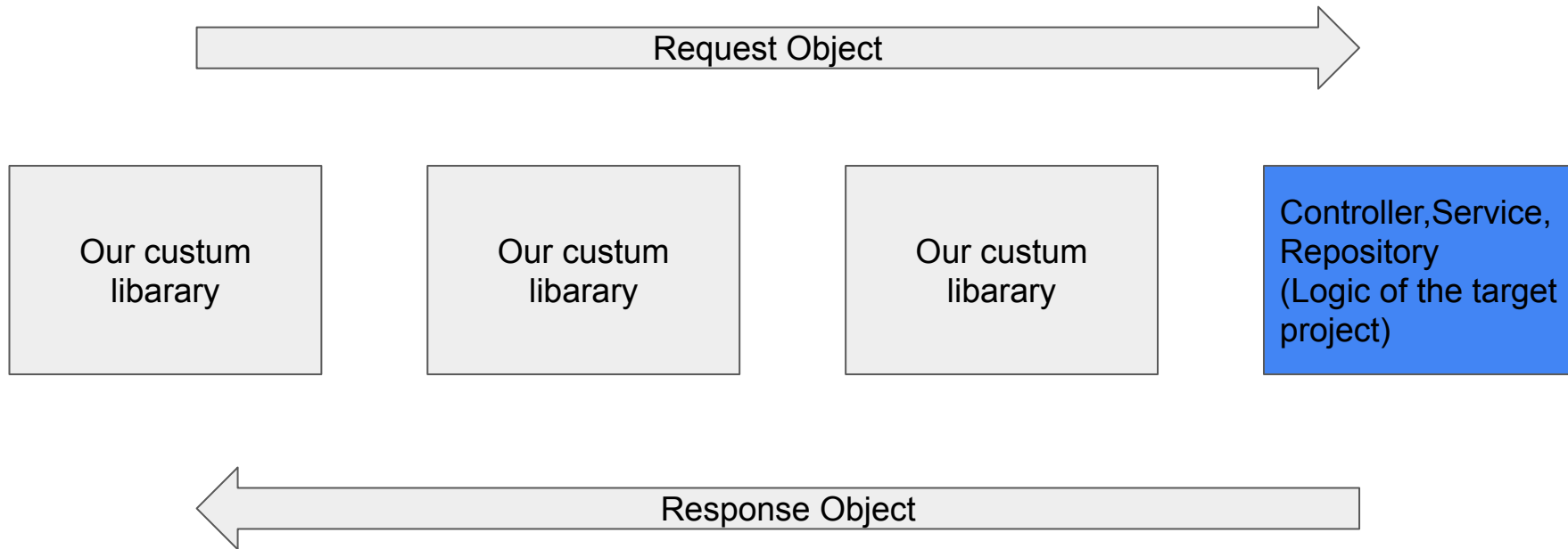
Weeks	2 3	4 5	6 7	8 9	10 11	12 13	14 15
Define Problem	O O	O					
Tech analysis and Study		O O	O				
Implement data collecting methods			O O				
deploy as library				O .			
develop web application				. O	O O	O	
UI/UX Design						O O	
Testing							O

Table 5: Weekly Schedule

# Consideration of exceptional situations

1. Multi-thread api and Asynchronous task
2. Error occurred in the handler interceptor
3. Thread is in sleep state

# Multi-thread api and Asynchronous task



# Situation that error occurs in handler interceptor

Write error handling code in each handler interceptor method

- `preHandle()`
- `postHandle()`
- `afterCompletion()`

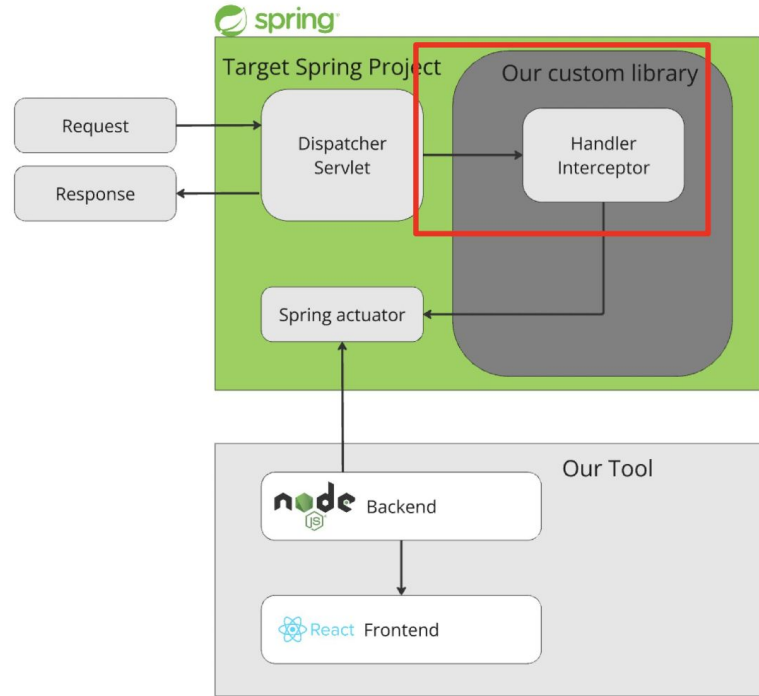
## Situation when thread is in sleep state



