

# — JAVASCRIPT —

*Firebase*

# — Firebase —

**Firebase** — поставщик облачных сервисов и приложений.



## Firebase

Основное направление Firebase — **облачная NoSQL БД** для real-time приложений, которая предоставляет API, позволяющее разработчикам хранить и синхронизировать данные между несколькими клиентами.

# — Firebase —

## СЕРВИСЫ:

### База данных для real-time приложений

Firebase предоставляет облачную NoSQL БД для real-time приложений как сервис. Данный сервис предоставляет API для разработчиков, позволяющий синхронизировать данные приложения между клиентами и хранить их в облаке FireBase.

Работа напрямую с БД реализована через REST сервисы для некоторых JavaScript фреймворков, таких как AngularJS, React, Ember.js и Backbone.js. Для шифрования данных разработчики могут воспользоваться API, предоставляемый также Firebase.

### Hosting

Firebase Hosting — это хостинг сервис, запущенный 13 мая, 2014. Данный сервис поддерживает хранение статических файлов, таких как **CSS, HTML, JavaScript**. Сервис передает файлы по Content Delivery Network (CDN) с использованием HTTPS и SSL.

### Simple Login

Firebase Simple Login — сервис, позволяющий аутентифицировать пользователей, используя код только на стороне клиента (client-side code). Данный сервис поддерживает вход в такие системы как **Facebook, GitHub, Twitter и Google**. В дополнение к этому, разработчики имеют возможность аутентифицировать пользователей с помощью данных, хранящихся в Firebase облаке.

# — Firebase —

**Firebase Console** — управление проектами. Создаем новый проект.

```
<script src="https://www.gstatic.com/firebasejs/4.1.1/firebase.js"></script>
<script>
  // Initialize Firebase
  // TODO: Replace with your project's customized code snippet
  var config = {
    apiKey: "<API_KEY>",
    authDomain: "<PROJECT_ID>.firebaseapp.com",
    databaseURL: "https://<DATABASE_NAME>.firebaseio.com",
    storageBucket: "<BUCKET>.appspot.com",
    messagingSenderId: "<SENDER_ID>",
  };
  firebase.initializeApp(config);
</script>
```

# — Firebase —

Возможна инициализация и настройка отдельных Firebase JavaScript SDK:

- `firebase-app` - The core `firebase` client (required).
- `firebase-auth` - Firebase Authentication (optional).
- `firebase-database` - The Firebase Realtime Database (optional).
- `firebase-storage` - Cloud Storage (optional).
- `firebase-messaging` - Firebase Cloud Messaging (optional).

```
<script src="https://www.gstatic.com/firebasejs/4.1.1/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/4.1.1/firebase-auth.js"></script>
<script src="https://www.gstatic.com/firebasejs/4.1.1/firebase-database.js"></script>
<script src="https://www.gstatic.com/firebasejs/4.1.1/firebase-messaging.js"></script>

<!-- Leave out Storage -->
<!-- <script src="https://www.gstatic.com/firebasejs/4.1.1/firebase-storage.js"></script> -->

<script>
  var config = {
    // ...
  };
  firebase.initializeApp(config);
</script>
```

# — Firebase —

Создаем скелет приложения, или уже в готовом проекте

```
$ npm install -g firebase-tools  
$ firebase init      # Generate a firebase.json (REQUIRED)  
$ firebase serve     # Start development server
```



.firebaserc



database.rules.json



firebase.json

# — Firebase Authentication —

## Password Authentication

В firebase console ВКЛЮЧИТЬ  
Password Authentication:

### Auth section:

On the *Sign in method* tab,  
enable the Email/password  
sign-in method and click Save

```
firebase.auth().createUserWithEmailAndPassword(email, password).catch(function(error) {  
  // Handle Errors here.  
  var errorCode = error.code;  
  var errorMessage = error.message;  
  // ...  
});
```

```
firebase.auth().signInWithEmailAndPassword(email, password).catch(function(error) {  
  // Handle Errors here.  
  var errorCode = error.code;  
  var errorMessage = error.message;  
  // ...  
});
```

```
firebase.auth().signOut().then(function() {  
  // Sign-out successful.  
}).catch(function(error) {  
  // An error happened.  
});
```

# — Firebase Authentication —

## Google Authentication

В firebase console ВКЛЮЧИТЬ  
Google Authentication:

### Auth section:

On the *Sign in method* tab,  
enable the Google sign-in  
method and click Save

```
var provider = new firebase.auth.GoogleAuthProvider();
```

```
firebase.auth().signInWithPopup(provider).then(function(result) {  
  // This gives you a Google Access Token. You can use it to access the Google API.  
  var token = result.credential.accessToken;  
  // The signed-in user info.  
  var user = result.user;  
  // ...  
}).catch(function(error) {  
  // Handle Errors here.  
  var errorCode = error.code;  
  var errorMessage = error.message;  
  // The email of the user's account used.  
  var email = error.email;  
  // The firebase.auth.AuthCredential type that was used.  
  var credential = error.credential;  
  // ...  
});
```

```
firebase.auth().signOut().then(function() {  
  // Sign-out successful.  
}).catch(function(error) {  
  // An error happened.  
});
```



# — Firebase Authentication —

```
firebase.auth().onAuthStateChanged(function(user) {  
  if (user) {  
    // User is signed in.  
  } else {  
    // No user is signed in.  
  }  
});
```

## Manage users

```
var user = firebase.auth().currentUser;  
  
if (user) {  
  // User is signed in.  
} else {  
  // No user is signed in.  
}
```

### Get a user's profile

To get a user's profile information, use the properties of an instance of `User`. For example:

```
var user = firebase.auth().currentUser;  
var name, email, photoUrl, uid, emailVerified;  
  
if (user != null) {  
  name = user.displayName;  
  email = user.email;  
  photoUrl = user.photoURL;  
  emailVerified = user.emailVerified;  
  uid = user.uid; // The user's ID, unique to the Firebase project. Do NOT use  
                  // this value to authenticate with your backend server, if  
                  // you have one. Use User.getToken() instead.  
}
```

# — JAVASCRIPT —

*Firebase*