

오픈마루 APM 운영 매뉴얼

온-나라 2.0 모니터링 시스템 (APM) 운영 매뉴얼

온-나라 2.0 모니터링 시스템 간단 사용법

On-Nara Monitoring System Simple User Guide

온-나라 2.0 모니터링 시스템 접속 방법

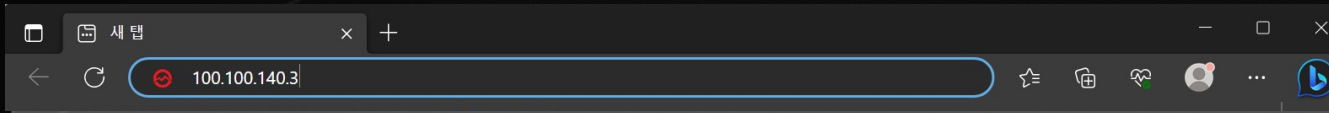
1



[다양한 브라우저 지원]

- 온-나라 2.0 모니터링 시스템, 관리자 PC에 설치되어 있는 웹 브라우저 프로그램을 실행합니다.

2



[온-나라 2.0 모니터링 시스템 주소 URL 입력]

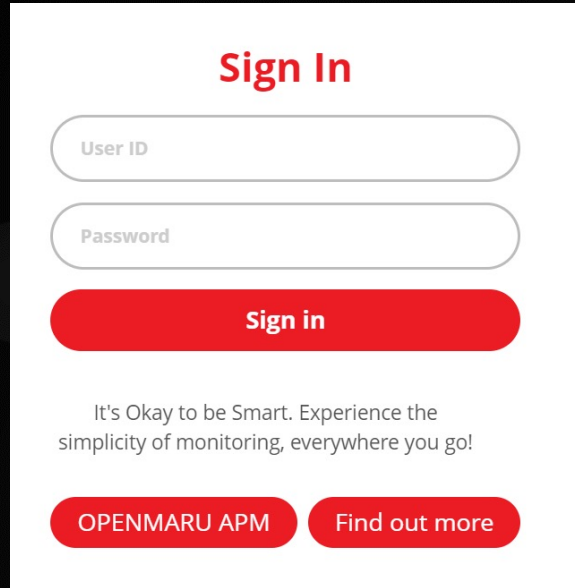
- 실행된 웹 브라우저 프로그램 주소 창에 온-나라 모니터링 시스템의 접속 IP를 입력합니다.

예) <https://100.100.140.3>

온-나라 2.0 모니터링 시스템 접속 IP를 모른다면 오픈마루로 연락주세요! (Tel. 070-4044-5426 | 운영 시간 09시 30분 ~ 18시)

온-나라 2.0 모니터링 시스템 접속 방법

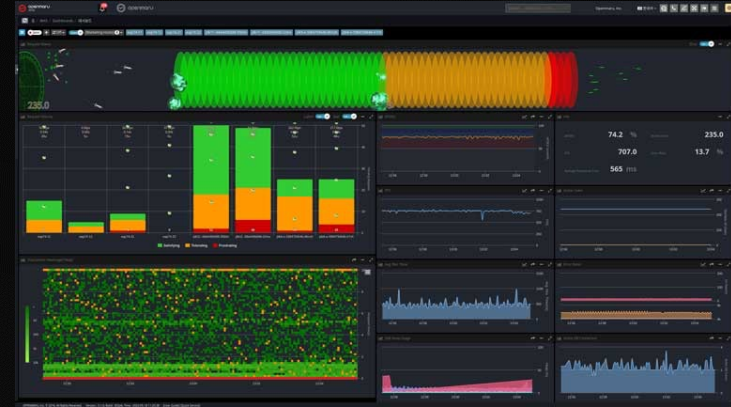
3



The image shows a 'Sign In' form with a white background and rounded corners. At the top, the text 'Sign In' is written in a bold, red font. Below this, there are two input fields: 'User ID' and 'Password', both with rounded rectangular borders. Underneath the input fields is a prominent red button with the text 'Sign in' in white. Below the button, there is a line of text: 'It's Okay to be Smart. Experience the simplicity of monitoring, everywhere you go!'. At the bottom of the form, there are two red buttons: 'OPENMARU APM' and 'Find out more', both with white text.

[APM 로그인 화면]

4



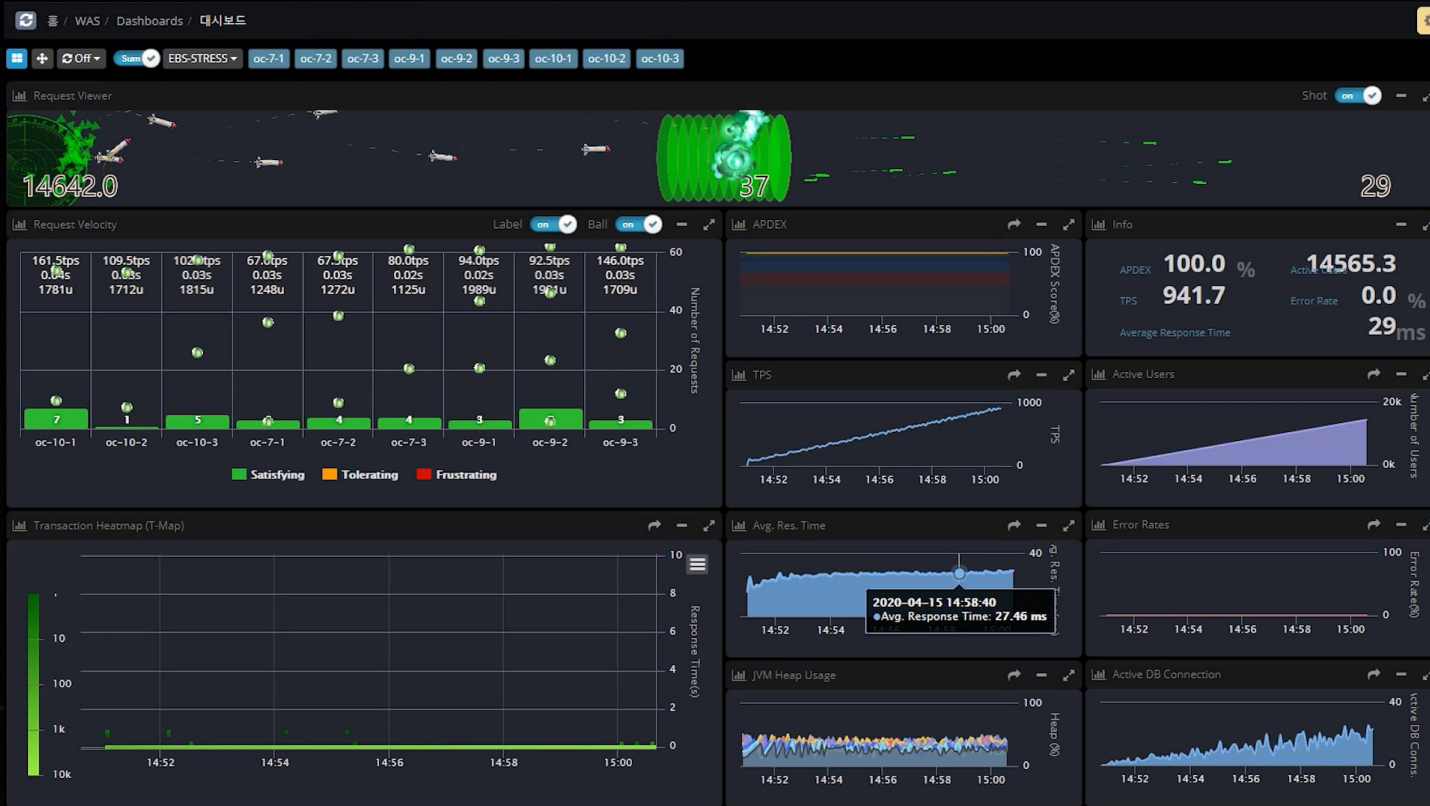
[APM 대시보드 화면]

