Quizapp

ארכיטקטורה והסבר על הקוד

תוכן עניינים

[תוספים 2](#_Toc3035590)

[FireBase 3](#_Toc3035591)

[פרויקט FireBase Functions 4](#_Toc3035592)

[המחלקה QuizApp 5](#_Toc3035593)

[המחלקה FireBaseHandler 6](#_Toc3035594)

[המחלקה BatteryBroadcastReceiver 7](#_Toc3035595)

[המחלקה ExitDialog 7](#_Toc3035596)

[המחלקה Question 7](#_Toc3035597)

[המחלקה Quiz 8](#_Toc3035598)

[המחלקה QuestionFragment 8](#_Toc3035599)

[המחלקה ScoreFragment 8](#_Toc3035600)

[המחלקה MainActivity 9](#_Toc3035601)

[המחלקה QuestionListActivity 9](#_Toc3035602)

[המחלקה QuestionAdapter 9](#_Toc3035603)

# 

# 

# תוספים

להלן התוספים שהשתמשנו בהם.

* [Zoomage](https://github.com/jsibbold/zoomage) – תוסף המאפשר תצוגת תמונות באופן חכם: הגדלה / הקטנה והזזה שלהם.
* [Android Universal Image Loader](https://github.com/nostra13/Android-Universal-Image-Loader) – תוסף המאפשר טעינת תמונות מהירה וקלה.
* [Firebase](https://firebase.google.com/) – טעינת שאלות ממסד, קבלת התראות בעת עדכון המסד.

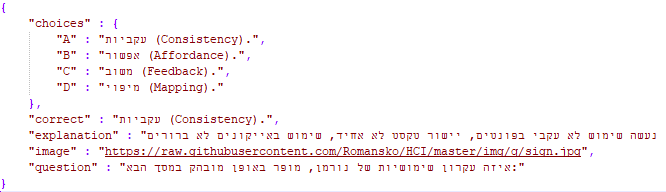
# FireBase

הגישה לFireBase הינה דרך: <https://console.firebase.google.com>

החשבון שאיתו נעשה שימוש הינו:

[RKCSQuizApp@gmail.com](mailto:RKCSQuizApp@gmail.com)  
quizapp12

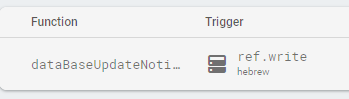
גישה לשאלות דרך סרגל כלים מצד שמאל: **Develop -> Database**

השאלות נשמרות בצורת קובץ json. דוגמא לשאלה:  


**הסבר לשאלה** / **תמונה** לא חובה ואם השדות ריקים, פשוט לא יוצגו באפליקציה.

כל השאלות שמורות תחת צומת יחיד (Root Node) בשם המפתח **Hebrew.**

בנוסף, נעשה שימוש ב [**FireBase Cloud Messaging (FCM) Functions**](https://github.com/firebase/functions-samples)על מנת לקבל התראה על עדכון במסד. אל פונקציונליות זו, ניתן לגשת גם דרך סרגל הכלים מצד שמאל: **Develop -> Functions**

שם נראה כי קיימת פונקציה **dataBaseUpdateNotification** (שאנו הגדרנו) שמאזינה לכל כתיבה אל מסד השאלות:  


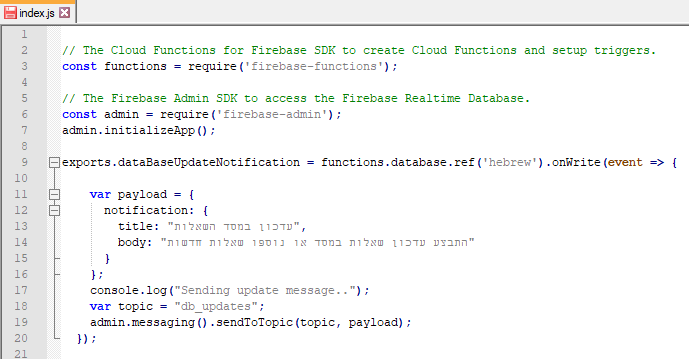
## פרויקט FireBase Functions

בתוך הפרויקט של QuizApp, יש פרויקט (**לא** Android) תחת תיקיית FireBase שנוצר בכדי להוסיף את הפונקציה שמאזינה על מסד השאלות.

