

## א'יור. 1-22



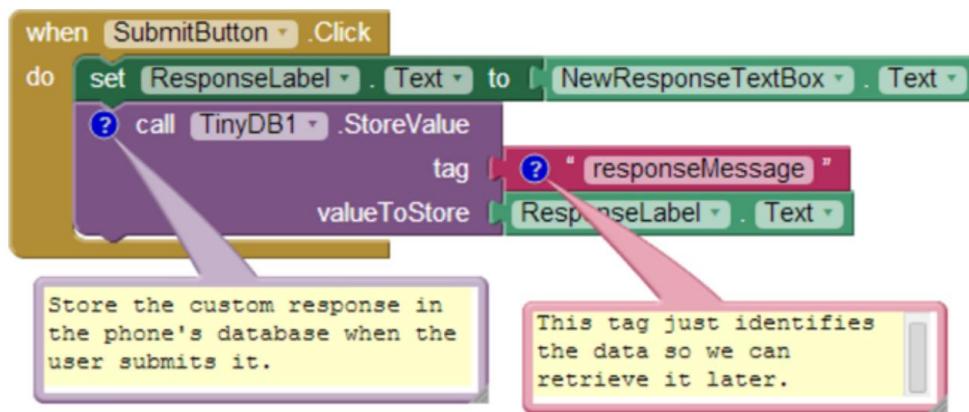
כדוגמה, שקול את אפליקציית No Texting While Driving של פרק 4, ששולחת תגובה אוטומטית להודעות טקסט נכנסות. לאפליקציה יש תגובת ברירת מחדל שנשלחת, אך היא מאפשרת למשתמש להזין הודעה מותאמת אישית שתישלח, במקום זאת. אם המשתמש משנה את ההודעה המותאמת אישית ל"אני ישן; תפסיק להציק לי" ולאחר מכן סוגר את האפליקציה, ההודעה עדיין צריכה להיות "אני ישן; תפסיק להציק לי", ולא ברירת המחדל המקורית, כאשר האפליקציה נפתחת מחדש. לפיכך, ההודעה המותאמת אישית חייבת להיות מאוחסנת במסד נתונים, ובכל פעם שהאפליקציה נפתחת, יש לאחזר את ההודעה הזו מהמסד חזרה לאפליקציה.

## אחסון נתונים מתמשכים TinyDB-ב

TinyWebDB. TinyDB- מספק שני רכיבים כדי להקל על פעילות מסד הנתונים: TinyDB. אתה משתמש ב- TinyDB כדי לאחסן נתונים קבועים ישירות במכשיר האנדרואיד; זה שימושי עבור אפליקציות אישיות שעבורן המשתמש לא יצטרך לשתף נתונים עם מכשיר או אדם אחר, כמו ב'אין לשלוח הודעות טקסט בזמן נהיגה'. מצד שני, אתה משתמש ב- TinyWebDB כדי לאחסן נתונים במסד נתונים אינטרנטי שניתן לשתף בין מכשירים. היכולת לגשת לנתונים ממסד נתונים אינטרנטי חיונית למשחקים ואפליקציות מרובי משתמשים שאיתם משתמשים יכולים להזין ולשתף מידע (כמו אפליקציית "MakeQuiz" בפרק 10).

רכיבי מסד הנתונים דומים, אבל TinyDB היא קצת יותר פשוטה, אז אנחנו נחקור את זה ראשון. עם TinyDB, אתה לא צריך להגדיר את מסד הנתונים בכלל; הנתונים מאוחסנים במסד נתונים ישירות במכשיר ומשויכים לאפליקציה שלך.

אתה מעביר נתונים לזיכרון לטווח ארוך עם בלוק `as TinyDB.StoreValue`, מוצג באיור 22-1, שמגיע מהאפליקציה ללא הודעות טקסט בזמן נהיגה.



איור 22-2. בלוק `TinyDB.StoreValue` מאחסן נתונים בזיכרון לטווח ארוך של המכשיר

סכימת תג-ערך משמשת לאחסון מסד נתונים. באיור 22-1, הנתונים מתוגים בטקסט "responseMessage". הערך הוא טקסט כלשהו שהמשתמש הקליד בתיבת טקסט עבור התגובה המותאמת אישית החדשה - משהו כמו "אני ישן; תפסיק להטריד אותי".

