

최신 정보통신기술 Google에서 제공하는 Firebase의 실시간 데이터베이스 이용에 관한 연구

A Study on the realtime database systems using google firebase

저자 (Authors)	한예림, 양세현, 홍원길 Ye-Lim Han, Se-Hyeon Yang, Won-Gil Hong
출처 (Source)	한국통신학회 학술대회논문집 , 2017.6, 1272-1273 (2 pages) Proceedings of Symposium of the Korean Institute of communications and Information Sciences , 2017.6, 1272-1273 (2 pages)
발행처 (Publisher)	한국통신학회 Korea Institute Of Communication Sciences
URL	http://www.dbpia.co.kr/Article/NODE07218640
APA Style	한예림, 양세현, 홍원길 (2017). 최신 정보통신기술 Google에서 제공하는 Firebase의 실시간 데이터베이스 이용에 관한 연구. 한국통신학회 학술대회논문집, 1272-1273.
이용정보 (Accessed)	김포고등학교 183.109.35.*** 2018/08/02 16:29 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독 계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

최신 정보통신기술 Google에서 제공하는 Firebase의 실시간 데이터베이스 이용에 관한 연구

한예림, 양세현, 홍원길

중앙대학교

dpfla0376@hanmail.net tiamo5237@gmail.com wghong@cau.ac.kr

A Study on the realtime database systems using google firebase

Ye-Lim Han, Se-Hyeon Yang, Won-Gil Hong

Chung-Ang University

요 약

본 논문은 Google에서 제공하는 Firebase상의 실시간 데이터베이스를 이용해서 모든 클라이언트가 실시간으로 데이터를 동기화하고, 변경되는 데이터를 실시간으로 감지하고 이용하는 '클라이언트 간 interaction' 구현에 관한 논문이다. 클라이언트는 각각 아두이노 프로 미니와 버튼, LED모듈, 블루투스 모듈로 구성된 장치를 가지며 데이터베이스는 이 장치에 관한 정보를 가진다. 장치의 상태가 변하면 그 즉시 데이터베이스가 변경되고 Interaction을 수행한다.

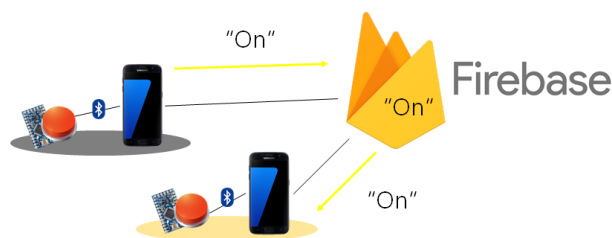
I. 서 론

본 논문에서는 Google의 Firebase 실시간 데이터베이스를 이용한다. 다수의 클라이언트가 있을 때 클라이언트는 interaction을 주고받을 클라이언트를 peer 클라이언트로 등록할 수 있다. 이 때 등록은 일방향이 아니라 서로를 모두 등록하는 양방향으로 한다.

클라이언트는 각각 아두이노 프로 미니, 버튼, LED모듈, 블루투스 모듈로 구성된 장치를 가진다. 이 장치는 안드로이드 애플리케이션과 블루투스 통신을 통해 데이터를 주고받는다. 장치는 상태 변경 시 그 내용을 애플리케이션에 전송한다. 애플리케이션은 수신된 장치 상태를 확인하고 변경할 수 있으며, 그 내용을 실시간으로 데이터베이스에 적용한다. 동시에 클라이언트 간 interaction을 수행한다.

이 프로그램에서는 특정 클라이언트가 버튼을 누르면 데이터베이스 내용이 변경되고, 등록된 peer가 내용 변경을 확인해 본인의 LED장치를 점등한다. 반대로 peer가 버튼을 누르면 데이터베이스 내용이 변경되면서 클라이언트 A도 자신의 LED를 점등한다.

프로그램 구조는 아래와 같다.



