

Stappenplan: Firebase koppelen aan Articulate Storyline

Stap 1 – Voorbereiding

- Zorg dat je een Storyline-project hebt dat je kunt publiceren als **HTML5**.
- Log in met je **Google-account** (voor Firebase).
- Open eventueel **Notepad** of een tekstbestand om codes tijdelijk in op te slaan.

Stap 2 – Firebase-project aanmaken

1. Ga naar firebase.google.com.
2. Klik rechtsboven op **Go to console**.
3. Klik op **Create a project**.
4. Geef het project een naam, bijvoorbeeld *Articulate Escape Room*.
5. Zet **Google Analytics** uit (optioneel).
6. Klik op **Create project** en wacht tot het klaar is.
7. Klik daarna op **Continue**.

Stap 3 – Realtime Database instellen

1. Klik in de linkerbalk op **Build** → **Realtime Database**.
2. Klik op **Create Database**.
3. Kies een **serverlocatie** (bij voorkeur *Europa*).
4. Selecteer **Start in test mode** (voor ontwikkelfase).
 - Tip: gebruik testmodus alleen tijdens het bouwen.
 - In productie stel je later strengere beveiligingsregels in.
5. Klik op **Enable**.
→ Je database wordt nu aangemaakt.

Stap 4 – Web App toevoegen aan Firebase

1. Klik op het **tandwielicoon (Project settings)**.
2. Scroll naar **Your apps** en klik op het **webicoon (</>)**.
3. Geef de app een naam, bijvoorbeeld *Escape Room App*.
4. Klik op **Register app** (Firebase Hosting hoeft je niet aan te zetten).
5. Kopieer de **Firebase-configuratiecode** die verschijnt.

```
apiKey: "AIzaSyXXXXXXX-REDACTED-XXXXXXX",
authDomain: "exampleproject.firebaseio.com",
databaseURL: "https://exampleproject-default-
rtbd.firebaseio.com",
projectId: "exampleproject",
storageBucket: "exampleproject.firebaseio.com",
messagingSenderId: "000000000000",
appId: "1:000000000000:web:xxxxxxxxxxxxxxxxxxxxx"
```

6. Plak deze code in je Notepad-bestand – je hebt dit later nodig.
7. Klik op **Continue to console**.

Stap 5 – Firebase koppelen aan Storyline

1. Open je **Storyline-project**.
2. Ga naar de slide waar je de koppeling wilt maken.
3. Voeg een **trigger** toe: *Execute JavaScript → When timeline starts*.
4. Plak hier de Firebase-configuratiecode die je in stap 4 hebt gekopieerd.
5. Publiceer een testversie en open die in je browser.
6. Controleer in de **browserconsole** of er een melding verschijnt dat Firebase is geladen.

Stap 6 – Variabelen synchroniseren tussen spelers

1. Maak in Storyline de variabelen aan die spelers moeten delen, bijvoorbeeld:
 - codeEntry1, codeEntry2, codeEntry3, codeEntry4.
2. Voeg voor elke variabele twee triggers toe:
 - **Trigger 1 (bij wijziging)**: stuur de nieuwe waarde via JavaScript naar Firebase.
 - **Trigger 2 (bij start van de slide)**: luister naar Firebase en update de Storyline-variabele als er iets verandert.
3. Zo blijven de waarden tussen spelers automatisch synchroon.

Stap 7 – Extra functionaliteit (voorbeeld: hintknop)

1. Voeg een **button** toe (via *Insert* → *Button*).
2. Noem deze knop **Hint**.
3. Maak een **numerieke variabele** aan: hintCount (startwaarde = 0).
4. Voeg een tekstveld toe met een **reference** naar %hintCount% zodat het getal zichtbaar is.
5. Voeg een **JavaScript-trigger** toe aan de knop: verhoog hintCount en stuur de nieuwe waarde naar Firebase.

```
onFirebaseReady(function () {  
  var fb = _getFB();  
  var player = GetPlayer();  
  var roomId = (player.GetVar("RoomId") || "demo-room") + "";  
  
  var ref = fb.database().ref("rooms/" + roomId + "/hintsCount");  
  ref.transaction(  
    function (n) { return (typeof n === "number" ? n : 0) + 1; },  
    function (error, committed, snap) {  
      if (error) console.error("[FB] hintsCount transaction  
error:", error);  
      else if (!committed) console.warn("[FB] hintsCount transaction  
not committed");  
      else console.log("[FB] hintsCount now:", snap.val());  
    }  
  );  
});
```

6. Voeg een **listener-trigger** toe (When timeline starts): haal de actuele hintCount uit Firebase op.

```
onFirebaseReady(function () {  
  var fb = _getFB();  
  var player = GetPlayer();  
  var roomId = (player.GetVar("RoomId") || "demo-room") + "";  
  
  window._hintListeners = window._hintListeners || {};  
  if (window._hintListeners[roomId]) return;  
  window._hintListeners[roomId] = true;  
  
  var ref = fb.database().ref("rooms/" + roomId + "/hintsCount");  
  ref.on("value", function (snap) {  
    var n = Number(snap.val() || 0);  
    player.SetVar("HintsCount", n);  
    console.log("[FB] hintsCount -> Storyline:", n);  
  });  
});
```

7. Test of beide spelers tegelijk de juiste waarde zien.

Stap 8 – Publiceren en testen

1. Klik in Storyline op **Publish** → **Web**.
 2. Open de gepubliceerde HTML5-versie in twee verschillende browsers (bijv. Chrome en Edge).
 3. Test of wijzigingen direct bij de andere speler verschijnen.
 4. Open je **Firebase-console** en controleer of de data realtime wordt bijgewerkt.
-

Stap 9 – Veilig maken voor productie

1. Ga in Firebase naar **Realtime Database** → **Rules**.
 2. Pas de regels aan zodat niet iedereen kan lezen en schrijven.
 3. Vraag eventueel een AI-tool (zoals ChatGPT) om veilige regels voor jouw situatie.
 4. Zet je module daarna online of op je LMS.
-

Resultaat

Je hebt nu een werkende koppeling tussen **Firebase en Storyline**:

- Meerdere spelers kunnen tegelijk in één module werken.
- Gegevens worden realtime gedeeld via de cloud.
- De basis staat klaar voor extra functies zoals timers, scores of chat.

Heb je vragen of loop je vast?

Neem contact op met **Mark Spermon** en **Marloes Berkers**

via info@operationlistenup.nl