



G's ACADEMY
TOKYO

補足資料

Firestore/Auth

サンプル最後にあります！！



Auth/Login

Google認証を使う

Firebase設定

「Authentication」 → 「ログイン方法」 → 「新しいプロバイダを追加」 → 「Google」の順番に選択

The screenshot shows the Firebase Authentication console interface. On the left sidebar, the 'Authentication' option is highlighted with a red box. A red arrow points from this box to the 'Sign-in method' tab in the main content area, which is also highlighted with a red box. Another red arrow points from the 'Sign-in method' tab to a blue button labeled '新しいプロバイダを追加' (Add new provider), which is also highlighted with a red box. Below this button, the 'Add login providers' section is shown. It is divided into 'ネイティブのプロバイダ' (Native providers) and '追加のプロバイダ' (Additional providers). The 'Google' provider is highlighted with a red box in the '追加のプロバイダ' section, with a red arrow pointing to it from the 'Add new provider' button. The 'ネイティブのプロバイダ' section includes 'メール/パス...' (Email/Password), '電話番号' (Phone number), and '匿名' (Anonymous). The '追加のプロバイダ' section includes 'Google', 'Apple', 'Yahoo!', 'Facebook', 'GitHub', 'Microsoft', 'Play ゲーム' (Play Games), 'Game Center', and 'Twitter'.

Firebase

プロジェクトの概要

構築

Authentication

Firestore Database

Realtime Database

Storage

Hosting

Functions

Machine Learning

リリースとモニタリング

Crashlytics、Performance、Test ...

分析

Dashboard、Realtime、Events、C...

エンゲージメント

Chat

Authentication

Users Sign-in method Templates Usage

ログインプロバイダ

ログイン方法を追加して Firebase Auth の利用を開始しましょう

ネイティブのプロバイダ

追加のプロバイダ

新しいプロバイダを追加

メール/パス...

電話番号

匿名

Google

Apple

Yahoo!

Facebook

GitHub

Microsoft

Play ゲーム

Game Center

Twitter

有効にする

①

☒ 有効にする

Google ログインが、接続済みの iOS およびウェブアプリ上で自動的に設定されます。Android アプリ用の Google ログインを設定するには、[\[プロジェクトの設定\]](#) でアプリごとに [SHA1 フィンガープリント](#) を追加する必要があります。