הפרויקט נוצר ע"י הפקודות הבאות **בTerminal** (שמופנה לתיקייה הנ"ל):

|  |
| --- |
| npm install -g firebase-tools  firebase login --interactive  firebase init |

את הפונקציה **dataBaseUpdateNotification** הוספנו בקובץ QuizApp/FireBase/functions/**index.js**



נשים לב למפתח **“db\_updates”** (Topic). באפליקציה שלנו הגדרנו מאזין לפי Topic ולכן המפתח חשוב. (FireBase תומכת באפשרות לעשות גם האזנה לפי Token או לפי Groups).

# המחלקה QuizApp

|  |
| --- |
|  |

זוהי נקודת הכניסה לאפליקציה. השימוש במחלקה זאת הוא בכדי לאתחל **ImageLoader** (תוסף) סטטי בשביל טעינת תמונות קלה יותר בהמשך.

כמו כן, כאן שמורה המחרוזת **QATAG** בשביל הLog.

הגישה לטוען התמונות ולמחרוזת היא בצורה סטטית.

# המחלקה FireBaseHandler

המחלקה יורשת מ**FirebaseMessagingService** והיא אחראית לתקשורת בין האפליקציה לבין מסד הFireBase.

הפונקציונליות העיקרית של המחלקה היא טעינת שאלות משרת הFireBase. זה נעשה על ידי הפונקציה **getQuestions.**

|  |
| --- |
| public static void getQuestions**(**final IResponse caller**,** boolean hardReset**)** |

פונקציה זו אחראית על קריאת השאלות מהשרת באינטרנט. **IResponse** זהו ממשק של המחלקה שצריך לממש כאשר מחלקה אחרת קוראת לפונקציה בכדי לקבל את השאלות. **הממשק** נראה כך:



כאשר הפונקציה מקבלת את השאלות בצורת אובייקטים גנריים מהמסד, היא בונה אותם לפי צרכי האפליקציה באמצעות הפונקציה הסטטית ***parseQuestions.*** מוחזר מערך מסוג Question.

|  |
| --- |
| private static ArrayList**<**Question**>** parseQuestions**(**Object fbObject**)** |

כאשר השאלות מוכנות, המחלקה שומרת עותק של מערך השאלות בצורה סטטית, כך שבפעם הבאה שהשאלות יתבקשו, היא תחזיר עותק זה ולא תיגש שוב לאינטרנט ותבזבז זמן על תקשורת ובניית אובייקטים.

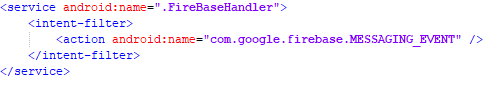
אם הדגל **hardReset** הוא **true,** אז תתבצע גישה למסד האינטרנטי והעותק השמור ידרס.

הפונקציונליות השנייה של המחלקה היא האזנה לשינויים במסד הFireBase. זה נעשה ע"י הפונקציה **subscribeToUpdates** שרושמת את האפליקציה להאזנה עם המפתח “**qb\_updates**”.

כאשר האפליקציה פתוחה וישנו שינוי בשרת, הפונקציה שתקרא במחלקה היא **onMessageReceived**. זוהי פונקציה שנדרסת מהמחלקה מ**FirebaseMessagingService.**

**onMessageReceived** תודיע למשתמש שהיה שינוי במסד ותקפיץ popup אם לטעון מחדש את השאלות או לא. אם המשתמש יבחר לא, לא יקרה כלום. אם כן, אז השאלות יטענו מחדש מהמסד. יש צורך לממש את הממשק הפנימי ISubscribeHandler בשביל הפונקציונליות הזו. השאלה נשאלת והטעינה לא מבוצעת אוטומטית בשביל לא להרוס למשתמש את ההתקדמות שביצע בסבב השאלות.

