

데브와이 대기열 서비스 메가펜스 기술문서 (웹사이트 운영자용)

2021.6

DEV-Y



MEGA-FENCE 동작 개요-I (REPLACE 방식)

홈페이지 방문자

응답페이지

대기표 발급 서버

홈페이지

(DEVY-IDC)

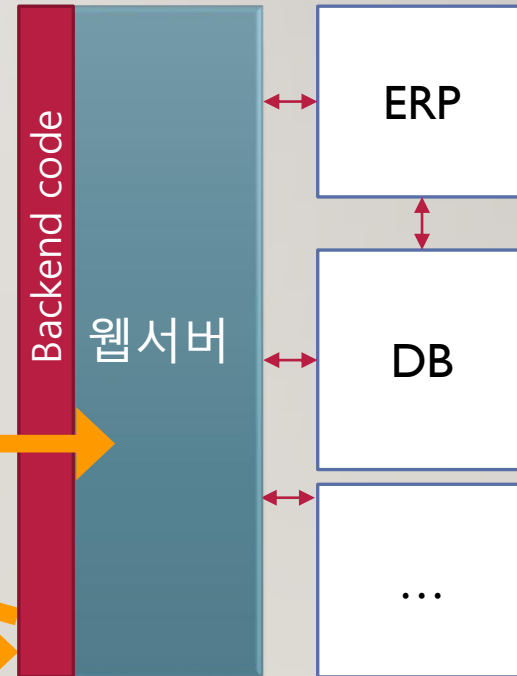
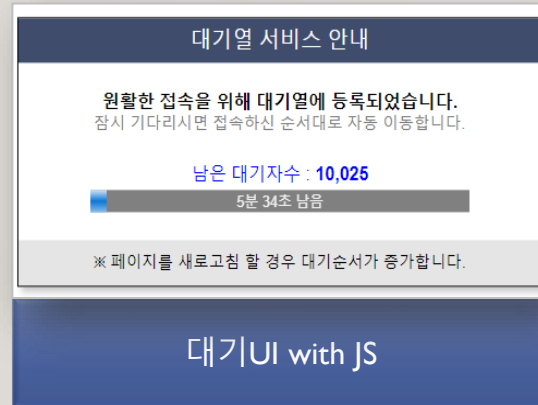
(ON-PREMISE)

- Email 내의 Link
- SMS 내의 Link
- 검색포탈 검색
- 직접 URL 입력

...



- HTTP (WEB-API, JSON)



대기UI response

- 페이지에 삽입된 Backend code가 대기열 UI응답
- 대기가 완료되면 업무페이지가 로드됨
- 트래픽이 적은 경우에는 대기UI 응답없이 업무페이지 응답 (BYPASS)

MEGA-FENCE 동작 개요-2 (EMBEDDED 방식)

홈페이지 방문자

응답페이지

대기표 발급 서버

(DEVY-IDC)

홈페이지

(ON-PREMISE)

- Email 내의 Link
- SMS 내의 Link
- 검색포탈 검색
- 직접 URL 입력

...



Static page(HTML)
with module(JS)

+

callback script (JS)

대기열 서비스 안내

원활한 접속을 위해 대기열에 등록되었습니다.
잠시 기다리시면 접속하신 순서대로 자동 이동합니다.

남은 대기자수 : 10,025

5분 34초 남은

※ 페이지를 새로고침 할 경우 대기순서가 증가합니다.

서버#1

서버#2

...

▪ HTTP (WEB-API, JSON)

웹서버

ERP

DB

...

