

Add Firebase to a server Set up projects programmatically B Manage your Firebase projects Platforms and frameworks Prototype and test with Emulator get starten with Filebase Add Firebase to an app Apple platforms (iOS+) Android Flutter Add Firebase to a game Add Firebase to a server Set up projects programmatically B Manage your Firebase projects Prototype and test with Emulator ▼ Add Firebase to an ann Apple platforms (iOS+) Android Web Flutter Add Firebase to a game Set up projects programmatically B Manage your Firebase projects Platforms and frameworks Prototype and test with Emulator Add Firebase to an app Apple platforms (iOS+) Android Weh Flutter Add Firebase to a game Set up projects programmatically B Manage your Firebase projects Platforms and frameworks Prototype and test with Emulator

The example below shows how you could use the Cloud Firestore Lite SDK to retrieve a list of data.

```
• •
import { initializeApp } from 'firebase/app';
import { getFirestore, collection, getDocs } from 'firebase/firestore/lite';
// Follow this pattern to import other Firebase services
// import { } from 'firebase/<service>':
// TODO: Replace the following with your app's Firebase project configuration
const firebaseConfig = {
}:
const app = initializeApp(firebaseConfig);
const db = getFirestore(app);
// Get a list of cities from your database
async function getCities(db) {
 const citiesCol = collection(db, 'cities');
 const citySnapshot = await getDocs(citiesCol);
 const cityList = citySnapshot.docs.map(doc => doc.data());
 return cityList;
```

Step 4: Use a module bundler (webpack/Rollup) for size reduction

★ Note: You can skip this step if you are using a JavaScript framework CLI tool like the <u>Angular CLI</u>, <u>Next.js</u> [2], <u>Yue CLI</u> [2], or <u>Create React App</u> [2]. Check out <u>our guide on module bundling</u> for more information.

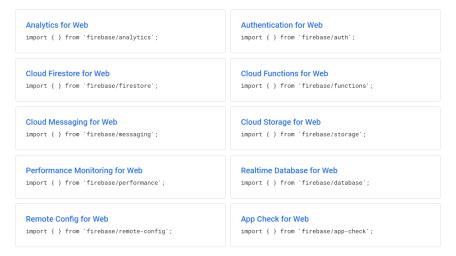
The Firebase Web SDK is designed to work with module bundlers to remove any unused code (tree-shaking). We strongly recommend using this approach for production apps. Tools such as the Angular CLI 💆, Next.js 💆, Vue CLI 💆, or Create React App 🔀 automatically handle module bundling for libraries installed through npm and imported into your codebase.

See our guide Using module bundlers with Firebase for more information.

Available Firebase services for web

Now that you're setup to use Firebase, you can start adding and using any of the following available Firebase services in your web app.

The following commands show how to import Firebase libraries installed locally with npm. For alternative import options, see the available libraries documentation.



Next steps

Learn about Firebase:

- Explore sample Firebase apps.
- Get hands-on experience with the Firebase Web Codelab.
- Explore the open source code in GitHub.
- Review the supported environments for the Firebase JavaScript SDK.
- Speed up your development with additional Firebase-maintained open source libraries, like AngularFire, RxFire, and FirebaseUI for web.
- Prepare to launch your app:
 - Set up budget alerts for your project in the Google Cloud Console.
 - Monitor the Usage and billing dashboard in the Firebase console to get an overall picture of your project's usage across multiple Firebase services.
 - Review the Firebase launch checklist