פרמטר התג נותן לנתונים שאתה מאחסן במסד הנתונים שם, דרך להתייחס למידע. הערך הוא הנתונים עצמם. אתה יכול לחשוב על התג כמפתח שבו תשתמש מאוחר יותר כשתרצה לאחזר את הנתונים ממסד הנתונים.

באופן דומה, אתה יכול לחשוב על מסד נתונים של TinyDB App Inventor כטבלה של ערך תג זוגות. לאחר ביצוע ה- `TinyDB1.StoreValue` באיור 22-1, למסד הנתונים של ההתקן יהיה הערך המופיע בטבלה 22-1.

טבלה 1.22-הערך המאוחסן במסדי הנתונים

תג	ערך
responseMessage	אני ישן; תפסיק להטריד אותי

אפליקציה עשויה לאחסן צמדי תג-ערך רבים עבור פריטי הנתונים השונים שברצונך שיהיו מתמידים. התג הוא תמיד טקסט, בעוד שהערך יכול להיות פיסת מידע בודדת (טקסט או מספר) או רשימה. לכל תג יש רק ערך אחד; בכל פעם שאתה מאחסן לתג, הוא מחליף את הערך הקיים.

## אחזור נתונים TinyDB-m

אתה מחזיר נתונים ממסד הנתונים באמצעות בלוק `TinyDB.GetValue`. כאשר אתה מתקשר `GetValue`, לאתה מבקש נתונים מסוימים על ידי מתן תג. עבור האפליקציה ללא הודעות טקסט בזמן נהיגה, אתה יכול לבקש את התגובה המותאמת אישית באמצעות אותו תג שבו השתמשת ב-`responseMessage`, `StoreValue` הקריאה ל-`GetValue` מחזירה את הנתונים, לכן עליך לחבר אותם למשתנה.

לעתים קרובות, תוכל לאחזר נתונים ממסד הנתונים כאשר האפליקציה נפתחת. `Inventor` `App` מספק מטפל מיוחד באירועים, `Screen.Initialize`, המופעל כאשר האפליקציה מופעלת. עליך להקפיד לשקול את המקרה כאשר עדיין אין נתונים במסד הנתונים (למשל, האפליקציה הראשונה שבה מופעלת). כאשר אתה קורא `GetValue` לאתה מציין פרמטר `valueIfTagNotThere` אין נתונים, הערך הזה יוחזר מהשיחה.

הבלוקים באיור 22-2, עבור המסך. אתחול של אי שליחת הודעות טקסט בזמן נהיגה `app`, מעידים על הדרך בה אפליקציות רבות טוענות נתוני מסד נתונים בעת האתחול. הבלוקים מכניסים את הנתונים שהוחזרו מ-`GetValue` לתווית `ResponseLabel`. אם יש נתונים כבר במסד הנתונים, הם ממוקמים ב-`ResponseLabel`. אין נתונים עבור התג הנתון, הערך `valueIfTagNotThere` "אני נוהג עכשיו, אני אשלח לך טקסט מאוחר יותר" במקרה זה, ממוקם ב-`ResponseLabel`.



איור 22-3: כאשר האפליקציה מופעלת, לעתים קרובות תוכל לאחזר מידע על מסד נתונים

## נתונים משותפים TinyWebDB-I

רכיב TinyDB מאחסן נתונים במסד נתונים הממוקם ישירות במכשיר האנדרואיד. זה מתאים לאפליקציות לשימוש אישי שאינן צריכות לשאת נתונים בין משתמשים. לדוגמה, אנשים רבים עשויים להתקין את האפליקציה ללא הודעות טקסט בזמן נהיגה, אך אין צורך שהאנשים השונים המשתמשים באפליקציה ישתפו את התגובות המותאמות אישית שלהם עם אחרים.

כמובן, אפליקציות רבות אכן חולקות נתונים: חשבו על פייסבוק, טוויטר ומשחקים מרובי משתמשים. עבור אפליקציות לשיתוף נתונים כאלה, מסד הנתונים חייב להימצא באינטרנט, לא במכשיר, כך שמשתמשי אפליקציה שונים יוכלו לתקשר איתו ולגשת למידע שלו.

