

## Pracovný list – Programujeme etudu 2.11 Hlasovanie na internete

**Úloha 1:** Vytvorte aplikáciu `pmz_2_11_kliker.aia`, ktorá bude zaznamenávať a vypisovať kliknutia viacerých používateľov pripojených na internet (napr. učiteľov na výlete, ktorí spočítavajú svojich žiakov vo viacerých skupinách, aby vedeli chatárovi nahlásiť záujem o dané jedlo alebo pitie).

Podme spoločne krok za krokom vyriešiť túto úlohu. Budeme postupovať nasledovne:

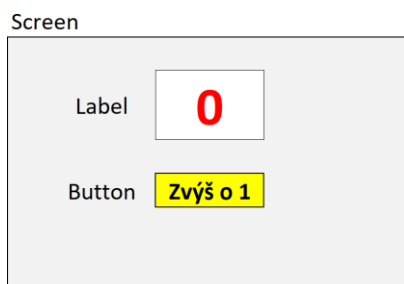
1. Navrhujeme funkcionality a používateľské rozhranie aplikácie
2. Navrhujeme štruktúru databázy
3. Na serveri Firebase pod svojim Google účtom zriadime vlastnú databázu s navrhnutou štruktúrou
4. Naprogramujeme aplikáciu s navrhnutými funkcionalitami

### 1 Návrh funkcionality a používateľského rozhrania aplikácie

Používateľské rozhranie bude obsahovať len dva vizuálne komponenty:

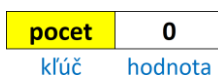
- tlačidlo na zvýšenie hodnoty spoločného počítadla o 1 (**Button**)
- popisok na výpis aktuálnej hodnoty spoločného počítadla (**Label**)

Na prácu s databázou použijeme nevizuálny komponent **Firebase** (uvedený v skupine Experimental)



### 2 Návrh štruktúry databázy

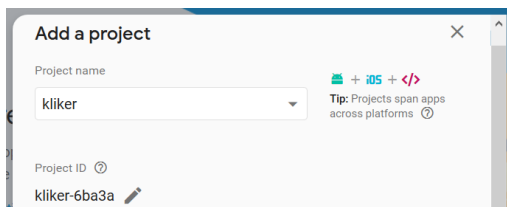
Naša databáza bude obsahovať len jeden kľúč **pocet**, v ktorom bude uložená aktuálna hodnota počtu žiakov (na začiatku 0).



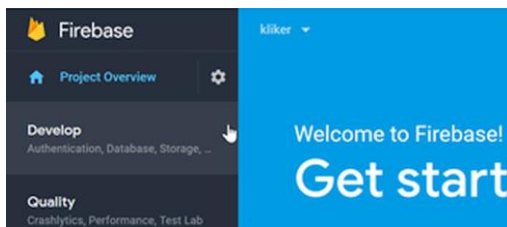
### 3 Zriadenie vlastnej databázy s navrhnutou štruktúrou na serveri Firebase

Navštívime webovú stránku <https://console.firebase.google.com/> a prihlásime sa na ňu svojim Google účtom.

