셀 데이터 그래프에서 데이터 수가 많을 경우(데이터 5만개 이상) 마우스 커서가 느리게 따라오는 delay 문제와 비슷한 사례 (실시간 배달운영시스템 - 배민라이더스 BROS 1.0)

<http://woowabros.github.io/woowabros/2017/09/12/realtime-service.html>

**Browser의 성능**

실시간으로 화면이 렌더링 되는 것은 사실 아무런 문제가 없었습니다. 하지만 피크시간인 저녁시간에 배달 건수가 급격히 늘어나기 시작하면서 배차를 위해 라이더리스트 다이얼로그를 클릭한다던지 했을 때 0.5초 정도 움찔하는 delay가 발생하였습니다. 사용자 입장에서는 꽤나 신경이 거슬리는 이슈였습니다.

일반적인 초기 화면 진입 이후 뷰의 렌더링이 거의 없는 정적인 페이지와 달리 BROS의 배달/라이더 현황 페이지들 javascript가 실시간으로 이벤트를 수신받고 모델에 반영하고 뷰를 렌더링하고…… 쉴 새 없이 일을 하고 있었습니다. 그래서 성능최적화 작업에 들어갔습니다.

**모든 loop를 native for로 변경**

**forEach, angular.foreach**등으로 된 loop를 제거하고 **순수 javascript의 역행 루프**reversed loop로 변경하였습니다. 배열의 개수가 적을 때는 크게 상관 없지만 수백 수천개가 되면 그 때는 이야기가 다릅니다.

// length를 지역 변수에 미리 선언(매번확인하지 않는다), 루프 순서를 역으로

for (var i = delivery.length-1; i >= 0; i--) {

// for문 내부에 배열을 지역변수에 할당

var deliveryItem = delivery[i];

deliveryItem.deliveryStatus = "pickup";

};

코드 4. 역행 루프

* for (var i = delivery.length-1; i >= 0; i–) 루프 순서를 역으로 하면 조건문을 0 즉, false로 평가하게 하여 속성검색(조건문 비교)을 최소화하는 효과를 얻는다.
* for (**var i = delivery.length**;i >= 0; 배열의 크기를 미리 할당 해놓아 배열을 크기를 매번 확인하지 않는다.
* **var deliveryItem = delivery[i]** 배열을 미리 지역변수에 할당 해놓고 deliveryStatus 연산시 참조 하면 delivery[i].status, delivery[i].shopName 처럼 매번 배열을 검색하지 않는다.

**setTimeout을 사용하여 로직을 큐로 넘긴다.**

javascript는 **단일 쓰레드로 동작**하며. 먼저 수행된 작업이 끝날 때 까지 다음작업은 대기하게 됩니다. 무거운 작업이 있다면 당연히 사용자는 delay를 느끼게 되고. 이러한 점을 해결 하기 위해 **setTimeout을 이용하여 작업을 실행하면 javascript engine에서 UI 작업 큐로 작업은 넘겨** 지게 되고 event loop가 **큐의 쌓여 있는 task들을 처리** 하게 됨으로써 **blocking이 감소**하여 좀더 성능향상을 시킬 수 있습니다. [JAVASCRIPT Event Loop 링크](https://github.com/nhnent/fe.javascript/wiki/June-13-June-17,-2016)

setTimeout(function () {

Rider.updatePosition(position);

},0)

코드 5. setTimeout 사용