TinyWebDB היא המקבילה האינטרנטית ל-TinyDB. עם זאת, אתה יכול לכתוב אפליקציות שמאחסנות נתונים באינטרנט, באמצעות פרוטוקול `StoreValue/GetValue` דומה לזה של TinyDB. כברירת מחדל, רכיב TinyWebDB מאחסן נתונים באמצעות מסד נתונים אינטרנטי שהוגדר על ידי צוות App Inventor ונגיש בכתובת <http://appinvtinywebdb.appspot.com>. האתר מספק גם ממשק אינטרנט קריא בנושאי שמנהל מסד נתונים (אתה) יכול להשתמש בו כדי לבחון את הנתונים המאוחסנים בו.

מסד נתונים ברירת מחדל זה מיועד לפיתוח בלבד; הוא מוגבל בגודלו ונגיש לכול מתכנני App Inventor. מכיוון שכל אפליקציית App Inventor יכולה לאחסן שם נתונים, אין לך ביטחון שאפליקציה אחרת לא תחליף את הנתונים שלך! אם אתה רק בוחן את App Inventor או בשלבים מוקדמים של פרויקט, מסד הנתונים האינטרנטי המוגדר כברירת מחדל הוא `tfne`. אם אתה יוצר אפליקציה לפריסה בעולם האמיתי, בשלב מסוים תצטרך להגדיר מסד נתונים אינטרנטי משלך. מכיוון שאנחנו רק חוקרים עכשיו, נשתמש במסד הנתונים המוגדר כברירת מחדל. בהמשך הפרק, תלמד כיצד ליצור מסד נתונים אינטרנטי משלך ולהגדיר את TinyWebDB להשתמש בו במקום זאת.

בסעיף זה, נבנה אפליקציית הצבעה (מתוארת באיור 22-3) כדי להמחיש כיצד TinyWebDB עובד. לאפליקציה יהיו התכונות הבאות:

- משתמשים מתבקשים להזין את כתובת האימייל שלהם בכל פעם שהאפליקציה נטענת. זה שם החשבון ישמש לתיגוף ההצבעה של המשתמש במסד הנתונים.

- משתמשים יכולים להגיש הצבעה חדשה בכל עת. במקרה זה, ההצבעה הישנה שלהם תהיה מוחלף.

- משתמשים יכולים לראות את ההצבעות של כל אחד בקבוצה.

- למען הפשטות, הנושא עליו מצביעים נקבע מחוץ לאפליקציה, כמו למשל במסגרת כיתתית בה המורה מכריזה על הנושא ומבקשת מכולם להצביע באופן אלקטרוני. (שים לב שניתן להרחיב את הדוגמה הזו כדי לאפשר למשתמשים להצביע על ידי פרסום נושאים להצבעה מתוך האפליקציה.)



**What to Eat? App**

Signed in as: joe@zmail.com

enter your vote

**Current Votes**

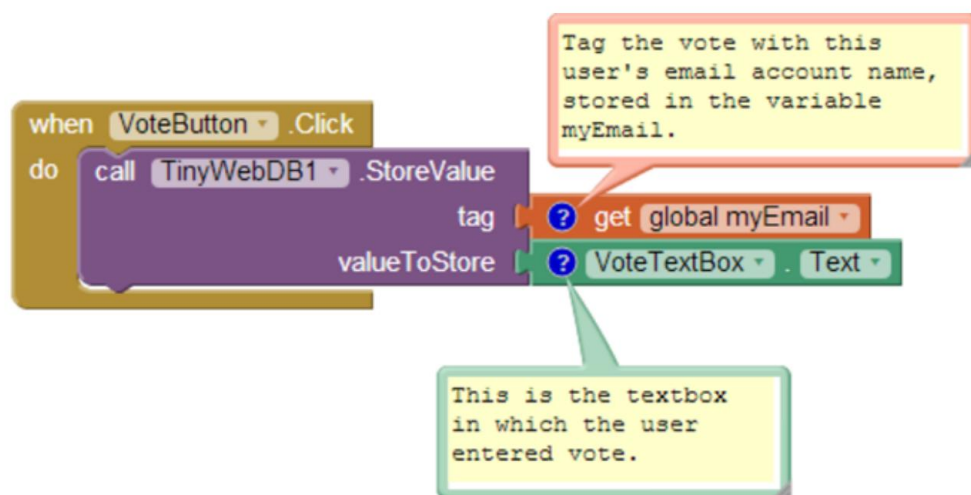
bill@zmail.com Hot Dogs  
joe@zmail.com Pizza