II. 본론

1. 장치의 연결과 통신

장치와 애플리케이션은 페어링 이후 블루투스 통신한다. 장치 연결을 위

해 고유한 uuid를 부여한 후 소켓을 연결한다. 연결 이후에 애플리케이션은 별도의 스레드에서 장치로부터의 수신을 대기(listen)한다. 예를 들어 장치A의 버튼을 누르면 장치는 버튼의 상태 변화를 감지하여 애플리케이션으로 'on'을 전송한다. 장치로부터 'on'을 수신한 애플리케이션은 해당하는 항목의 database의 값을 변경한다. 애플리케이션은 write를 통해 장치 상태 변경을 할 수 있으며 연결된 소켓의 스트림을 통해 바로 write한다.

2. Realtime Database(실시간 데이터베이스)

Google의 Firebase Realtime Database는 일반적인 Http 요청이 아닌 동기화를 사용하여 연결된 모든 기기가 데이터가 업데이트 되었을 때 수초 안에 변경 내용을 받아올 수 있다. 기기가 오프라인 상태일 때도 데이터 업데이트에 대한 이벤트는 실시간으로 누적 발생하며 기기가 네트워크에 연결되었을 때 데이터가 동기화되고 충돌이 자동 해결된다. Realtime Database는 별도의 보안 규칙을 가지고 있는데 개발자는 Firebase에서 제공하는 또다른 기능인 인증 기능을 통해 데이터 액세스 권한 및 기능을 제한할 수 있으나 본 논문에서는 해당 기능을 사용하지 않았다.

3. Realtime Database의 통신은 asynchronous하게 진행된다.

Realtime Database는 실시간 이벤트와 동기화를 제공하지만 해당 기능은 모두 asynchronous하게 진행된다. 즉 Firebase의 변경 사항이 기기에 전달되어 기기의 로컬 데이터가 변경될 때까지 기기는 block상태가 되지 않고 별도의 작업을 수행할 수 있다. 이는 기기가 Database에 데이터를 변경하는 write의 경우 문제가 되지 않고 오히려 더 좋은 성능을 보이지만 데이터를 받아오는 read의 경우 문제가 발생한다.

4. Realtime Database에서의 Read Issue

기기가 Realtime Database에서 데이터를 읽어오는 방법은 값 이벤트 리스너를 등록하는 것이다. Firebase의 Database는 JSON 형식을 취하고 있으며 이 값 이벤트 리스너는 각 child에 등록 가능하다. 리스너의 종류에는 `addValueEventListener()`와 `addListenerForSingleValueEvent()`를 사용할 수 있으며 후자의 경우는 일회성 리스너로 한번만 값을 읽어오고 전자의 경우는 리스너가 잔존하여 별도의 해제 과정을 거치지 않는 한 지속적으로 해당 경로의 값 변경에 대한 이벤트를 읽어온다.

기기에서 값 리스너를 통해 Database의 내용을 읽어와 어떤 작업을 수행하기 위해서는 데이터가 모두 기기의 로컬 데이터에 저장될 때까지 기기의 다른 작업들이 기다릴 수 있어야 한다. 따라서 기기가 Database로부터 어떤 데이터를 읽어올 때에는 해당 부분에 callback을 사용하여 기기의 다른 기능들이 block되도록 구현한다.

콜백의 구현과 설정은 아래 내용과 같이 한다.

```
read(); // Database로부터 값을 읽어오는 메소드.

Callback callback = new Callback() {

    // read()가 수행되고 나서 기기가 해야할 일
}

setCallback(callback); // callback을 설정
```

5. 프로젝트에의 적용

클라이언트들은 자신의 고유한 이름(id값)을 Database에 저장해 두었으며 연결이 이루어질 두 클라이언트 A와 B는 서로의 이름을 알고 있다고 가정한다. 이때 A는 B의 데이터에 대해 값 리스너를 설정하고 B는 A의 데이터에 대해 값 리스너를 설정한다.

