

輪講ハンズオン

brewがある前提

全体の流れ

1. Node.js 用バージョン管理ツール(nvm)のインストール
2. nvm で Firebase が推奨するバージョンをインストール
3. npm で Firebase-Tools パッケージをインストール
4. Firebase-Tools でプロジェクト作成
5. npm で残りの必要パッケージをインストール
6. TypeScriptに必要なパッケージやトランスパイル周りについて
7. Firebase-Tools でデブロイ
8. VSCodeで開発しやすい環境準備

nvmのインストール(CLI)

```
$ brew install nvm

# ↓ インストール成功後
$ mkdir ~/.nvm

$ vim ~/.bash_profile

# ↓の2つを.bash_profileに追記します
export NVM_DIR=~/.nvm
source $(brew --prefix nvm)/nvm.sh

# インストール成功したか確認します
$ nvm --version
0.35.1 # 最新
```

node.jsのインストール(CLI)

```
$ nvm install v8.15.0
~~~~~
Checksums matched!
Now using node v8.15.0 (npm v6.4.1)
Creating default alias: default -> v8.15.0
```

nvm : Node.js 用バージョンマネージャー

npmでFirebase-Toolsパッケージをインストール(CLI)

```
$ npm --verison
6.4.1

$ npm install -g firebase-tools
```

npm : Node.js 用パッケージマネージャー

Web上でFirebase console を開く(web)

1. プロジェクト追加
2. プロジェクト名を指定->続行(1/3)
3. 2/3はそのまま
4. 3/3 アナリティクスを選択
5. 完了

Authenticationの追加(web)

1. ダッシュボードの Authentication を押す
2. google を選択
3. 有効にする

Cloud firestoreの追加(web)

1. ダッシュボードの Database を押す
2. 始める ボタンを押す
3. 次へ→完了

サンプルのクローン

```
$ git clone https://github.com/firebase/friendlychat-web.git
$ cd friendlychat-web/web-start
```

firebase のログイン(CLI)

```
$ firebase login
i Firebase optionally collects CLI usage and error reporting information to help
improve our products. Data is collected in accordance with Google's privacy policy
(https://policies.google.com/privacy) and is not used to identify you.

? Allow Firebase to collect CLI usage and error reporting information? Yes
```

-> webでgmailアカウントの選択画面に移動→承認

```
i To change your data collection preference at any time, run `firebase logout` and log
in again.

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-
fgrhgmd47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20http%
platform&response_type=code&state=885267642&redirect_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...

✓ Success! Logged in as {your gmail adress}
```

ユーザーの追加(CLI)

```
$ firebase use --add
```

ローカル環境で実行(CLI)

```
$ firebase serve --only hosting
i hosting: Serving hosting files from: ./public
✓ hosting: Local server: http://localhost:5000
```

Firestore SDK をインポート(VScode)

VScode で `friendlychat-web/web-start` を開く index.htmlを開く 以下が追加されていることを確認

```
<!-- Import and configure the Firebase SDK -->
<!-- These scripts are made available when the app is served or deployed on Firebase
Hosting -->
<!-- If you do not want to serve/host your project using Firebase Hosting see
https://firebase.google.com/docs/web/setup -->
<script src="/__/firebase/7.2.3/firebase-app.js"></script>
<script src="/__/firebase/7.2.3/firebase-auth.js"></script>
<script src="/__/firebase/7.2.3/firebase-storage.js"></script>
<script src="/__/firebase/7.2.3/firebase-messaging.js"></script>
<script src="/__/firebase/7.2.3/firebase-firestore.js"></script>
<script src="/__/firebase/7.2.3/firebase-performance.js"></script>
<script src="/__/firebase/init.js"></script>
```

ユーザー認証の追加(VScode)

```
function signIn() {
  var provider = new firebase.auth.GoogleAuthProvider();
  firebase.auth().signInWithPopup(provider);
}
```

```
function signOut() {
  firebase.auth().signOut();
}
```

認証状態の追跡(VScode)

```
function initFirebaseAuth() {
  firebase.auth().onAuthStateChanged(authStateObserver);
}
```

サインインしているユーザ情報の取得(VScode)

```
function getProfilePicUrl() {
  return firebase.auth().currentUser.photoURL || '/images/profile_placeholder.png';
}

function getUsername() {
  return firebase.auth().currentUser.displayName;
}
```

サインインしているかどうかを確認(VScode)

```
function isUserSignedIn() {  
    return !!firebase.auth().currentUser;  
}
```

以上のことが完了したら一度デプロイして確認する