Static page response

- 버튼 클릭 등의 event callback부에 대기열 호출하도록 수정
- 대기완료 callback부에는 기존의 업무처리 구현부(ajax, post 등...)를 이동하여 대기가 끝나면 실행되도록 함



사전 체크

❖ 고객사 웹서버(WAS)의 HTTP OUT-BOUND 방화벽 해제 (Replace 방식 적용 시)

- 일반적으로 HTTP OUT-BOUND 방화벽은 해제되어 운영되지만
만일 강제로 설정된 SITE라면 있다면 아래 IP에 대해서 개별 해제 필요

SERVER	IP	Direction	PORT	비고
wc2.devy.kr	116.125.140.110	Out-bind	80	대기열 처리서버
wk3.devy.kr	58.230.162.210			
wk2.devy.kr	58.230.162.217			
ws1.devy.kr	49.247.3.178			
wc1.devy.kr	116.125.140.74			

참고

❖ 제공된 샘플 중 환경에 적합한 1가지 방식만 적용하세요

- JAVA+JSP 환경인 경우 JAVA방식을 권장하지만 JSP도 무방

❖ 제공된 샘플소스는 아래의 환경에서 정상 동작을 확인하였습니다.

개발 환경이 일치하지 않아도 소스 적용 후 대기UI가 정상 표시된다면 적용 가능합니다.

- IDE : Eclipse 2020-12
- Framework : SpringBoot 2.4.2 - Maven
- JAVA : JDK 1.8



적용 준비

❖ 제공된 샘플 소스 확인 (이메일로 안내)

- 샘플 소스를 참고하여 대기열 호출코드를 삽입해 주세요
(페이지의 진입부에 삽입해야 부하방지 효과가 발생합니다)

❖ Admin 페이지에서 할당된 SERVICE ID 및 GATE ID 확인 (샘플동작 확인 후 제공됩니다)

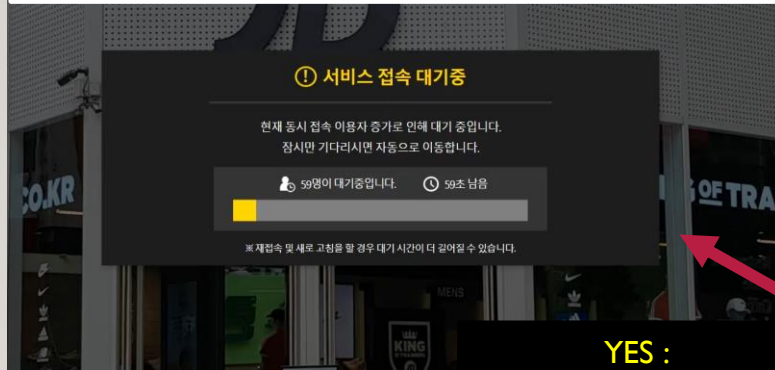
- 대기열 상태 확인 및 통과수량을 조절할 수 있는 Admin URL이 제공됩니다.
이용 가능한 대기열 식별자(GATE ID)를 확인하세요.
- 예) <https://cust.devy.kr/dashboard/?ServiceId=XXXX&LoginKey=YYYY>

GATE	현황 (10초평균)			초당 통과수 설정 (초과 시 대기자처리)
	상태	총대기자수/최대대기시간	초당 실통과자수/설정값	
POC-1 GATE-ID : 153	준비	0 00:00:00	0 / 5	<div><div>-</div><div>5</div><div>+</div></div> <div>-100 -10 +10 +100</div>
POC-2 GATE-ID : 154	준비	0 00:00:00	0 / 5	<div><div>-</div><div>5</div><div>+</div></div> <div>-100 -10 +10 +100</div>
POC-3 GATE-ID : 155	준비	0 00:00:00	0 / 5	<div><div>-</div><div>5</div><div>+</div></div> <div>-100 -10 +10 +100</div>
DEV-TEST GATE-ID : 156	혼잡	189 00:00:37	5 / 5	<div><div>-</div><div>5</div><div>+</div></div> <div>-100 -10 +10 +100</div>

대기열 동작 개요 (REPLACE 방식)