간단한 데이터베이스의 구조는 다음 아래와 같다.

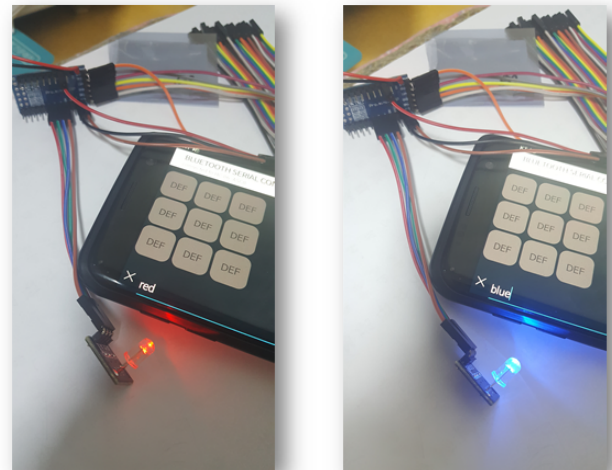
```
Project
└─clients
  └─"A"
    └─"connect" : "B"
    └─"button" : "off"
  └─"B"
    └─"connect" : "A"
    └─"button" : "off"
  └─"C"
    └─"button" : "off"
  └─"D"
    └─"button" : "off"
```

A가 버튼을 누르게 되면 A의 기기는 Firebase의 Database에 값을 갱신해야 한다. 즉 root인 Project의 하위 clients에 접근하여 key값이 A인 데이터의 value값을 "on"으로 write한다. 이때 A와 연결된 B는 A의 데이터에 대해 값 리스너를 가지고 있으므로 A의 value값이 "on"으로 변경된 것을 감지할 수 있다. 따라서 B는 기기의 LED등을 점등하라고 명령을 내릴 수 있었으며 Firebase의 Realtime Database 기능을 이용해 통신이 무사히 이루어짐을 확인할 수 있다.

6. 동작 확인

두 개의 장치에 각각 red와 blue색을 설정한 후 peer로 등록했다. Id는 각각 blue와 red이다. 이후 blue가 버튼을 누르면 데이터베이스의 내용이 변경되며, blue의 peer인 red의 LED가 점등된다.

반대의 경우로 red가 버튼을 누르면 위와 같은 방식으로 데이터베이스의 내용이 변경되고 blue의 LED가 점등된다.



III. 결론

본 논문에서는 구글의 Firebase 실시간 데이터베이스를 이용해 모든 클라이언트에서 실시간으로 데이터를 동기화하고 이를 이용해보았다. 이벤트 리스너, 콜백 등을 구현하여 클라이언트 간 interaction을 수행하며 클라이언트 간 interaction은 peer의 정보 변경 시 자신의 LED장치를 점등하는 것으로 시각화하였다.

ACKNOWLEDGMENT

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 서울어코드활성화 지원사업(IITP-2017-R0613-16-1205)의 연구결과로 수행되었음

참 고 문 헌

- [1] Watane, Swapnil, and Manbeer Singh. "Project Report: Transit Routes Management." (2016).
- [2] 정재곤, 이지스퍼블리싱, Do it! 안드로이드 앱 프로그래밍, 2017
- [3] GUPTA, Neetu; MANDAL, Ritabarta; CHADDHA, Vinay. Internet of Things based Boor Locking System-An Architecture. 2016.
- [4] Kumar, K. N., et al. "Implementing smart home using firebase." International Journal of Research in Engineering and Applied Sciences 6.10 (2016): 193-198.
- [5] Kumar, Rohit, et al. "An App-based Smart Interconnected Parking System." Computer Science & Telecommunications . 2016, Vol. 50 Issue 4, p114-122. 9p. 8 Diagrams.
- [6] Google Firebase Document(<https://firebase.google.com/products/database/>)