איור 22-4. אפליקציית הצבעה המאחסנת הצבעות ל-TinyWebDB

אחסון נתונים על ידי שימוש ב-TINYWEBDB

בלוק TinyWebDB.StoreValue פועל באותו אופן כמו TinyDB.StoreValue מלבד העובדה שהנתונים מאוחסנים באינטרנט. עבור מדגם ההצבעה שלנו, נניח שהמשתמש יכול להזין הצבעה בתיבת טקסט VoteTextBox ולהקיש על כפתור בשם VoteButton כדי לשלוח את ההצבעה. כדי לאחסן את ההצבעה במאגר האינטרנט כך שאחרים יכולו לראות את זה, נקודד את VoteButton.Click מטפל באירועים כמו הדוגמה באיור 22-4.

## פרק 338: עבודה עם מסדי נתונים



איור 22-5. כאשר המשתמש מזין הצבעה, היא מאוחסנת במסד הנתונים האינטרנטי

התג המשתמש לזיהוי הנתונים הוא האימייל של המשתמש, שנשמר בעבר במשתנה myEmail (תראה זאת מאוחר יותר). הערך הוא מה שהמשתמש הקליד. VoteTextBox-בלכן, אם האימייל של המשתמש היה joe@zmail.com וההצבעה שלו הייתה "פיצה", הערך היה מאוחסן במסד הנתונים כפי שמוצג בטבלה 22-2.

טבלה 22-2. התג והערך להצבעה נרשמים במסד הנתונים

ערך	תג
joe@zmail.com	פיצה

בלוק TinyWebDB.StoreValue שולח את צמד הערך-תג דרך האינטרנט לשרת מסד הנתונים בכתובת <http://appinvtinywebdb.appspot.com>. בזמן שאתה בודק את האפליקציה שלך, אתה יכול לעבור לכתובת האתר הזו, ללחוץ על getValue ולהזין תג שעבורו אחסנת ערך. האתר יציג לך את הערך הנוכחי של התג הזה.

בקשה ועיבוד נתונים עם TINYWEBDB

אחזור נתונים עם TinyWebDB מסובך יותר מאשר עם TinyDB. עם TinyDB, פעולת GetValue מחזירה מיד ערך מכיוון שהאפליקציה שלך מתקשרת עם מסד נתונים ישירות במכשיר האנדרואיד. עם TinyWebDB, האפליקציה מבקשת נתונים דרך האינטרנט, מה שעשוי לקחת זמן, ולכן אנדרואיד דורשת סכימה דו-שלבית לטיפול בה.

עם TinyWebDB, קריאה ל- GetValue מבקשת רק את הנתונים; זה באמת צריך להיקרא "RequestValue" מכיוון שהוא פשוט מעביר את הבקשה למסד הנתונים האינטרנטי ולא עושה זאת

למעשה לקבל ממנו ערך מיד. כדי לראות זאת בצורה ברורה יותר, בדוק את ההבדל בין בלוק TinyDB.GetValue לבלוק TinyWebDB.GetValue המוצג באיור 22-5.



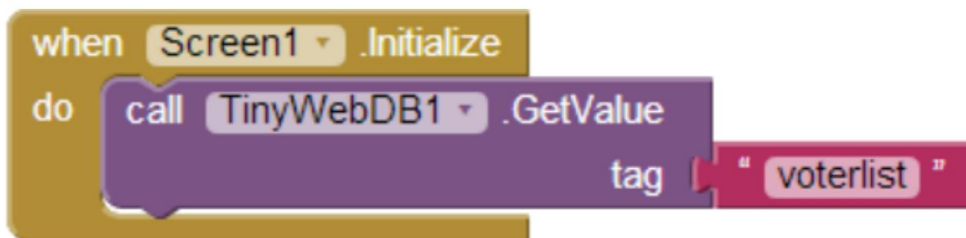
איור 22-6. החסימות TinyDB.GetValue ו-TinyWebDB.GetValue

בלוק TinyDB.GetValue מחזיר ערך מיד, וכך מופיע פלאג בצדו השמאלי כך שניתן להכניס את הערך המוחזר למשתנה או תכונה. הבלוק TinyWebDB.GetValue אינו מחזיר ערך באופן מיידי, כך שאין תקע בצד שמאל שלו.