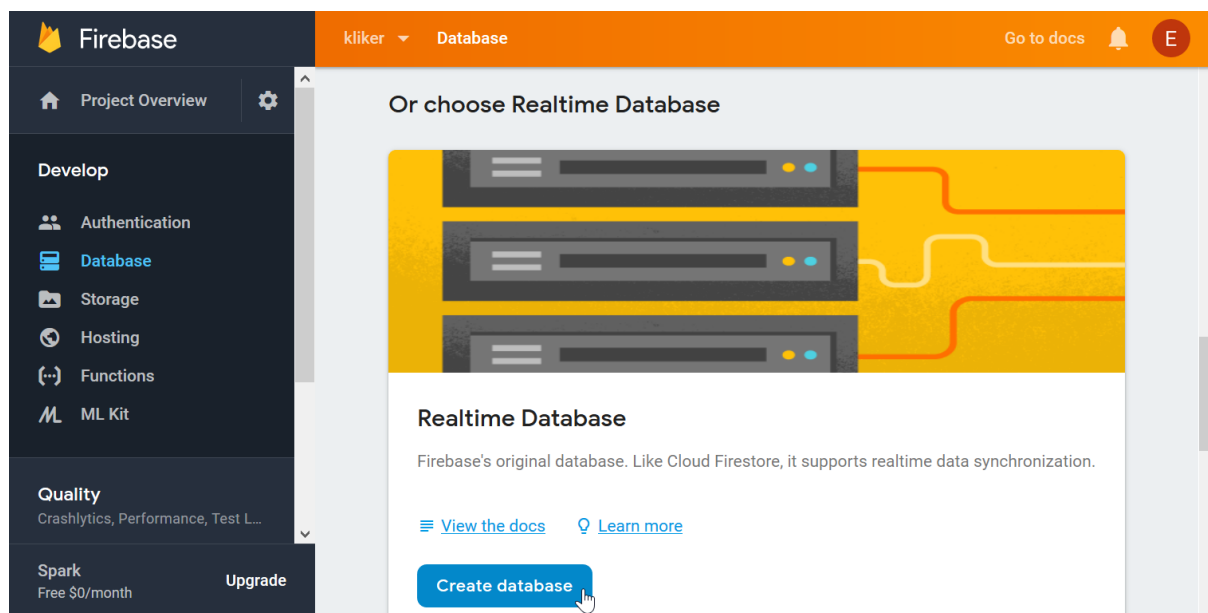
Stlačením tlačidla **Add project** na obrazovke vyvoláme okno, do ktorého uvedieme meno nášho projektu **kliker**. Nášmu projektu sa automaticky priradí **Project ID**, v našom prípade **kliker-6ba3a** (čo bude tiež menom našej vytvárateľnej databázy).



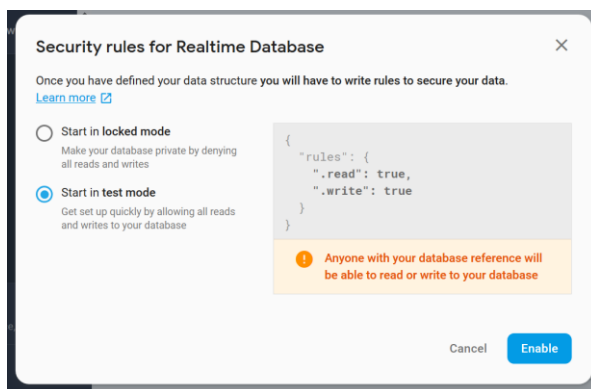
Po akceptovaní súhlasov bude v priebehu niekoľkých sekúnd vytvorený nový projekt, ktorý ďalej spravujeme.



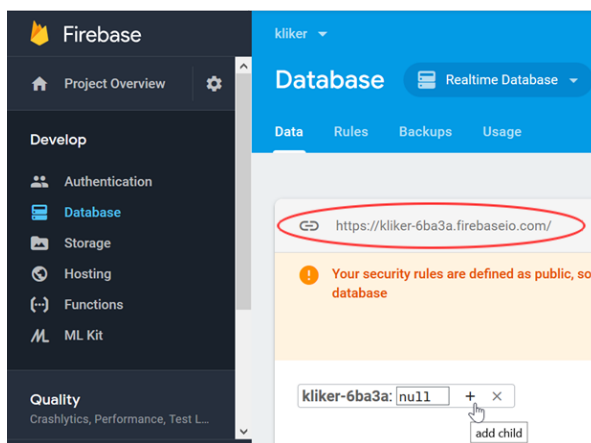
Teraz podme v tomto projekte vytvoriť novú databázu. Rozbalíme ponuku **Develop** v ľavej časti okna a vyberieme podponuku **Database**. Môžeme si vybrať jednu z dvoch druhov databáz: **Cloud Firestore** alebo **Realtime Database**. Pre naše účely nám poslúži tradičná praxou overená **Realtime Database**, ktorú vyberieme tlačidlom v dolnej časti okna.