```
$ firebase serve
```

localhost:5000/ を開く

メッセージの追加(VScode)

main.js 内の saveMessage() 関数で

```
function saveMessage(messageText) {  
    return firebase.firestore().collection('messages').add({  
        name: getUsername(),  
        text: messageText,  
        profilePicUrl: getProfilePicUrl(),  
        timestamp: firebase.firestore.FieldValue.serverTimestamp()  
    }).catch(function(error) {  
        console.error('Error writing new message to database', error);  
    });  
}
```

firebase console でメッセージを確認(web)

確認できたらOK

メッセージの取得(VScode)

main.js 内の loadMessage() 関数で

```
function loadMessages() {  
    var query = firebase.firestore()  
        .collection('messages')  
        .orderBy('timestamp', 'desc')  
        .limit(12);  
    query.onSnapshot(function(snapshot) {  
        snapshot.docChanges().forEach(function(change) {  
            if (change.type === 'removed') {  
                deleteMessage(change.doc.id);  
            } else {  
                var message = change.doc.data();  
                displayMessage(change.doc.id, message.timestamp, message.name,  
                    message.text, message.profilePicUrl,  
                    message.imageUrl);  
            }  
        });  
    });  
}
```

limit : メッセージの最大表示数

firebase console でメッセージを確認(web)

確認できたらOK

画像を送信

main.js 内の saveImageMessage() 関数で

```
function saveImageMessage(file) {
  firebase.firestore().collection('messages').add({
    name: getUsername(),
    imageUrl: LOADING_IMAGE_URL,
    profilePicUrl: getProfilePicUrl(),
    timestamp: firebase.firestore.FieldValue.serverTimestamp()
  }).then(function(messageRef) {
    var filePath = firebase.auth().currentUser.uid + '/' + messageRef.id + '/'
+ file.name;
    return
firebase.storage().ref(filePath).put(file).then(function(fileSnapshot) {
  return fileSnapshot.ref.getDownloadURL().then((url) => {
    return messageRef.update({
      imageUrl: url,
      storageUri: fileSnapshot.metadata.fullPath
    });
  });
})).catch(function(error) {
  console.error('There was an error uploading a file to CloudStorage:',error);
});
}
```

画像の保存先: Cloud strage

通知の設定

1. GCMの追加

web-start/public/manifest.json を開く 以下を追加

```
{
  "name": "Friendly Chat",
  "short_name": "Friendly Chat",
  "start_url": "/index.html",
  "display": "standalone",
  "orientation": "portrait",
  "gcm_sender_id": "103953800507"
}
```

2. FCMの追加

web-start/public/ に firebase-messaging-sw.js を新規作成 以下を追加

```
importScripts('/__/firebase/6.0.4/firebase-app.js');
importScripts('/__/firebase/6.0.4/firebase-messaging.js');
```

```
importScripts('/__/firebase/init.js');
firebase.messaging();
```

FCMのロード→初期化 処理

3. FCMデバイストークンの取得

public/scripts/main.js の saveMessagingDeviceToken に以下を追記

```
function saveMessagingDeviceToken() {
  firebase.messaging().getToken().then(function(currentToken) {
    if (currentToken) {
      console.log('Got FCM device token:', currentToken);
      firebase.firestore().collection('fcmTokens').doc(currentToken)
        .set({uid: firebase.auth().currentUser.uid});
    } else {
      requestNotificationsPermissions();
    }
  }).catch(function(error) {
    console.error('Unable to get messaging token.', error);
  });
}
```

4. 通知権限の取得

public/scripts/main.js の requestNotificationsPermissions に以下を追記

```
function requestNotificationsPermissions() {
  console.log('Requesting notifications permission...');
  firebase.messaging().requestPermission().then(function() {
    // Notification permission granted.
    saveMessagingDeviceToken();
  }).catch(function(error) {
    console.error('Unable to get permission to notify.', error);
  });
}
```

firebase console でメッセージを確認(web)

確認できたらOK

5. サーバーキーを取得(web)

1. Firebase console
2. 歯車マーク→ Project setting
3. Cloud Messaging
4. サーバーキーをコピー

6. リクエストの設定(CLI)

```
curl -H "Content-Type: application/json" \
-H "Authorization: key=YOUR_SERVER_KEY" \
-d '{
  "notification": {
    "title": "New chat message!",
```

```
"body": "There is a new message in FriendlyChat",  
"icon": "/images/profile_placeholder.png",  
"click_action": "http://localhost:5000"  
},  
"to": "YOUR_DEVICE_TOKEN"  
}' \  
https://fcm.googleapis.com/fcm/send
```