result :  $\text{executionTime}(\text{ms}) = 5 + 500 + 7$

# Performance Degradation

only component which affect the target project is customHandlerInterceptor



# Performance Degradation

Single Request Test (100times) (unit : ms)

sample api	before (avg)	before(min )	before(max )	after(avg)	after(min)	after(max)
1	167	134	200	158(-8)	131(-3)	185(-15)
2	75	41	131	78(+3)	39(-2)	212(+91)
3	2	2	2	7(+5)	7(+5)	9(+2)
4	5	2	7	2(-3)	1(-1)	4(-3)
5	10016	10008	10028	10040(+24)	10008(0)	10108(+80)



# Performance Degradation

Load Test ( 1000 requests in 1 sec)

sample api	before (avg)	before(min )	before(max)	after(avg)	after(min)	after(max)
1	7407	150	15308	7405(-2)	277(+127)	14889(-419)
2	2047	37	4872	2085(+38)	67(+30)	4829(-48)
3	0	0	15	1(+1)	0(0)	15(0)
4	1	0	18	1(0)	0(0)	18(0)
5	29556	10001	53573	29620(+64)	10001(0)	49266(-4370)

# Backend: Start URL

- (POST) localhost:9000/api, request body: port, host, threadPoolSize
  - api of getting actuator metrics of threads in target spring program
  - response body

```
{ "data" :  
  [  
    {  
      "uri" : "String",  
      "memoryUsage": "String",  
      "executionTime" : "String",  
      "time" : "String",  
      "isError" : "Boolean",  
      "calledNum" : "String"  
    },  
    {  
      "uri" : "String",  
      "memoryUsage": "String",  
      "executionTime" : "String",  
      "time" : "String",  
      "isError" : "Boolean",  
      "calledNum" : "String"  
    }  
  ]  
}
```

# Backend: Sort&Search URL

- (GET) localhost:9000/api, request param: search, date, time, executionTime, memoryUsage
  - api of getting actuator metrics of threads sorted and searched in target spring program
  - response body

```
{ "data" :  
  [  
    {  
      "uri" : "String",  
      "memoryUsage": "String",  
      "executionTime" : "String",  
      "time" : "String",  
      "isError" : "Boolean",  
      "calledNum" : "String"  
    },  
    {  
      "uri" : "String",  
      "memoryUsage": "String",  
      "executionTime" : "String",  
      "time" : "String",  
      "isError" : "Boolean",  
      "calledNum" : "String"  
    }  
  ]  
  "number": 23  
}
```

# Backend: Rank URL

- (GET) localhost:9000/api/rank, request param: title(memoryUsage or callCount), date, time
  - api of getting actuator metrics of threads rank in target spring program
  - response body

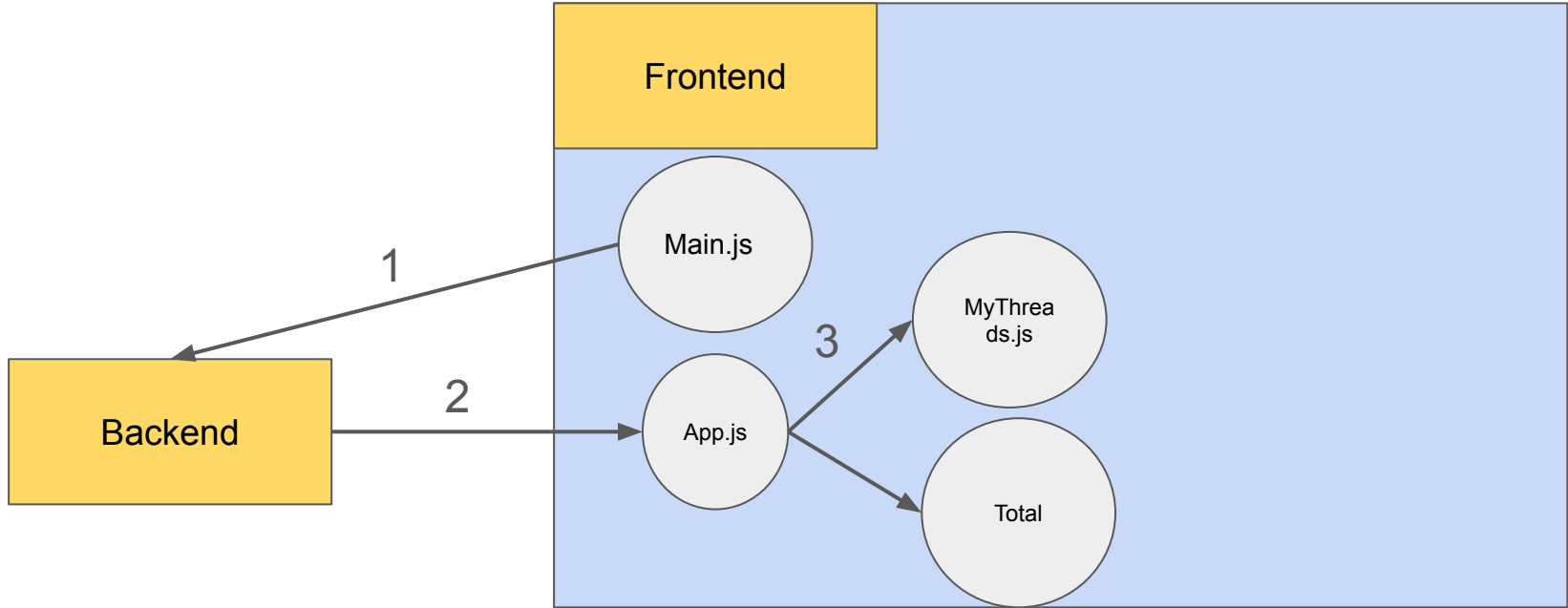
```
{ "data" :  
  [  
    {  
      "uri" : "String",  
      "memoryUsage": "String",  
      "executionTime" : "String",  
      "time" : "String",  
      "isError" : "Boolean",  
      "calledNum" : "String"  
    },  
    {  
      "uri" : "String",  
      "memoryUsage": "String",  
      "executionTime" : "String",  
      "time" : "String",  
      "isError" : "Boolean",  
      "calledNum" : "String"  
    }  
  ]  
}
```