- 온-나라 2.0 모니터링 관리자 로그인 화면이 표시되면 시스템의 관리자 계정 아이디와 비밀번호를 입력합니다.
(온-나라 2.0 모니터링 접속 IP 또는 아이디, 비밀번호를 모르신다면 하단에 있는 오픈마루 연락처로 문의주세요)

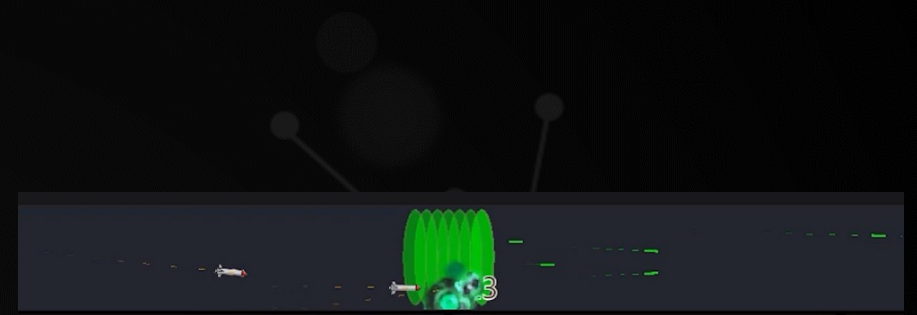
- 모니터링 대시보드가 표시되면 정상적으로 온-나라 2.0 모니터링 시스템에 접속된 것입니다.

오픈마루 문의처 (Tel. 070-4044-5426 | 운영 시간 09시 30분 ~ 18시)

온-나라 2.0 모니터링: 정상 예시



[APM 대시보드 화면]



[동시접속자 처리 현황]



[인스턴스 별 요청 처리 현황]

화면에 초록색으로 표시되는 상태는 온-나라 2.0 시스템의 서비스가 원활한 상태

온-나라 2.0 모니터링: 장애 상황 예시



[APM 대시보드 화면]



[동시접속자 처리 현황]



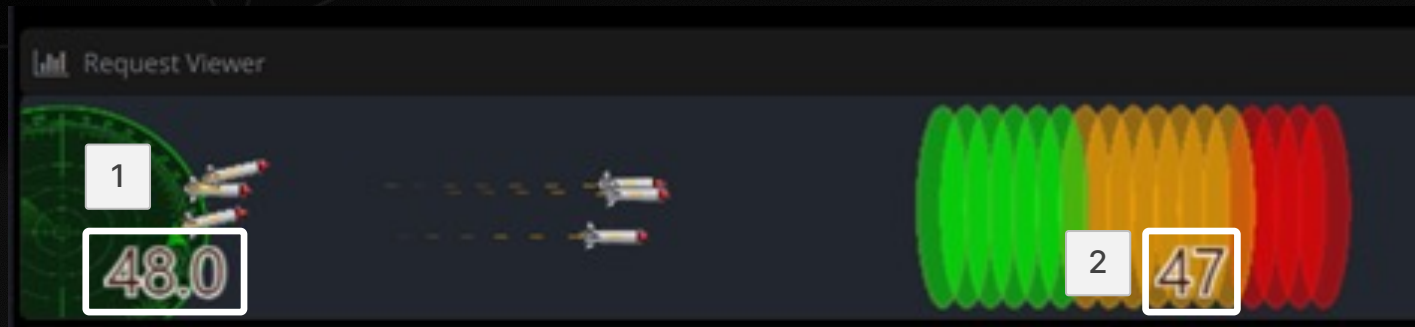
[인스턴스 별 요청 처리 현황]

붉은색으로 표시되는 상태는 온-나라 2.0 시스템의 서비스가 매우 느린 상태

대시보드 색상으로 서비스 상태 파악

구분	내용	
	초록	응답시간 3초 미만으로 서비스 양호 (안정적)
	주황	응답시간 3초 ~ 12초로 서비스 느림 (주의)
	빨강	응답시간 12초 이상으로 서비스 위험 (비정상)

온-나라 모니터링 시스템(OPENMARU APM)은 대시보드 색상으로 온-나라 2.0 서비스 상태를 직관적으로 구분이 가능합니다.



동시 접속자 수

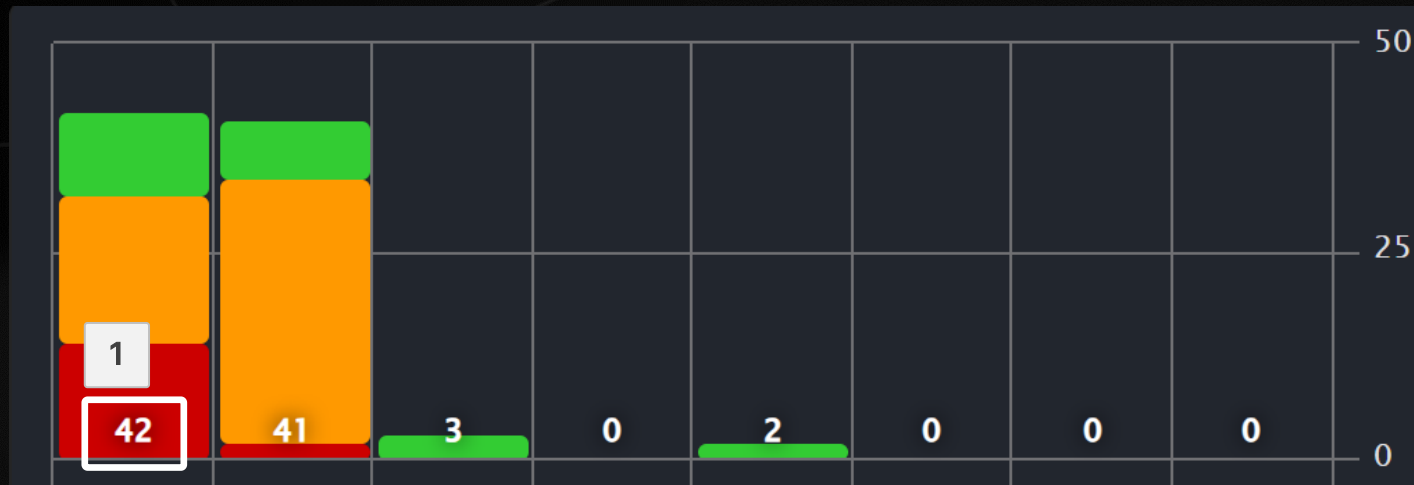
동시접속자 처리현황

- 1 온-나라 2.0 서비스 이용자 수
- 2 온-나라 2.0 서비스 처리중인 요청 수 (Request)

대시보드 색상으로 인스턴스 별 상태 파악

구분	내용	
	초록	응답시간 3초 미만으로 서비스 양호 (안정적)
	주황	응답시간 3초 ~ 12초로 서비스 느림 (주의)
	빨강	응답시간 12초 이상으로 서비스 위험 (비정상)

온-나라 모니터링 시스템(OPENMARU APM)은
대시보드 색상으로 온-나라 2.0 서비스의
인스턴스 별 상태를 직관적으로 구분할 수 있습니다.

