

РАЗВЕРТЫВАНИЕ ПРИЛОЖЕНИЙ В ОБЛАЧНОМ СЕРВИСЕ В РАМКАХ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ WEB-ПРОГРАММИРОВАНИЕ

Е. Д. Рафеенко, О. М. Кондратьева

*Белорусский государственный университет,
пр. Независимости 4, г. Минск, Беларусь, rafeenko@bsu.by, kondratjeva@bsu.by*

Рассматриваются особенности развертывания JavaScript и Angular приложений в облачном сервисе Firebase и использование в приложениях NoSQL базы данных Firestore. Данная задача актуальна при изучении дисциплины Web-программирование, так как позволяет обучающимся понять принципы работы приложения в глобальной сети.

Ключевые слова: облачная платформа; FireBase; FireStore.

При изучении дисциплины Web-программирование важным является умение запустить и протестировать приложение на внешнем хостинге. Для целей обучения хорошо подходила облачная PaaS (platform as a service) платформа Heroku, поддерживающая ряд языков программирования, в том числе Java [1]. Однако компания – владелец данного сервиса – отказалась от бесплатных тарифных планов. Поэтому пришлось выбрать другую платформу – FireBase, хотя она и не подходит для развертывания полноценного backend-a.

Firebase – облачная платформа [2], поддерживающая ряд языков программирования, в том числе, JavaScript. Firebase имеет бесплатные сервисы, такие как:

- хостинг для хранения статических файлов (таких как CSS, HTML, JavaScript) с использованием кода на стороне клиента;
- облачную СУБД класса NoSQL, позволяющую разработчикам приложений хранить данные и синхронизировать их между несколькими клиентами.

Чтобы добавить поддержку Firebase в JavaScript или Angular web-приложение, необходимо после регистрации на сайте Firebase (например, через Google account) создать проект в Firebase консоли. Затем можно зарегистрировать свое web-приложение в этом проекте. Для этого нужно ввести nickname приложения (nickname является внутренним идентификатором и виден только разработчику в консоли Firebase) и добавить Firebase хостинг для приложения, отметив соответствующий переключатель.

Следующим шагом является установка SDK и инициализация Firebase в web-приложении. На этом этапе используется node package manager (npm) и сборщик модулей webpack. На локальном компьютере в корне приложения необходимо и выполнить команду:

```
npm install firebase
```

Также для удобства разработки стоит установить утилиту Firebase CLI [2], которая позволяет выполнять команды Firebase в командной строке своего компьютера. Для установки используется команда:

```
npm install -g firebase-tools
```

Прежде, чем развертывать приложение на Firebase, его необходимо собрать сборщиком модулей, например, webpack [3]. Если будет разворачиваться Angular приложение, то для его сборки нужно в корне приложения выполнить команду:

```
ng build
```

После сборки в папке развертывания появятся статические файлы приложения – html, css, js, ts и др. По умолчанию для приложения Angular используется папка «dist\<имя приложения>».

Далее можно перейти непосредственно к разворачиванию приложения, для чего в Firebase CLI необходимо выполнить команды:

```
firebase login  
firebase init  
firebase deploy
```

В результате выполнения команды `init` в корне проекта будет создан файл `firebase.json`, в который будут записаны все настройки, выполненные на этапе инициализации. В результате успешного разворачивания в командной строке появится ссылка, по которой доступно приложение.

Взаимодействие с NoSQL базой данных Firestore

Cloud Firestore – это гибкая, масштабируемая NoSQL база данных для разработки web-приложений и серверов от Firebase и Google Cloud [4]. Для работы с ней необходимо в Firebase консоли в разделе `Firebase products and features` создать базу данных Cloud Firestore, которая состоит из коллекций и документов в коллекциях.

Инициализировать Firestore в Angular приложении можно в командной строке своего компьютера в корне приложения с помощью команды:

```
ng add @angular/fire
```

В результате выполнения данной команды в файл корневого модуля Angular приложения – `app.module.ts` – будет добавлен нужный импорт из модуля `@angular/fire/app`, а также добавлен код инициализации Firebase приложения:

```
provideFirebaseApp(() => initializeApp(environment.firebase)),  
provideFirestore(() => getFirestore())
```

В файлы `environment.ts` и `environment.prod.ts` будет записана конфигурация Firestore базы данных.

Пример получения «снимка» (snapshot) документов коллекции и извлечения из него списка с данными:

```
db=inject(Firestore);  
const q=query(collection(this.db, this.dbPath),      or-  
derBy("id"));  
const querySnapshot=await getDocs(q);  
const dataList=querySnapshot.docs.map(doc=> doc.data());
```

В статье рассмотрены возможности для хостинга web-приложений в облачном сервисе. Описанные подходы могут быть использованы в рамках лабораторного практикума по дисциплине Web-программирование.

Библиографические ссылки

1. Heroku DevCenter [Electronic resource]. URL: <https://devcenter.heroku.com> (date of access: 21.02.2024).

2. Developer documentation for Firebase [Electronic resource]. URL: <https://firebase.google.com/doc> (date of access: 21.02.2024).

3. Webpack concepts [Electronic resource]. URL: <https://webpack.js.org/concepts> (date of access: 18.01.2024).

4. Google Cloud Storage [Electronic resource]. URL: <https://cloud.google.com> (date of access: 16.01.2024).