

알람기능 구현

작성: 이찬영

목차

1. web 앱 알람

- 구글 FCM 활용

2. kakaoAPI

- REST API 사용 방법
- DB 서버에서 API 사용 하는 방법

web 앱 알람 구현

- 구글 FCM 활용
 - FCM : Firebase Clud Messaging
 - 구글 FireBase 에서 제공하는 웹앱 푸쉬 기능을 사용
 - 장점 : 추가적 설치없이 웹앱에서 실행가능
 - 단점: IOS 에서 브라우저 push 알람 지원 X

Firestore?

- Firestore란? [<https://dev-youngjun.tistory.com/8>]
 - 모바일, 웹 애플리케이션 개발 플랫폼, 서버리스 애플리케이션 지원
 - BaaS(Backend as a Service)
- URL[<https://firebase.google.com>]

FCM 기본 설정

1. 계정 생성, 프로젝트 생성

- 구글 계정 / 계정 이름 만 설정하면 생성 가능
- 일정 트래픽 이하는 무료로 사용가능

2. 사용자 인증 정보 구성

- **Firebase Console** - **Setting** - **Cloud messaging** - **웹 구성**
- 웹 푸시 인증서 탭에서 키 쌍 생성을 클릭한다.
- 여기서 가져온 키를 잘 저장해 둔다

FCM 앱 개발

```
// 파이어 베이스 기본 설정, 도메인 root에 저장  
import '@firebase/messaging';
```

```
const config = {  
  projectId: "team-5dd61",  
  apiKey: "Your api key",  
  messagingSenderId: "Your sender ID",  
};
```

```
firebase.initializeApp(config);
```

```
const messaging = firebase.messaging()
```

```
messaging.usePublicVapidKey("Your Web push key");
```

```
// 사용자로부터 푸시 알람 허락 받아 토큰 값을 가져 온다.
```

```
Notification.requestPermission().then(function(permission) {  
  if (permission === 'granted') {  
    console.log('Notification permission granted.');  } else {  
    console.log('Unable to get permission to notify.');  }  
});
```

FCM 앱 개발

- 토큰 값 가져 오기

```
messaging.getToken().then(token =>{  
    console.log(token)  
})
```

FCM 앱 개발

- 메시지 생성, ajax 를 사용해서 요청

```
newPostPush(url,token){  
  const message = {  
    notification :{  
      "body" : "새 포스트가 등록되었습니다.",  
      "title": "진격의巨鳥",  
      "icon": "favicon.ico",  
      "click_action": "http://localhost:8080/",  
    },  
    to : token  
  }  
}
```



```
fetch(url ,{
  method: 'POST',
  body: JSON.stringify(message),
  headers: new Headers({
    'Content-type': 'application/json',
    'Authorization' : 'Your API Key'
  })
}).then(response => {
  if (response.status < 200 || response.status >= 400) {
    throw 'Error subscribing to topic: '+response.status + ' - ' + response.text();
  }
}).catch(e =>{
  console.log(e)
})
},
```

FCM 앱 개발

- Topic 설정 (해당 Topic을 구독한 여러명에게 전송 가능)

```
subscribeTokenToTopic(token, topic) {  
  fetch('https://iid.googleapis.com/iid/v1/'+token+'/rel/topics/'+topic, {  
    method: 'POST',  
    headers: new Headers({  
      'Authorization': 'Your API key'  
    })  
  }).then(response => {  
    if (response.status < 200 || response.status >= 400) {  
      throw 'Error subscribing to topic: ' + response.status + ' - ' + response.text();  
    }  
  }).catch(error => {  
    console.error(error);  
  })  
},
```

FCM 앱 개발

- 메시지 전송 코드 수정

```
newPostPush(url){  
    to : "/topics/post"  
}
```

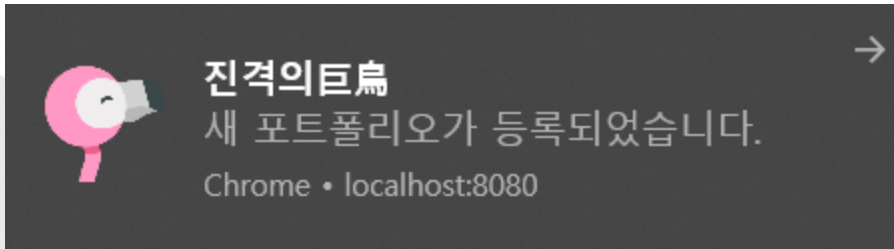
FCM 앱 개발

- 알람기능 구현 / 메시지를 받아서 알람을 띄워준다.

```
messaging.onMessage(function(payload){  
  console.log('onMessage: ', payload);  
  var title = payload.notification.title;  
  var options = {  
    body: payload.notification.body,  
    icon: payload.notification.icon  
  };  
  var notification = new Notification(title,options);  
});
```