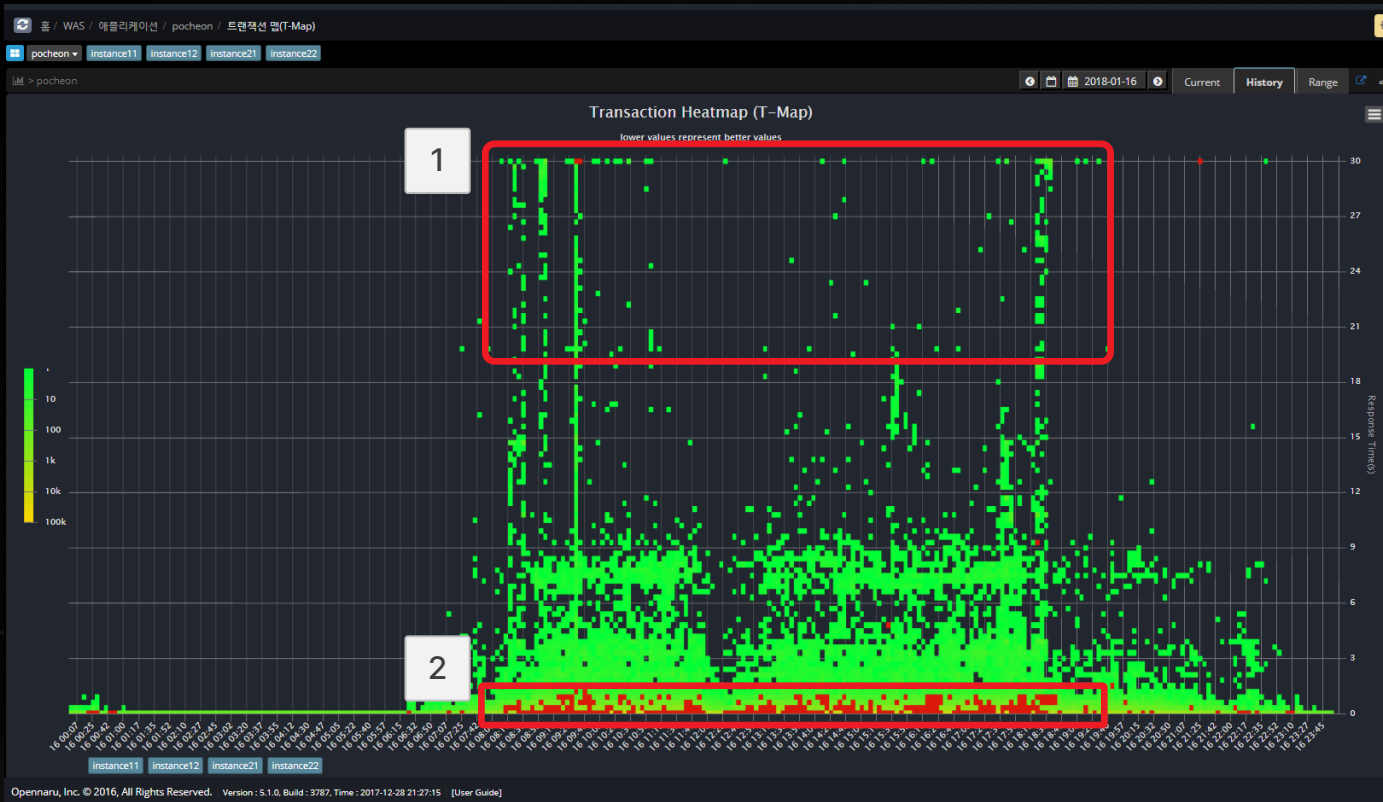


인스턴스별 동시 접속자 수

인스턴스별 요청 처리 현황

1 인스턴스 별 온-나라 2.0 서비스 이용자 수

온-나라 2.0 서비스 처리 현황을 표시하는 T-Map



시간 (Time)

응답시간 (Response Time)

온-나라 모니터링 시스템(OPENMARU APM)은 서비스 처리 현황을 표시하는 T-Map 대시보드를 제공하여 온-나라 2.0 서비스 상태 실시간 확인하는 기능 제공

- 1 서비스의 응답 시간이 30초 이상일 경우 30초 라인에 표시
- 2 HTTP 응답코드가 4xx, 5xx 에러가 발생할 경우 T-Map 하단에 붉은 점으로 표시

[APM 대시보드 - T-MAP 화면]

온-나라 2.0 서비스 처리 현황을 표시하는 T-Map



Agent Type	Server IP	Client IP	Instance ID	URL	Status	Duration(ms)	SQL Time(ms)	Fetch Gab	Fetch Count	External Time	CPU Time(ms)	Start Time
WAS	98.118.118.118	98.118.118.118	instance11	/bms/c...eport.do	200	21,280	8,099	3	186	0	380.16	2019-03-11 17:14:40.59
WAS	98.118.118.118	98.118.118.118	instance11	/blg/message/messe...n.do	200	16,445	16,444	0	2	0	1.27	2019-03-11 17:15:06.672
WAS	98.118.118.118	98.118.118.118	instance11	/blg/message/messe...n.do	200	17,167	17,165	0	2	0	1.50	2019-03-11 17:15:50.672
WAS	98.118.118.118	98.118.118.118	instance11	/blg/message/messe...m.do	200	16,321	16,320	0	2	0	1.63	2019-03-11 17:15:50.672
WAS	98.118.118.118	98.118.118.118	instance11	/blg/message/me...larm.do	200	16,334	16,333	0	2	0	1.95	2019-03-11 17:16:07.433
WAS	98.118.118.118	98.118.118.118	instance11	/blg/message/messageAlarm.do	200	16,293	16,292	0	2	0	1.57	2019-03-11 17:16:07.561
WAS	98.118.118.118	98.118.118.118	instance11	/blg/message/messageAlarm.do	200	15,885	15,884	0	2	0	1.66	2019-03-11 17:16:08.267