כאשר האפליקציה **כבויה**, אם היה שינוי בשרת אז הפלאפון יקבל **push notification** המודיע על שינויים במסד. זה נעשה ע"י service שהוגדר ב **manifest**:



# המחלקה BatteryBroadcastReceiver

מחלקה זו יורשת מ **BroadcastReceiver**והיא אחראית להודיע למשתמש על אחוז סוללה נמוך. האחוז מוגדר HardCoded דרך השדה ***THRESHOLD*** אשר כרגע מוגדר להיות 50.

הרישום להאזנה על הסוללה נעשה דרך הבנאי של המחלקה שמקבל כפרמטר את הActivity שקורא לו.

כל שינוי בסוללה יקרא לפונקציה **onReceive.** שם, אם האחוז קטן מ ***THRESHOLD*** שהוגדר, תוצג למשתמש הודעת popup שתתריע על האחוז הנמוך ותשאל אותו האם הוא רוצה לצאת מאפליקציה או להמשיך.

אם המשתמש יבחר להמשיך, הרישום למאזין יתבטל ולא יתחדש עד שהאפליקציה תופעל מחדש.

# המחלקה ExitDialog

מחלקה זו יורשת מ **DialogFragment.** בעת יצירת אובייקט חדש מסוג זה, יוצר דיאלוג שיוודא שהמשתמש אכן רוצה לצאת. אם ילחץ כן, האפליקציה תיסגר. לשם כך יש את הממשק הפנימי **IExitListener** המכיל את הפונקציה **onExitPressed** שתיקרא כאשר המשתמש יבחר לצאת.

# המחלקה Question

מחלקה זו מייצגת שאלה במערכת. היא מכילה את מחזורות השאלה, תשובות, הסבר וקישור לתמונה.

המחלקה מממשת את הממשק **Clonable** עבור העתקת השאלה. המחלקה מממשת את הממשק **Parcable** בכדי שיהיה ניתן לשמור שאלות כBundle בעת העברת לActivity או Fragment אחרי.

# המחלקה Quiz

זוהי מחלקת **הליבה** של האפליקציה. היא מכילה את הלוגיקה העיקרית. מחלקה זו מנהלת את הבחן הנוכחי שהמשתמש נבחן עליו.

הבנאי של המחלקה הוא פרטי וקבלת עותק נעשה ע"י הפונקציה **getInstance.** המחלקה אחראית לקרוא את העדפת המשתמש לגבי מספר שאלות בבוחן. אם לא הוגדר, ברירת המחדל היא 10.

המחלקה אחראית למלא בוחן בשאלות שהמשתמש עדיין לא נבחן בהם, כאשר סבב השאלות נגמר, המחלקה טוענת מחדש את העותק הסטטי של השאלות שנשמר מהמסד. בנוסף, המחלקה מציגה את השאלות והתשובות בסדר אקראי.

# המחלקה QuestionFragment

מחלקה זו יורשת מFragment והיא אחראית להציג שאלה למשתמש. כלומר היא מציגה את השאלה, תשובות ותמונה אם יש. אחרי מענה המשתמש, אם קיים הסבר לשאלה אז גם הוא יוצג.

ניתן למלא את הGUI ע"י הפונקציה **populateQuestion.**

יש שני קבצי xml שמשויכים עם מחלקה זו. אחד עבור portrait ואחד עבור landscape. שניהם קרויים בשם **fragment\_question.xml.** קבצי XML אלו, משתמשים בקובץ zoomage.xml שהוא בעצם מעטפת לתוסף zoomage ומיועד עבור הצגת תמונות. הוספת הzoomage נעשית באופן דינאמי.