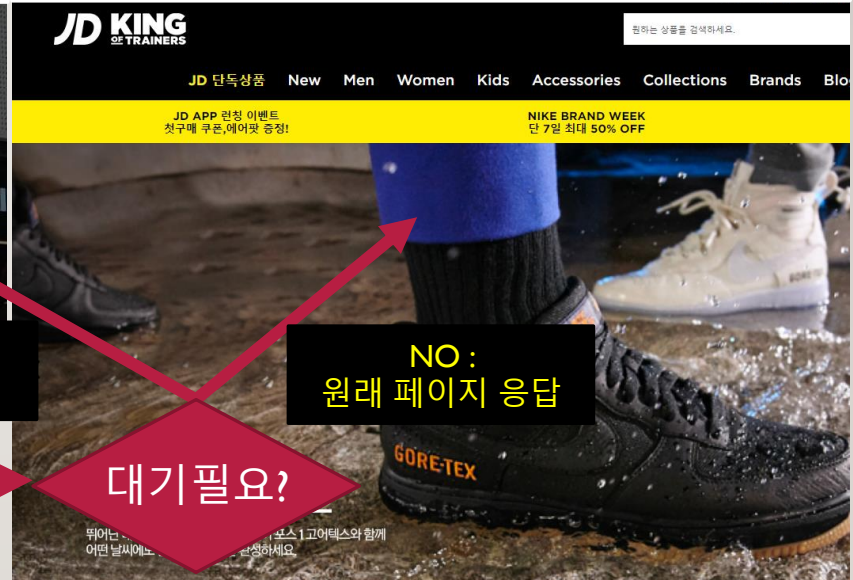
- 운영 중인 웹 페이지에 대기열 서비스 호출 코드를 삽입하여 페이지 로드 시 동작

대기 UI(HTML)



YES :
대기 UI 응답

운영 페이지 (backend code injection)



NO :
원래 페이지 응답

대기필요?

대상페이지 진입시도
(URL 이동)

대기열 적용예시 (JAVA / SPRINGBOOT)

1. 라이브러리 파일(WebGate.java) 파일을 프로젝트(패키지)에 추가①
2. 대기열을 적용할 업무페이지 컨트롤러②의 상단에 대기열서비스 코드 삽입③
 - SERVICE-ID : 계약된 SERVICE ID (메일로 안내됨)
 - GATE-ID : 할당받은 ID 중에서 하나를 선택하여 사용 (메일로 안내됨)

The screenshot shows an IDE with a project named 'megafence'. The project structure on the left includes 'megafence-sample-java [boot] [devtools]' with subfolders 'src/main/java' and 'src/main/resources'. Under 'src/main/java', there is a package 'com.devy.demo' containing 'Application.java' and 'BackendSampleController.java' (highlighted with a red box and a red circle with the number 2). Another package 'com.devy.megafence' contains 'package-info.java' and 'WebGate.java' (highlighted with a red box and a red circle with the number 1). The main code editor shows the 'BackendSampleController' class with a method 'index' that calls 'WebGate' service. A blue box highlights the code for inserting the queue logic, marked with a red circle with the number 3. The code includes comments in Korean and English, and uses 'WebGate' to check for a waiting status and return a UI HTML response.

```
36 @Controller
37 public class BackendSampleController {
38
39     @GetMapping("/backend-sample")
40     public String index(HttpServletRequest request, HttpServletResponse response) {
41
42         /*
43         Light business logic ...
44         예) 로그인 체크 : 쿠키나 세션을 체크해서 Login 페이지로 redirect
45         */
46
47         /* BEGIN OF 유량제어 코드삽입
48         ***** */
49         String serviceId = "9000"; // 할당받은 SERVICE ID (fixed)
50         String gateId = "1"; // 사용할 GATE ID (할당받은 GATE ID 범위내에서 사용)
51         WebGate webgate = new WebGate(serviceId, gateId, request, response);
52         if(webgate.WG_IsNeedToWaiting())
53         {
54             // 대기가 필요하면 대기UI로 응답 대체 후 종료
55             String uiHtml = webgate.WG_GetWaitingUi();
56             try {
57                 response.setContentType("text/html");
58                 PrintWriter out = response.getWriter();
59                 out.write(uiHtml);
60                 out.close();
61             } catch (Exception e) {}
62             finally {}
63         }
64         /* *****
65         END OF 유량제어 코드삽입 */
66
67         /*
68         Heavy business logic ...
69         예) 고객등급 GET : 고객 DB에서 고객등급(Bronze/Silver/Gold) 조회
70         예) 주문상태 GET : 주문 DB에서 주문상태(주문완료/배송중/배송완료)별 수량 조회
71         */
72     }
73 }
74
```