3

[DB 연결 Pool에서 Connection을 가져오는데 11초 소요되는 화면]

```

=====
[Num.][ Start Time | Elapsed | % | Excludi | A-Gab | CPUTime] Method Call
-----
[ 1][17:16:07.433] 16,334 | 100 | 0 | 0 | 2.0 | + org.springframework.web.servlet.FrameworkServlet.service()
[ 2][17:16:07.433] 16,334 | 100 | 0 | 16,334 | 1.9 | + org.springframework.web.servlet.DispatcherServlet.doService()
[ 3][17:16:07.433] 16,334 | 100 | 1 | 16,333 | 1.3 | + @RequestMapping onnara.blg.web.BlgMessageContoroller.messageAlara()
[ 4][17:16:07.433] 11,979 | 73 | 11,979 | 0 | 0.1 | + org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection()
[ 5][17:16:07.434] 11,979 | 73 | 11,979 | 0 | 0.1 | + org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection()
[ 6][17:16:19.413] 0 | 0 | 0 | 0 | 0.0 | + org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection()
[ 7][17:16:19.413] 0 | 0 | 0 | 0 | 0.1 | + org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection()
> DB : URL: jdbc:cubrid:98.12.100.120:34000:onnaracomon_db:::?:althost=98.12.100.120:34000;charset=utf-8
> Query : SELECT COUNT(*) FROM bms_com_message_blg WHERE menu_type = 'NCT' AND TO_CHAR(SYSDATETIME, 'yyyyMMdh24miss') BETWEEN SUBSTR(msg_title, -29
, 14) AND SUBSTR(msg_title, -14)
> Mapped: SELECT COUNT(*) FROM bms_com_message_blg WHERE menu_type = 'NCT' AND TO_CHAR(SYSDATETIME, 'yyyyMMdh24miss') BETWEEN SUBSTR(msg_title, -29
, 14) AND SUBSTR(msg_title, -14)
[ 8][17:16:19.413] 0 | 0 | 0 | 0 | 0.0 | + org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection()
[ 9][17:16:19.413] 0 | 0 | 0 | 0 | 0.0 | + cubrid.jdbc.driver.CUBRIDPreparedStatement.execute()
[ 10][17:16:19.413] 1 | 0 | 1 | 0 | 0.1 | + cubrid.jdbc.driver.CUBRIDResultSet.next()
[ 11][17:16:19.414] 0 | 0 | 0 | 0 | 0.0 | + cubrid.jdbc.driver.CUBRIDConnection.commit()
[ 12][17:16:19.414] 0 | 0 | 0 | 0 | 0.0 | + org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection()
[ 13][17:16:19.414] 2,687 | 16 | 2,687 | 0 | 0.1 | + org.jboss.as.connector.subsystems.datasources.WildFlyDataSource.getConnection()
[ 14][17:16:19.414] 2,687 | 16 | 2,687 | 0 | 0.1 | + org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection()
[ 15][17:16:22.101] 0 | 0 | 0 | 0 | 0.1 | + org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection()
[ 16][17:16:22.101] 1,665 | 10 | 1,665 | 0 | 0.2 | + org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection()
    
```

2

1

[APM 대시보드 - T-MAP]
드래그 & 드랍 화면

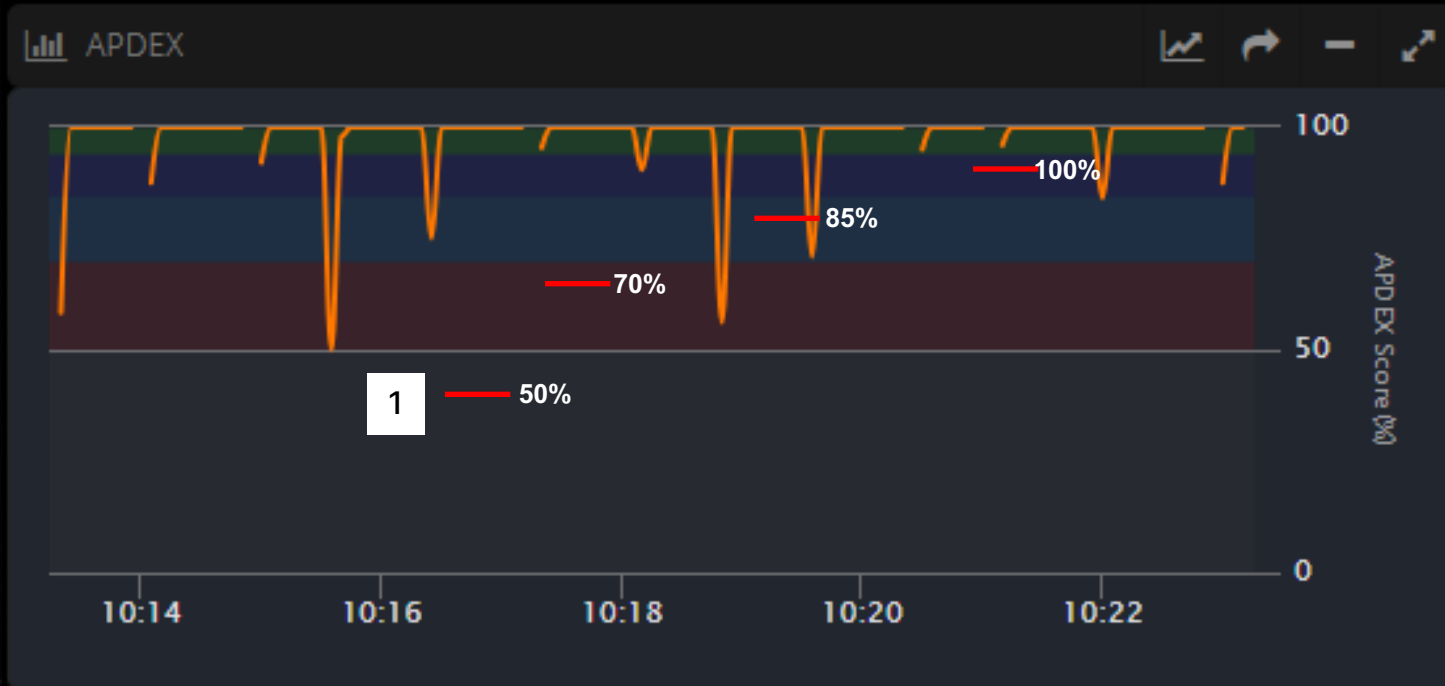
온-나라 모니터링 시스템(OPENMARU APM)은 서비스 처리 현황을 표시하는 T-Map 대시보드를 제공하여 온-나라 2.0 서비스 상태 실시간 확인 기능 제공

1 T-MAP을 드래그를 하여 Transaction Detail 확인 가능

2 HTTP 응답코드가 4xx, 5xx, 2xx 등 어떤 페이지에 접속 시 정상/비정상을 확인 가능

3 URL 접속 시간이 얼마나 걸렸는지 확인 가능

서비스 만족 지수(APDEX) 70점 이하는 서비스 문제 발생



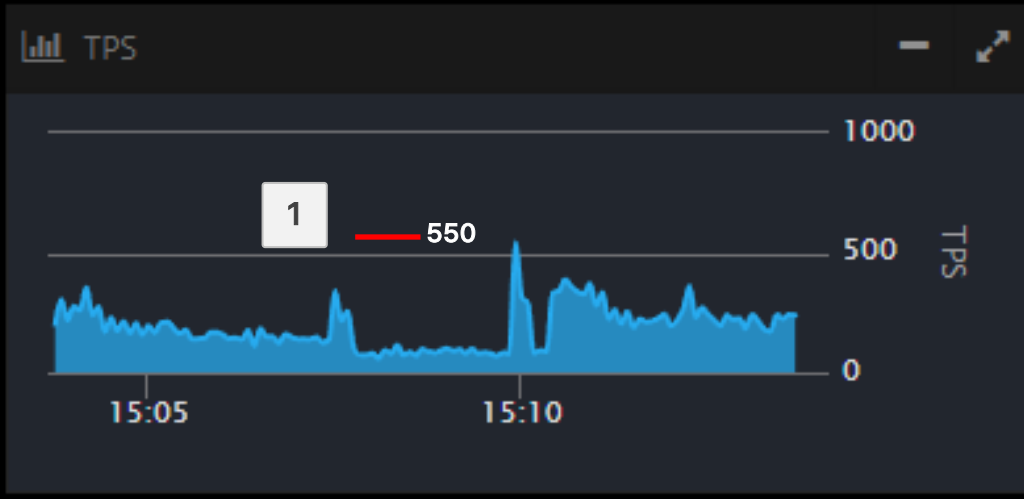
<APM 대시보드 - APDEX>

온-나라 모니터링 시스템(OPENMARU APM)은 서비스 만족 지수 (APDEX) 대시보드를 제공하여 온-나라 2.0 서비스 상태 실시간 확인 기능 제공