⚙️ 以下の[プロジェクトレベル設定](#)を更新して次に進む

プロジェクトの公開名 ①

Googleアカウントで登録しているメール

②

プロジェクトのサポートメール ①

外部プロジェクトの ID をホワイトリストに追加します（オプション） ③



ウェブ SDK 設定 ②



③

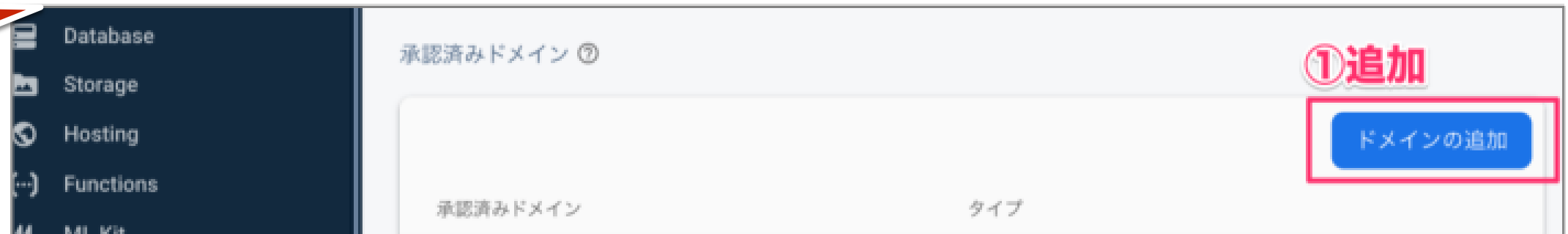
キャンセル

保存

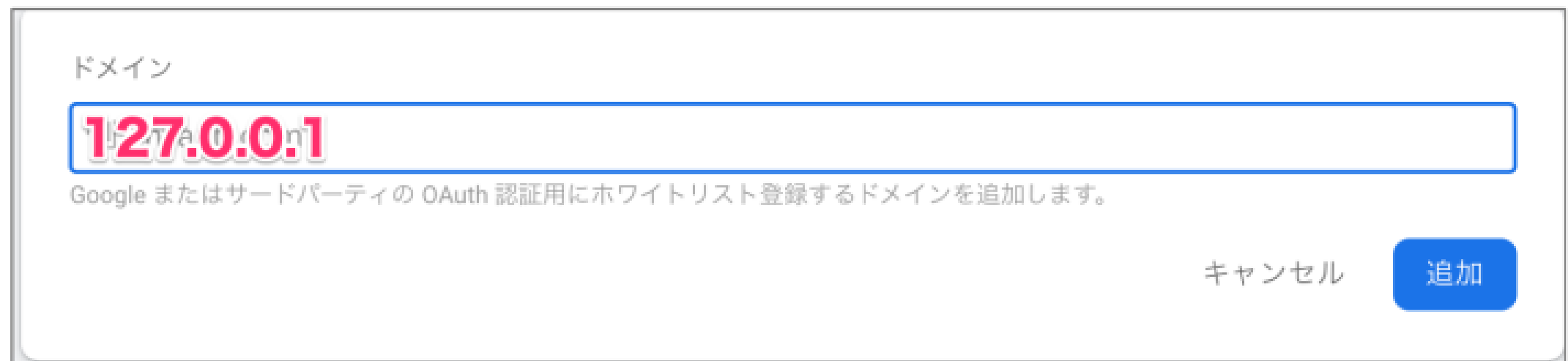
【承認済みドメインを追加: VisualStudioCode を使用した例】

- ・ 拡張機能: GoLive (プレビュー機能)を使用
- ・ "127.0.0.1" を追加 (http 通信でおこなう必要があり、ドメイン登録必須)

同じ画面



ドメイン「127.0.0.1」を追加



追加後に確認



※ Web サーバーにアップロードして確認する必要がある。(VScode の場合:GoLive)

ルールを更新

The screenshot shows the Firebase console interface for a project named 'authtest'. The left sidebar contains the 'Firebase' logo and a list of services: 'プロジェクトの概要', '構築' (with sub-items: Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), and 'リリースとモニタリング' (with sub-items: Crashlytics, Performance, Test Lab). The 'Realtime Database' item is highlighted with a red box. The main content area is titled 'Realtime Database' and has tabs for 'データ', 'ルール' (highlighted with a red box), 'バックアップ', and '使用状況'. Below the tabs are buttons for 'ルールを編集' and 'ルールをモニタリング'. A blue bar at the bottom of the main area contains buttons for '公開されていない変更', '公開' (highlighted with a red box), and '破棄'. Below this bar is a text editor showing the following JSON rules:

```
{  
  "rules": {  
    "users": {  
      "$uid": {  
        ".write": "$uid === auth.uid",  
        ".read": "$uid === auth.uid"  
      }  
    }  
  }  
}
```

Red arrows point from the 'ルール' tab to the '公開' button and from the 'ルール' tab to the text editor. A red box highlights the text editor area. A callout box at the bottom right contains the following text:

<書き換え>
rule.txt を用意してますので、
コピペしてください！！

補足：ルールの解説

難しければ飛ばしてください！

◇ ルールの解説

"users(データ階層名)" : {

"\$Suid(登録されてるID)" : {

.write : "登録しているID===現在認証してるユーザID"

.read : "登録しているID===現在認証してるユーザID"