למחלקה יש מחלקה פנימית בשם **ChoiceClickHandler** אשר מממשת את הממשק **View.OnClickListener** ומטרתה היא לצבוע בכחול תשובות שהמשתמש לוחץ עליהם.

כמו כן למחלקה יש ממשק פנימי בשם **ICorrectAnswerListener** המאזין ללחיצת תשובה נכונה ומפעיל את הפונקציה **correctQuestionPressed.**

# המחלקה ScoreFragment

מחלקה זו יורשת מFragment. היא אחראית להציג את התוצאה (ניקוד) של הבוחן בסיומו.

קבלת אובייקט מסוג זה נעשה ע"י הפונקציה **newInstance.**

המחלקה משתמשת בקצבי **fragment\_score.xml** עבור portrait / landscape.

# המחלקה MainActivity

זהו הActivity הראשי של האפליקציה. הוא משתמש בקבצי **activity\_main.xml** עבור portrait / landscape. קבצי הFragment של הScore / Question מוכלים בקבצי XML אלה באופן דינאמי.

המחלקה מממשת את הממשק **FireBaseHandler.IResponse** עבור קבלת השאלות מFireBase, ובעת קבלה היא יוצרת אובייקט מסוג **Quiz** חדש.

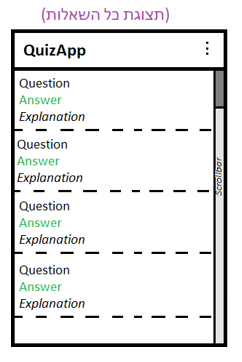
המחלקה מממשת את הממשק **FireBaseHandler.ISubscribeHandler** עבור הצגת דיאלוג כאשר יש שינוי במסד.

המחלקה מממשת את ExitDialog.IExitListener עבור האזנה ליציאה מהאפליקציה.

המחלקה מממשת את **QuestionFragment.ICorrectAnswerListener** עבור טיפול בלחיצה על תשובה נכונה – הוספת ניקוד דרך **Quiz**. כלומר ה **QuestionFragment**מתריע ל**MainActivity** שהמשתמש לחץ תשובה נכונה, וזו מעבירה את זה למחלקת **Quiz** ששומרת את הניקוד והתוצאה.

המחלקה הזו גם שומרת את לוגיקת ה **Menu**בתוכה. נעשה שימוש בקובץ **menu.xml** ובעת לחיצה על אחד מהפריטים של התפריט, הקוד המטפל נמצא בתוך מחלקה זו.

# המחלקה QuestionListActivity



מחלקה זו אחראית לפרוס את השאלות בצורה כללית כמתואר בהצעה:  
  
עבור פריט בודד של שאלה, נעשה שימוש בקובץ **question\_view.xml.**

המחלקה פורסת את השאלות, התשובות ואת ההסברים ותמונות אם קיימים בListActivity חדש.

אם קיימת תמונה, בעת לחיצה עליה יפתח התוסף zoomage שיציג אותה ויאפשר משחק עם התמונה. (בתצוגה הרגילה לא ניתן כיוון שזה תמונה רגילה ולא zoomage). הוספת מאזין הלחיצה נקבע במחלקה זו.

המחלקה הזו עושה שימוש במחלקת **QuestionAdapter.**

העברת השאלות לתצוגה מMainActivity לListAcitivty דרשה שהשאלות יהיו מסוג **Parcable**.

# המחלקה QuestionAdapter

מחלקה זו עובדת עבור מחלקת **QuestionListActivity.** היא אחראית על לוגיקת פריט שאלה בודד בתוך כל הרשימה. כלומר מחלקה זו אחראית להציג עבור כל שאלה את מחרוזת השאלה, תשובה הסבר ותמונות אם יש. כמו כן שם נקבע צבע הBackground עבור כל פריט: אם הפריט במיקום זוגי – צבע לבן. אחרת, תכלת.