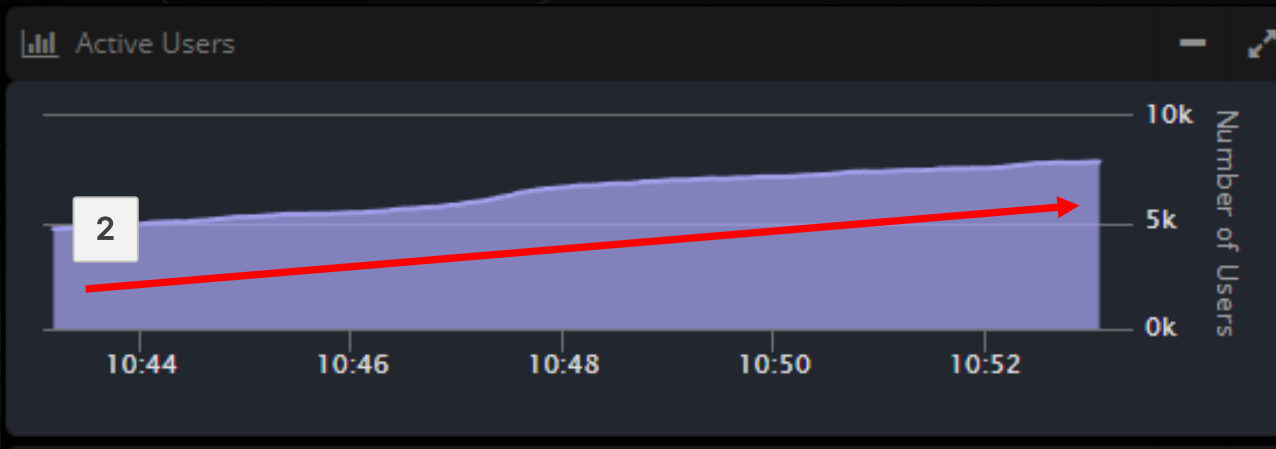
- 1 서비스 만족 지수 (APDEX)가 70점 이하라면 온나라 2.0 서비스에 문제 발생

등급	점수	상태
Excellent	94 ~ 100	매우 좋음
Good	85 ~ 94	좋음
Fair	70 ~ 85	보통
Poor	50 ~ 70	나쁨
Unacceptable	0 ~ 50	아주 나쁨

TPS & 액티브 유저의 값의 의미



<APM 대시보드 - TPS>



<APM 대시보드 - Active Users>

온-나라 모니터링 시스템(OPENMARU APM)은 대시보드에 TPS (Transaction per Second)와 Active Users (사용자 수)를 제공

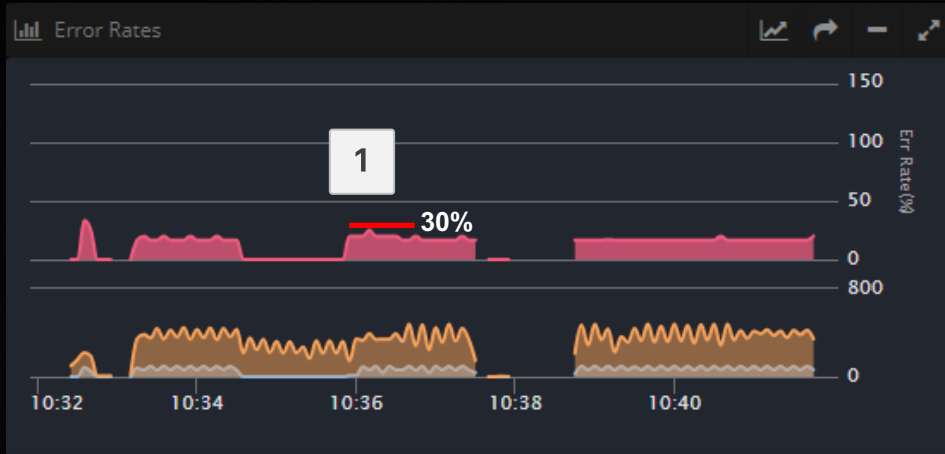
1 TPS(Transaction per Second)는 초당 처리 건수

구분	설명
TPS 높은 값	처리량이 많음
TPS 낮은 값	처리량이 적음

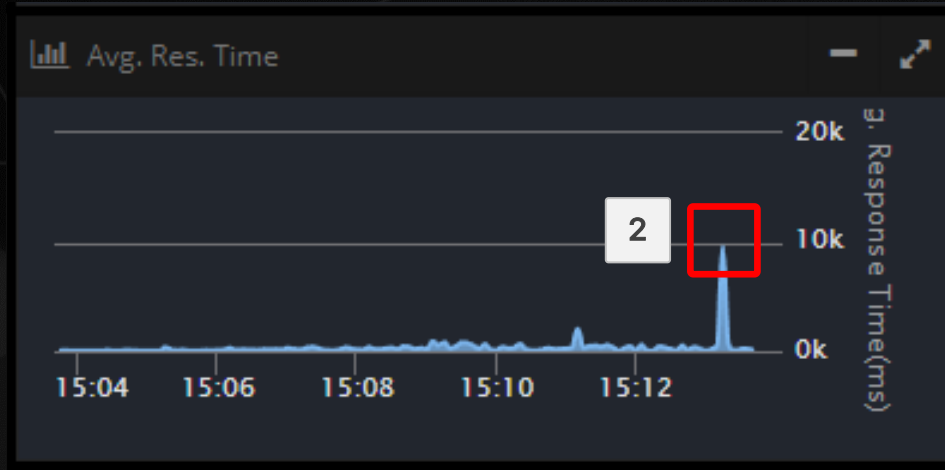
2 Active Users (액티브 유저)는 온-나라 2.0 동시 접속 사용자 수

구분	설명
Active Users 높은 값	사용자가 많음
Active Users 낮은 값	사용자가 적음

오류율, 평균 응답시간의 의미



<APM 대시보드 - 오류율>



<APM 대시보드 - 평균응답시간>

온-나라 모니터링 시스템(OPENMARU APM)은 대시보드에 Error Rates(오류율)와 Avg. Res. Time (평균 응답시간)을 제공