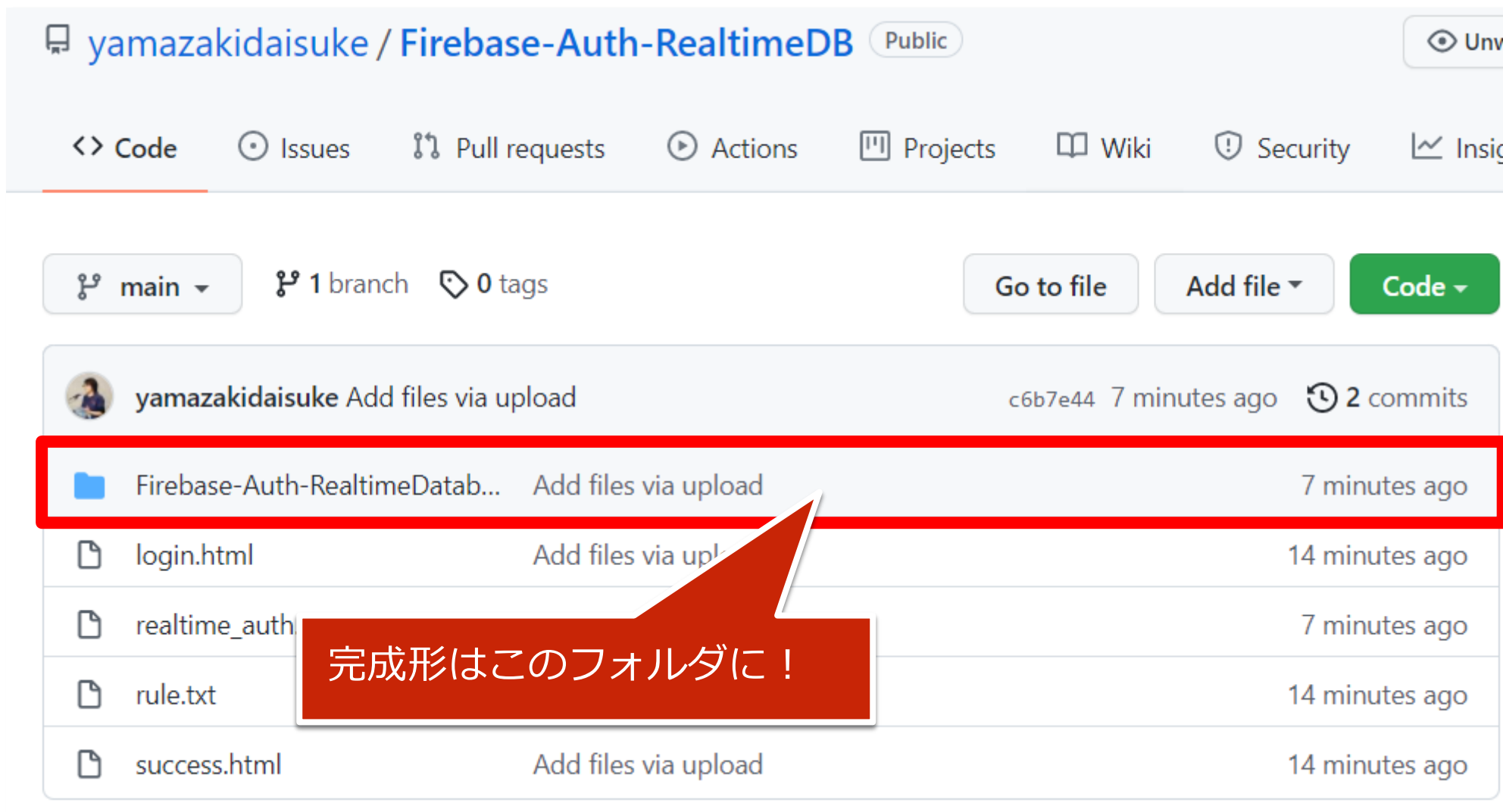
※.write (書き込みできる人) trueにすると全員OK

※.read (読み込みできる人) trueにすると全員OK

falseにすると全員NG

1. サンプルコード

<https://github.com/yamazakidaisuke/Firebase-Auth-RealtimeDB>



2. サンプルコード

<https://github.com/yamazakidaisuke/Firebase-Auth-RealtimeDB>

```
28 <!--** 以下Firebase **-->
29 <script type="module">
30   // Import the functions you need from the SDKs you need
31   import { initializeApp } from "https://www.gstatic.com/firebasejs/9.0.2/firebase-app.js";
32   // TODO: Add SDKs for Firebase products that you want to use
33   // https://firebase.google.com/docs/web/setup#available-libraries
34
35   // Your web app's Firebase configuration
36   const firebaseConfig = {
37     apiKey: "*****",
38     authDomain: "*****",
39     projectId: "*****",
40     storageBucket: "*****",
41     messagingSenderId: "*****",
42     appId: "*****:*****"
43   };
44
45   // Initialize Firebase
46   const app = initializeApp(firebaseConfig);
```

自分のKEYにしてください。