Počas vytvárania databázy sme vyzvaní, aby sme si vybrali jeden z dvoch bezpečnostných režimov – **zamknutý** (locked mode) alebo **testovací** (test mode). My si vyberieme **testovací režim**, aby sme umožnili používateľom aplikácie čítať a zapisovať do našej databázy.



Po odkliknutí tlačidla **Enable** sa objaví obrazovka, v strede ktorej je uvedené URL našej databázy (zvýraznené červenou elipsou). V dolnej časti môžeme do našej databázy **kliker-6ba3a** pridávať kľúče s hodnotami.



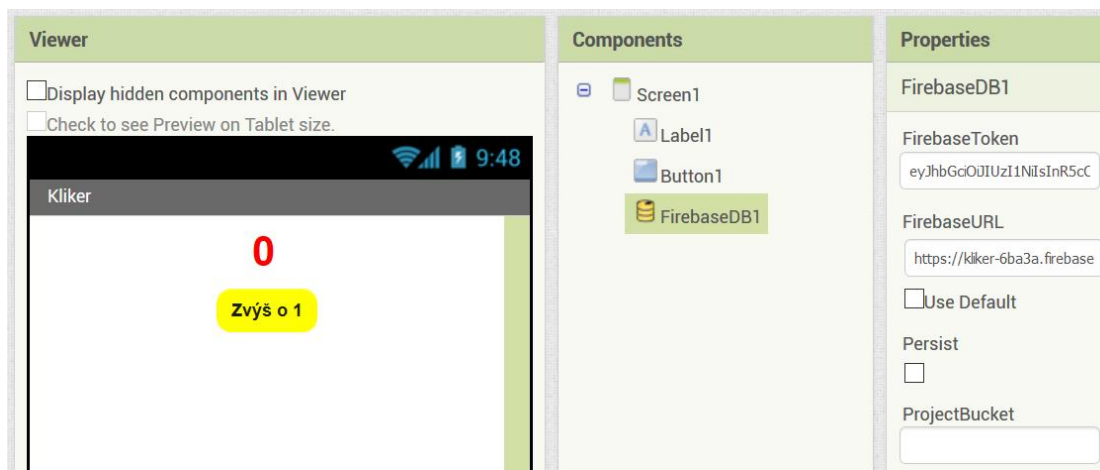
Pridaním kľúča **pocet** a jeho hodnoty **0** sme ukončili proces vytvárania štruktúry databázy a jej prvého obsahu.

**kliker-6ba3a**  
.....  
**pocet: 0**

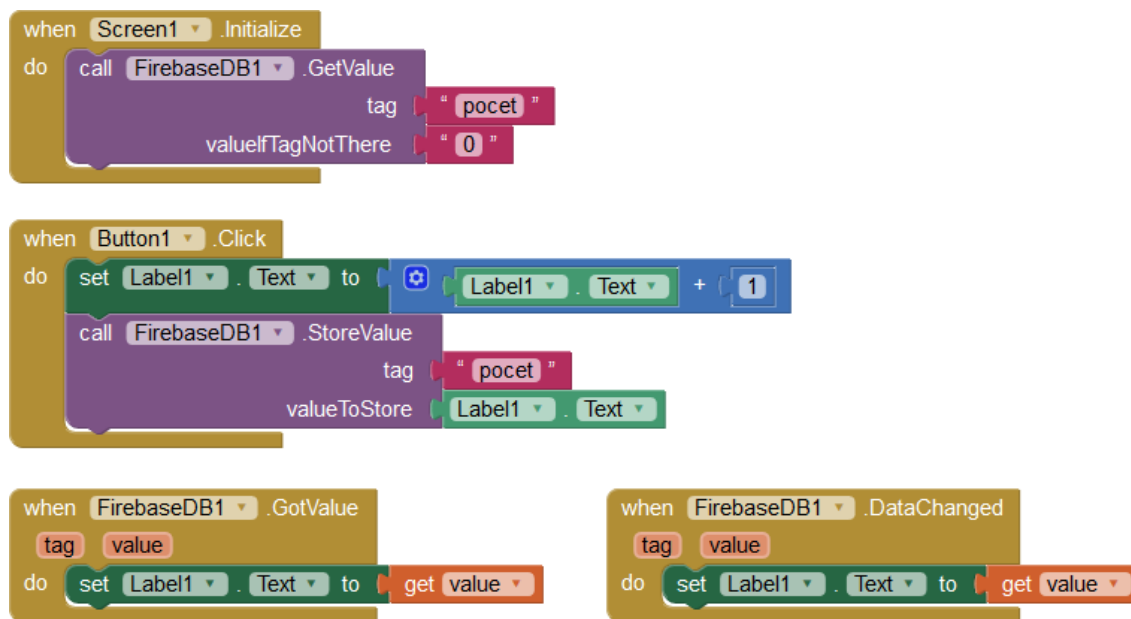
#### 4 Naprogramovanie aplikácie s navrhnutými funkcionalitami