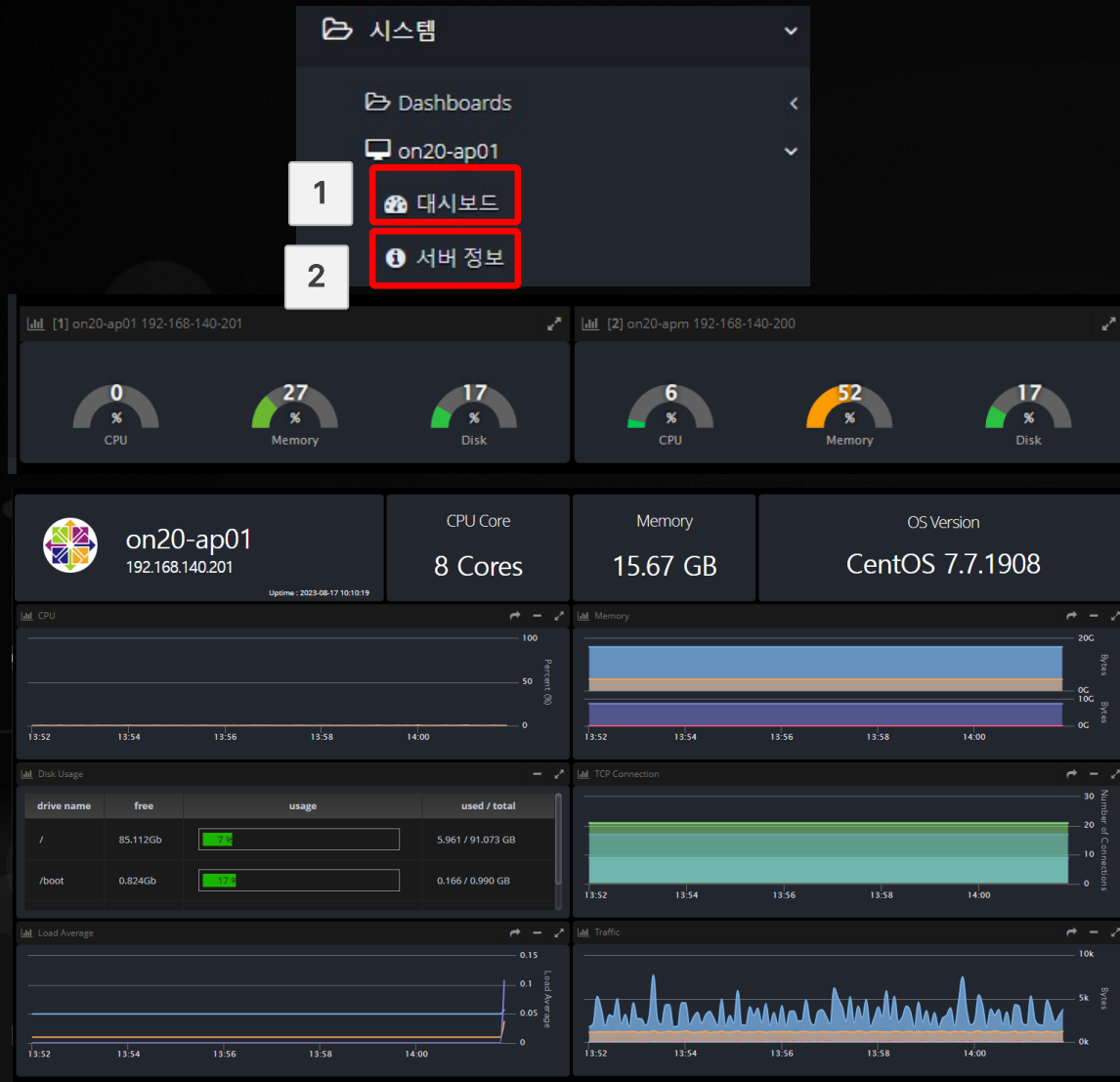
1 Error Rates(오류율)는 오류의 수치 표시

구분	설명
Error Rates 높은 값	문제가 많음
Error Rates 낮은 값	문제가 적음

2 Avg. Res. Time (평균 응답시간)은 온-나라 2.0 서비스 속도

구분	설명
Avg Res Time 높은 값	서비스가 느림
Avg Res Time 낮은 값	서비스가 빠름

시스템 사용량 확인



<APM 시스템 대시보드>

온-나라 모니터링 시스템(OPENMARU APM)은 SYSTEM의 리소스(사용량)을 한 눈에 모니터링 가능

1 대시보드 탭으로 온-나라 시스템 전체의 시스템 사용량 확인 가능

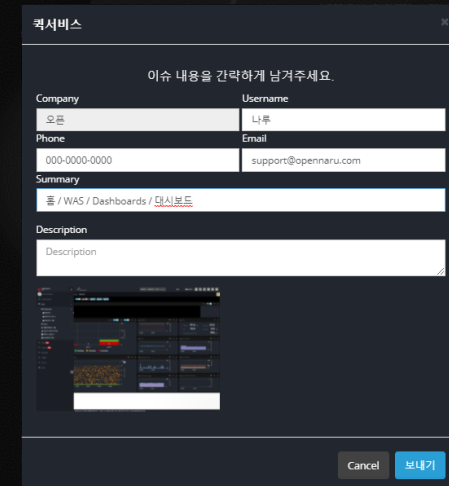
2 서버정보 탭으로 각 서버의 상세 정보를 확인 가능

전문가 기술지원 요청 방법 (Quick Service)



퀵서비스(Quick Service)는 시스템 장애 발생 시 해당 화면과 시스템 정보를 오픈마루의 전문가에게 e-mail 등으로 전달하여 원인이 무엇인지 분석해 주는 실시간 온라인 서비스

실시간으로 문의된 퀵서비스 내용은 신속하게 담당자가 할당되고 처리 과정을 포탈, 메일 등을 통해서 보고서 형태로 공유



퀵서비스

이슈 내용을 간략하게 남겨주세요.