FCM 앱 개발

- 결과 확인



- 참조 포스팅 [<https://dongsik93.github.io/til/2019/07/31/til-etc-fcm/>]

카카오 API

- 카카오에서 제공하는 open API
- 카카오 계정 개발자 등록으로 사용 가능
- URL[<https://developers.kakao.com/>]
- 메시지 공식 문서
[<https://developers.kakao.com/docs/latest/ko/message/common>]

카카오 API

1. 로그인 정보 입력 하여 자신의 uuid를 서버로 부터 받는다.
2. 친구 목록 받기 API를 전송할 친구의 uuid를 서버로 부터 요청 한다.
3. 친구의 uuid를 파라미터로 친구에게 기본 메시지 보내기 API를 사용하여 메시지 전송

로그인 API

- 카카오 인증 페이지로 리타이렉트 되고 카카오 인증 요청을 넣어서 인증 하게 된다.
- URL
 - 자바 스크립트
[<https://developers.kakao.com/docs/latest/ko/kakaologin/js>]
 - RestAPI[<https://developers.kakao.com/docs/latest/ko/kakaotalk-social/rest-api>]

```
Kakao.Auth.authorize({  
    redirectUri: 'https://developers.kakao.com/kakaoLogin.jsp'  
});
```


친구목록 가져오기 API

- 전체 목록을 가져오기때문에 따로 정리 기능이 필요하다
- URL
 - RestAPI
[<https://developers.kakao.com/docs/latest/ko/kakaotalk-social/rest-api>]
 - 자바스크립트
[<https://developers.kakao.com/docs/latest/ko/kakaotalk-social/js>]

```
Kakao.API.request({  
  url: '/v1/api/talk/friends',  
  success: function(response) {  
    console.log(response);  
  },  
  fail: function(error) {  
    console.log(error);  
  }  
});
```

친구에게 메시지 보내기 API

- 친구의 uuid를 받아서 API를 호출하면 전송 된다.
- 자바 스크립트
url[<https://developers.kakao.com/docs/latest/ko/message/js#send-friend-with-kakaotalk>]
- RestAPI[<https://developers.kakao.com/docs/latest/ko/message/rest-api>]

```
Kakao.API.request({
  url: '/v1/api/talk/friends/message/default/send',
  data: {
    receiver_uuids: ['receiver_uuids'],
    template_object: {
      object_type: 'feed',
      content: {
        title: '카카오톡 링크 4.0',
        description: '디폴트 템플릿 FEED',
        image_url:
          'http://mud-kage.kakao.co.kr/dn/Q2iNx/btqgeRgV54P/VLdBs9cvyn8BJXB3o7N8UK/kakaolink40_original.png',
        link: {
          web_url: 'https://developers.kakao.com',
          mobile_web_url: 'https://developers.kakao.com',
        },
      },
    },
    social: {
      like_count: 100,
      comment_count: 200,
    },
    button_title: '바로 확인',
  },
},
success: function(response) {
  console.log(response);
},
fail: function(error) {
  console.log(error);
},
});
```

DB 에서 직접 API 사용하기

- UDF(User define function)를 이용하여 만들어진 mysql-udf-httpmysql-udf-http lib 사용
- URL
 - 포스팅 [<https://thekeystore.blogspot.com/2019/04/mysql-http-post-json.html>]
 - GIT [<https://github.com/y-ken/mysql-udf-http>]

DB 에서 직접 API 사용하기

```
// 깃을 통해서 설치
git clone https://github.com/y-ken/mysql-udf-http.git

// 아래 코드 수정 하여 json 사용가능하도록
...
    if (curl)
    {
        struct curl_slist *hs=NULL;
        hs = curl_slist_append(hs, "Content-Type: application/json");
        curl_easy_setopt(curl, CURLOPT_HTTPHEADER, hs);
    }
...
```

DB 에서 직접 API 사용하기

// 함수 등록

```
create function http_post_json returns string soname 'mysql-udf-http.so';
```

// 정상 작동 확인

```
cd mysql-udf-http-master/
```

```
sudo CPPFLAGS="-I/usr/include/mysql" sh ./configure --with-mysql=/etc/my.cnf  
--enable-shared --libdir=/usr/lib64/mysql/plugin && sudo make && sudo make install
```

DB 에서 직접 API 사용하기

//mysql 콘솔에 함수들 등록

```
create function http_get returns string soname 'mysql-udf-http.so';  
create function http_post returns string soname 'mysql-udf-http.so';  
create function http_put returns string soname 'mysql-udf-http.so';  
create function http_delete returns string soname 'mysql-udf-http.so';
```


DB 에서 직접 API 사용하기

//사용법

```
SELECT http_get('<url>');  
SELECT http_post('<url>', '<data>');  
SELECT http_put('<url>', '<data>');  
SELECT http_delete('<url>');
```