V režime **Designer** nastavíme komponentu **Firebase**:

- vlastnosť **FirebaseURL** z pôvodnej hodnoty „DEFAULT“ na hodnotu **https://kliker-6ba3a.firebaseio.com/**
- odznačíme vlastnosť **Use Default**
- vlastnosť **project Bucket** z pôvodnej hodnoty „pmz\_2\_11\_kliker“ na prázdnu hodnotu.



V režime **Blocks** vytvoríme nasledovný zdrojový kód:



Po spustení aplikácie (vyvolaní udalosti **Screen.Initialize**) je zaslaná do databázy požiadavka na získanie hodnoty kľúča **pocet**. Ak databáza vie vrátiť hodnotu kľúča vyvolá sa udalosť **FirebaseDB.GotValue**, ktorá nastaví hodnotu popisku **Label1** na hodnotu získanú z databázy, ktorá je uložená v parametri **value** tejto udalosti.

Po stlačení tlačidla **Button1** sa zvýši hodnota popisku **Label1** o 1 a táto hodnota sa zapíše do databázy do kľúča **pocet**. Pri zmene kľúča **pocet** v databáze rôznymi používateľmi tejto aplikácie sa vyvolá udalosť **Firebase.DataChanged**, ktorá nastaví hodnotu popisku **Label1** na hodnotu získanú z databázy, ktorá je uložená v parametri **value** tejto udalosti.

Otázka	Odpoveď
a. Prečo metóda <b>Firebase.Getvalue</b> neposkytne programu priamo hodnotu daného kľúča ako to robí lokálna databáza <b>TinyDB</b> , ale tá hodnota sa získa až po vyvolaní udalosti <b>Firebase.GotValue</b> ?	a. ....
b. Čo majú spoločné a čo rozdielne udalosti <b>Firebase.GotValue</b> a <b>Firebase.DataChanged</b> ?	b. spoločné ..... rozdielne .....

**Úloha 2:** Vytvorte hlasovaciu aplikáciu `pmz_2_11_hlasovanie.aia`, ktorá umožní používateľom výber jednej z troch možností A, B, C. Bude ich tiež informovať, ktorá z možností hlasovania A, B, C je víťazom hlasovania.

(Odporúčanie: Pri spracovaní udalosti **Firestore.GotValue** a **Firestore.DataChanged** je potrebné pomocou podmienok rozlíšiť, ktorý kľúč (**tag**) sa načítal a ďalej spracovávať hodnotu (**value**) odpovedajúcu tomuto kľúču. Pri výpočte víťaza hlasovania sa dajú použiť funkcie na spracovanie zoznamu **select list item, index in list, make a list** a matematickú funkciu **max**.

Viac informácií o použití databázy Firebase pri programovaní v App Inventore nájdete na stránkach MIT, Experimental Components – App Inventor for Android:

<http://ai2.appinventor.mit.edu/reference/components/experimental.html#FirestoreDB>)