Company	Username
오픈마루	나루
Phone	Email
000-0000-0000	support@openmaru.com

Summary

동 / WAS / Dashboards / 대시보드

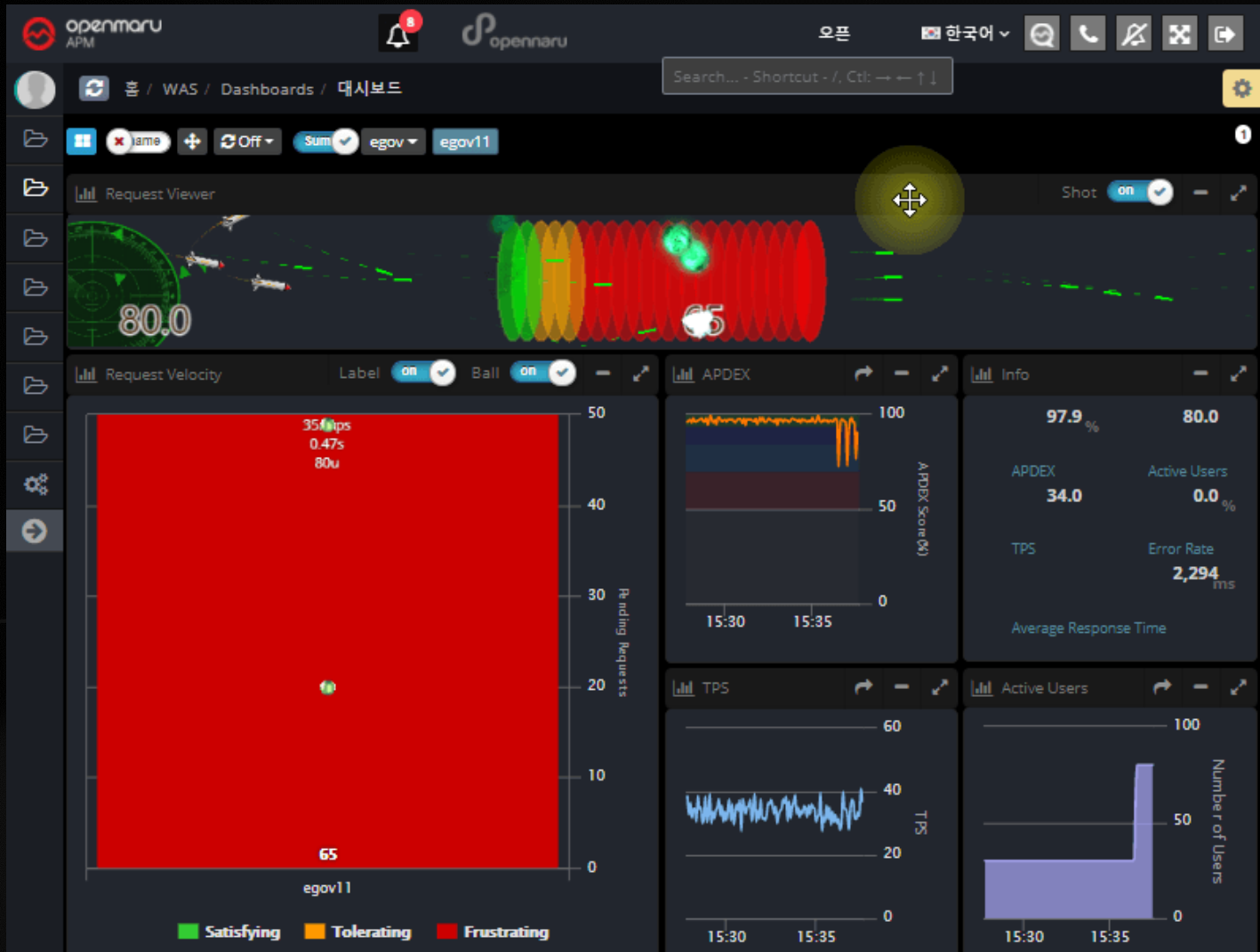
Description

Description

Cancel 보내기

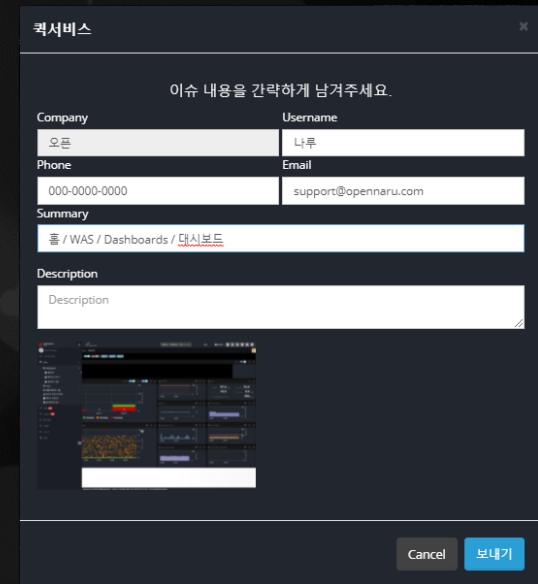
전문가 기술지원 요청 방법 (Quick Service)

- 대시보드에서 Quick Service 이용 방법



온-나라 2.0 모니터링 시스템 상단에 있는 전화기 모양의 Icon을 클릭 후 표시되는 팝업창에 기본 정보를 작성하여 보내기 버튼 클릭

- Company : 회사명
- Username : 계정 정보
- Phone : 연락처
- Email : 이메일 주소
- Description : 증상

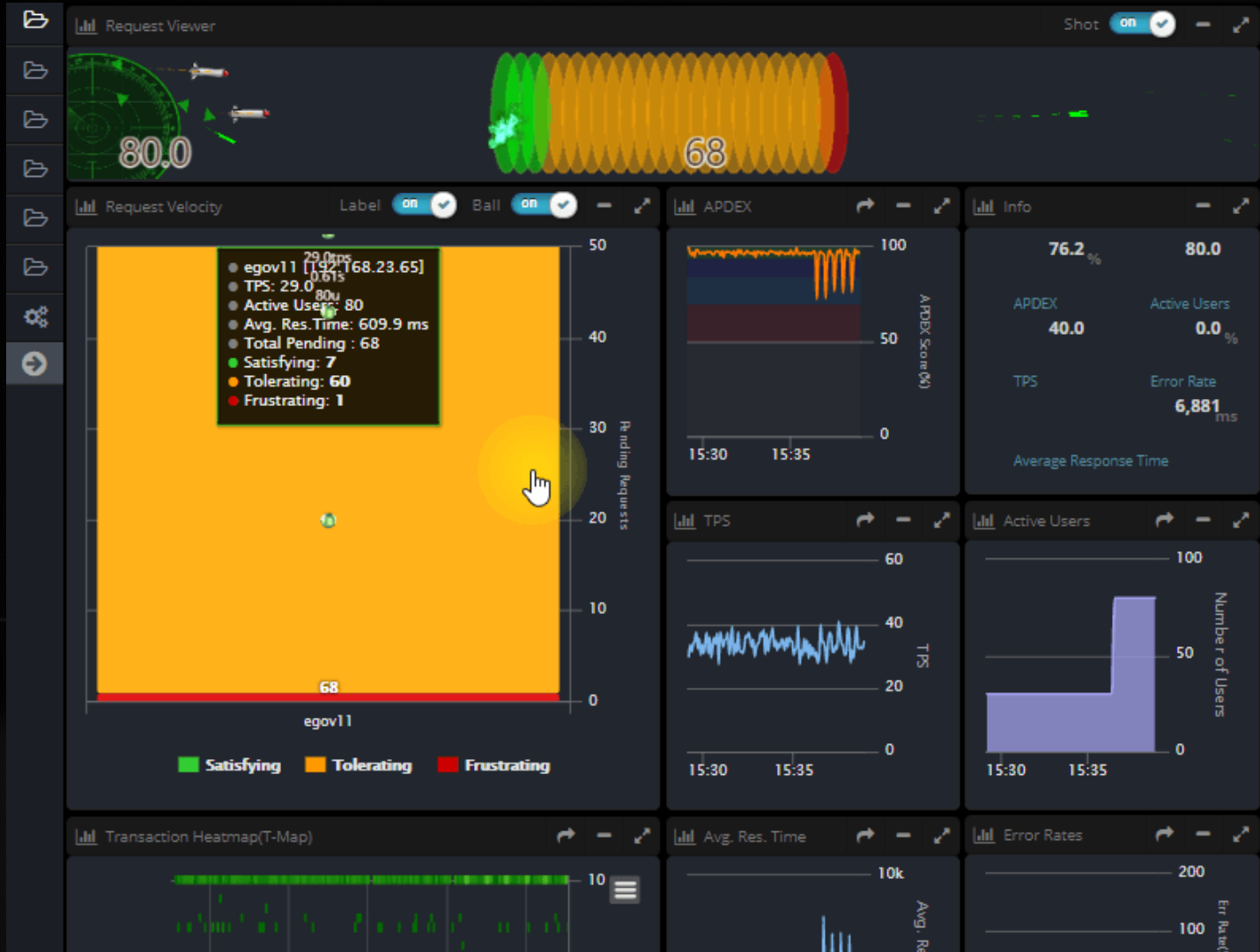


The screenshot shows a '퀵서비스' (Quick Service) popup form. At the top, there's a title bar with the text '퀵서비스' and a close button. Below the title bar, there's a message: '이슈 내용을 간략하게 남겨주세요.' (Please briefly describe the issue content). The form contains several input fields: 'Company' (with a dropdown menu showing '오픈'), 'Username' (with a dropdown menu showing '나루'), 'Phone' (with a dropdown menu showing '000-0000-0000'), and 'Email' (with a dropdown menu showing 'support@openmaru.com'). There's also a 'Summary' section with a text area containing '홈 / WAS / Dashboards / 대시보드'. Below that, there's a 'Description' section with a larger text area. At the bottom right, there are two buttons: 'Cancel' and '보내기' (Send).