# Backend: State URL

- (GET) localhost:9000/api/state
  - api of getting actuator metrics of threads state in target spring program
  - response body

```
{  
  "waitingCount": 11,  
  "runnableCount": 10,  
  "timedWaitingCount": 4,  
  "terminatedCount": 0  
}
```

# Frontend Structure



## Frontend : Main page

**Monitor your Thread pool**

Enter your Host

Enter your Port

Enter your thread number

**see your threads!**

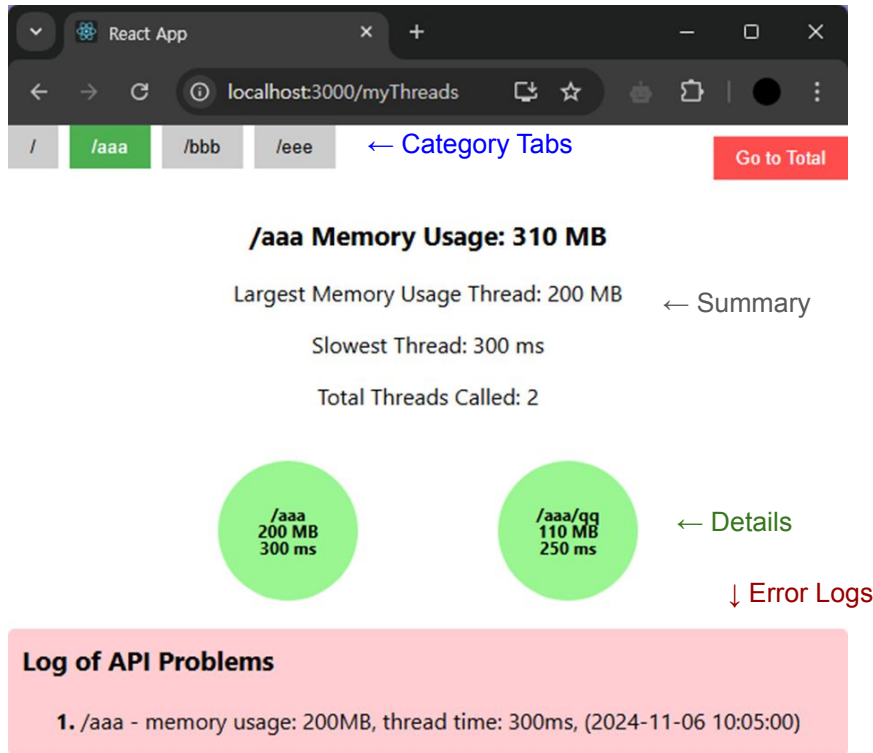
# Frontend: /myThreads

**Category Tabs:** Split uri on first '/', each tab includes the following:

- Summary: Total memory, largest memory, slowest execution...
- **Details:** /uri, memory, execution time
- **Error Logs:** /uri with time specified

To do:

- Idle thread count
- Differentiate thread status with colors
- Accumulate error logs
- Modify the design
- ...





# Frontend: Total Page

## Ranking

Call Count Ranking

Memory Usage Ranking

Average

Max

1. /user/apply - call count: 120

2. /user/submit - call count: 90

3. /user/review - call count: 60

## Ranking

Call Count Ranking

Memory Usage Ranking

Select Date and Time: 연도-월-일 -- --:--

1. /user/apply - memory usage: 63.2MB

2. /user/submit - memory usage: 50.5MB

3. /user/review - memory usage: 40.1MB

# Thank you