대기열 적용예시 (JSP)

1. 라이브러리 파일(webgate-lib.jsp) 파일 추가①
2. 대기열을 적용할 업무페이지②의 상단에 대기열 서비스 코드 삽입③
 - SERVICE-ID : 계약된 SERVICE ID (메일로 안내됨)
 - GATE-ID : 할당받은 ID 중에서 하나를 선택하여 사용 (메일로 안내됨)

The screenshot displays an IDE with two panels. The left panel shows a project structure for 'megafence'. Under the 'src' directory, the 'resources' folder contains 'static' and 'templates'. The 'webapp' directory contains 'WEB-INF', which in turn contains a 'views' folder. The 'views' folder contains two files: 'backend-sample.jsp' and 'webgate-lib.jsp'. A red box highlights these two files, with a red circle containing the number '1' next to 'webgate-lib.jsp' and a red circle containing the number '2' next to 'backend-sample.jsp'.

The right panel shows the content of the 'webgate-lib.jsp' file. The code is as follows:

```
22 *   ○ JAVA Framework을 사용하는 경우 반드시 JAVA OR JSP 중에 하나에만 적용합니다. (중복적용 금지!)
23 *   =====
24
25 <!--
26 Light business logic ...
27 예) 로그인 체크 : 쿠키나 세션을 체크해서 Login 페이지로 redirect
28 -->
29
30
31 <!--
32 BEGIN OF 유량제어 코드삽입
33
34 <%@ include file="webgate-lib.jsp" %>
35 <%
36
37 // Setting
38 String serviceId    = "9000"; // 할당받은 SERVICE ID (fixed)
39 String gateId       = "1";    // 사용할 GATE ID (할당받은 GATE ID 중에서 사용)
40
41 // 유량제어 체크 : 접속자가 많으면 대기UI로 응답 대체
42 if(WG_IsNeedToWaiting(serviceId, gateId, request, response))
43 {
44     out.print(WG_GetWaitingUi(serviceId, gateId));
45     return; // 응답종료
46 }
47 <%--
48
49 END OF 유량제어 코드삽입 -->
50
51
52 <!--
53 Heavy business logic ...
54 예) 고객등급 GET : 고객 DB에서 고객등급(Bronze/Silver/Gold) 조회
55 예) 주문상태 GET : 주문 DB에서 주문상태(주문완료/배송중/배송완료)별 수량 조회
56 -->
57
```

A red circle containing the number '3' is positioned next to the code block starting at line 32 and ending at line 49.

참고사항

<UI-TEST 모드 >

- 대기열 UI 동작 확인이 필요한 경우 Admin Page의 GATE 설정에서 UI-TEST 체크를 하면 실제 트래픽과 무관하게 해당 페이지 접속 시 대기UI가 30초간 표시됩니다.

대문페이지 GATE-ID : 6	준비 ?	0 ? 00:00:00 ?	0 ?	UI-TEST 모드 동작 중!!! 30초 동안 대기UI가 표시됩니다. 대기UI 확인용(운영GATE 사용주의!) <input checked="" type="checkbox"/> UI-TEST
상품상세 페이지 GATE-ID : 292	준비 ?	0 ? 00:00:00 ?	0 ?	<input type="checkbox"/> UI-TEST
주문페이지 GATE-ID : 293	준비 ?	0 ? 00:00:00 ?	0 ?	<input type="checkbox"/> UI-TEST

- 제공된 샘플(SERVICE ID : 9000)의 유량제어 설정은 대기UI가 표시되도록 설정되어 있습니다.
만일 대기UI가 표시되지 않으면 서비스 적용에 문제가 있다고 볼 수 있으므로 기술 지원을 요청 바랍니다.

THANK YOU

감사합니다!

기술지원/문의
양성덕 대표
010-7709-9036
contact@devy.kr