<퀵 서비스 보내는 방법>

전문가 기술지원 요청 방법 (Quick Service)

- Quick Service를 이용한 스레드 덤프(Thread Dump) 방법







온-나라 2.0 모니터링 대시보드에 있는 스레드 덤프 목록을 선택하고 분석할 스레드 덤프를 선택 후 퀵서비스 버튼을 클릭, 표시되는 팝업창에 기본 정보를 입력 후 보내기 버튼 클릭

- Company : 회사명
- Username : 계정 정보
- Phone : 연락처
- Email : 이메일 주소
- Description : 증상

이벤트 알람 유형

- 빨간색 알람 - 문제가 있는 상황

구분	레벨	설명
 <p>[정보-INFO] 사용자가 요청한 '전체 스레드 덤프'가 저장되었습니다. 📧 발생에이전트 : admin11@192.168.23.111 클릭하여 상세한 정보를 확인하세요.... ⌚ 5s.</p>	정보	사용자가 요청한 명령이 실행될 때의 알람
 <p>[경고-WARN] 'APDEX' (평균값: 65.97)이 경고 임계값 '70'을 넘었습니다. 📧 발생에이전트 : admin11@192.168.23.111 클릭하여 상세한 정보를 확인하세요.... ⌚ 8s.</p>	주의	경고 정책 Warn을 넘어설 경우 '현재 평균 값 xx가 경고 임계치 xx를 넘었다'는 메시지와 발생한 에이전트의 위치를 표시
 <p>[심각-CRITICAL] 'APDEX' (평균값: 6.25)이 CRITICAL 임계값 '50'을 넘었습니다. 📧 발생에이전트 : test21@192.168.23.112 클릭하여 상세한 정보를 확인하세요.... ⌚ 10s.</p>	경고	Warn 임계치를 초과하여 Critical로 설정된 임계치 값 범위에 도달한 경우 '현재 평균값 xx가 경고 임계값 xx를 넘었다.'는 메시지와 발생한 에이전트의 위치를 표시
 <p>[예측-FORECAST] 'APDEX' (현재값:25)이 '2016-02-12 22:34:38'쯤에 예상값 '8.95'으로 CRITICAL 임계값 '50'에 도달할 것으로 예상됩니다. 📧 발생에이전트 : admin21@192.168.23.112 클릭하여 상세한 정보를 확인하세요.... ⌚ 5s.</p>	예측	통계를 기반으로 임계 값에 곧 도달할 것을 미리 예측하여 알람을 표시

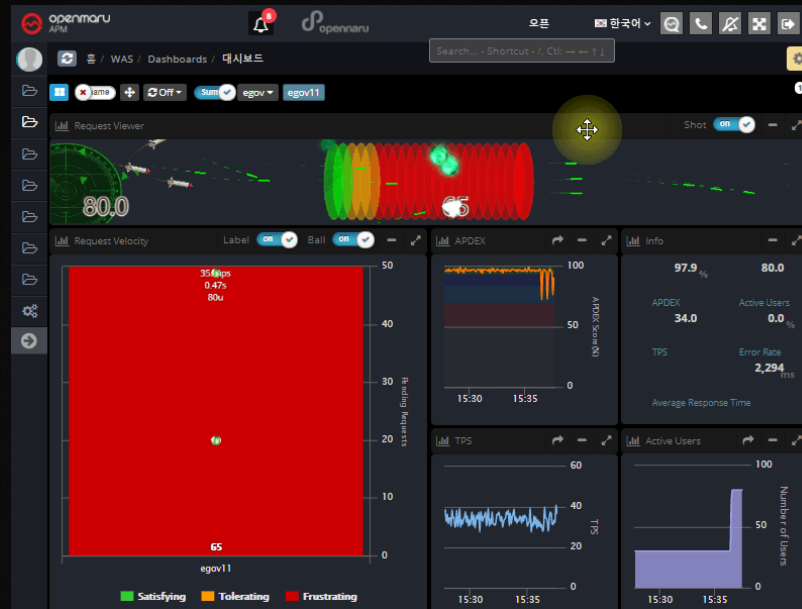
온-나라 2.0 모니터링 APM에는 4가지 유형의 알람 (정보, 주위, 경고, 예측)이 이벤트(Event)발생 시 표시 색상으로 구분되어 직관적으로 이벤트에 대한 정보 확인

다른 것은 몰라도 **3가지**는 꼭 기억하세요!

Agent Type	Server IP	Client IP	Instance ID	URL	Status	Duration(ms)	SQL Time(ms)	Fetch Gab	Fetch Count	External Time	CPU Time(ms)	Start Time
WAS	98.12.100.118	98.12.23.91	instance11	/oms/oc/edreport00	200	21,290	8,099	3	100	0	88.16	2019-03-11 17:16:40.597
WAS	98.12.100.118	98.12.12.88	instance11	/big/message/messageAlarm.do	200	16,445	16,444	0	2	0	1.27	2019-03-11 17:16:07.433
WAS	98.12.100.118	98.12.131.87	instance11	/big/message/messageAlarm.do	200	17,167	17,165	0	2	0	1.50	2019-03-11 17:16:07.433
WAS	98.12.100.118	98.12.48.86	instance11	/big/message/messageAlarm.do	200	16,321	16,320	0	2	0	1.63	2019-03-11 17:16:07.433
WAS	98.12.100.118	98.12.138.88	instance11	/big/message/messageAlarm.do	200	16,334	16,333	0	2	0	1.95	2019-03-11 17:16:07.433
WAS	98.12.100.118	98.12.15.41	instance11	/big/message/messageAlarm.do	200	16,293	16,292	0	2	0	1.57	2019-03-11 17:16:07.561
WAS	98.12.100.118	98.12.48.63	instance11	/big/message/messageAlarm.do	200	15,885	15,884	0	2	0	1.66	2019-03-11 17:16:08.267

```

[Num.] [Start Time] [Elapsed] [%] [Exclns] [A-Gab] [CPUTime] [Method Call]
-----
[ 1][17:16:07.433][ 36,334][ 100][ 0][ 0][ 0.0] + org.springframework.web.servlet.FrameworkServlet.service()
[ 2][17:16:07.433][ 36,334][ 100][ 0][ 16,334][ 1.9] + org.springframework.web.servlet.DispatcherServlet.doService()
[ 3][17:16:07.433][ 36,334][ 100][ 0][ 16,333][ 2.3] + @RequestMappingHandlerMethod@org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerMethod.invokeHandlerMethod()
[ 4][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[ 5][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[ 6][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[ 7][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[ 8][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[ 9][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[10][17:16:07.433][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[11][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[12][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[13][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[14][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[15][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[16][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[17][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[18][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[19][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[20][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[21][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[22][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[23][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[24][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[25][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[26][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[27][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[28][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[29][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[30][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[31][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[32][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[33][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[34][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[35][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[36][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[37][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[38][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[39][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[40][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[41][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[42][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[43][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[44][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[45][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[46][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[47][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[48][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[49][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[50][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[51][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[52][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[53][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[54][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[55][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[56][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[57][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[58][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[59][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[60][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[61][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[62][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[63][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[64][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[65][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[66][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[67][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[68][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[69][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[70][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[71][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[72][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[73][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[74][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[75][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[76][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[77][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[78][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[79][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[80][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[81][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[82][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[83][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[84][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[85][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[86][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[87][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[88][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[89][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[90][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[91][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[92][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[93][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[94][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[95][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[96][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[97][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[98][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[99][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
[100][17:16:07.434][ 36,334][ 100][ 0][ 0.1] + org.springframework.jdbc.datasource.DataSourceUtils.getConnection()
  
```



[T-MAP 드래그 & 드랍 화면]

[퀵서비스 요청 방법]

[APM 시스템 대시보드 리소스 확인]

사용 중 궁금하신 사항은 오픈마루로 연락주세요!

[업무 시간] 9시 30분 ~ 6시 | [점심 시간] 11시 30분 ~ 12시 30분

Tel. 070-4044-5426

E-Mail. support@openmaru.io