במקום זאת, כאשר מסד הנתונים האינטרנטי ממלא את הבקשה והנתונים חוזרים ל- התקן, מופעל אירוע TinyWebDB.GetValue, תתקשר TinyWebDB.GetValue לבמקום אחד באפליקציה שלך, ולאחר מכן תתכנת את מטפל האירועים TinyWebDB.GetValue כדי לציין כיצד לטפל בנתונים כשהם מגיעים בפועל. מטפל באירועים כגון TinyWebDB.GetValue נקרא לפעמים הליך התקשרות חוזרת, מכיוון שישות חיצונית כלשהי (מסד הנתונים האינטרנטי) פועלת ומתקשרת לאפליקציה שלך בחזרה לאחר עיבוד בקשתך. זה דומה להזמנה בבית קפה עמוס: אתה מבצע את ההזמנה שלך ואז מחכה שהבריסטה יקרא בשמך כדי ללכת לאסוף את המשקה שלך. בינתיים היא גם קיבלה הזמנות מכל השאר בתור (וגם האנשים האלה מחכים שיקראו בשמותיהם).

#### GETVALUE-GOTVALUE בפעולה

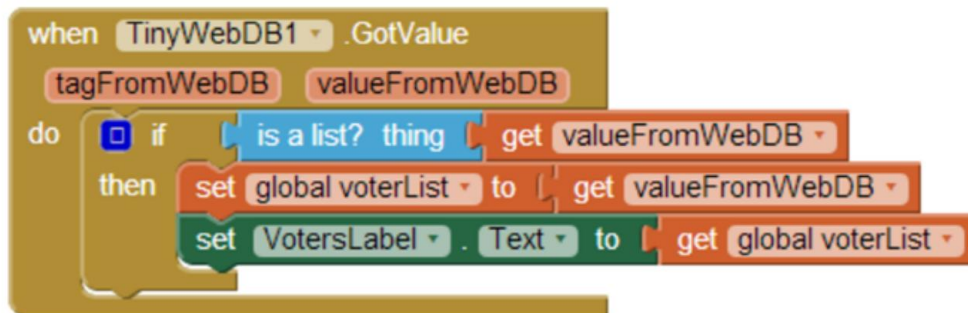
עבור האפליקציה לדוגמה שלנו, עלינו לאחסן ולאחזר רשימה של הבוחרים שיש להם את האפליקציה, מכיוון שהאפליקציה צריכה להציג את ההצבעות של כל המשתמשים. הסכימה הפשוטה ביותר לאחזר נתוני רשימה היא לבקש את הנתונים כאשר האפליקציה מופעלת, באירוע Screen.Initialize, כפי שמוצג באיור 22-6 (בדוגמה זו, פשוט נקרא למסד הנתונים עם התג של "רשימת בוחרים").



איור 22-7. בקשת נתונים באירוע Screen1.Initialize

## 340 פרק 22: עבודה עם מסדי נתונים

כאשר רשימת הבוחרים מגיעה ממסד הנתונים האינטרנטי, המטפל באירועים TinyWebDB1.GotValue מופעל. איור 22-7 מציג כמה בלוקים לעיבוד הרשימה המוחזרת.



איור 22-8. שימוש במטפל האירועים GotValue כדי לעבד את הרשימה שהוחזרה

הארגומנט valueFromWebDB של GotValue מחזיק את הנתונים המוחזרים מבקשת מסד הנתונים. לארגומנטים של אירועים כגון valueFromWebDB יש משמעות רק בתוך המטפל באירועים שמפעיל אותם. הם נחשבים מקומיים למטפל באירועים, מכיוון שאינך יכול להתייחס אליהם במטפלי אירועים אחרים.

מכיוון שארגומנטים כגון valueFromWebDB נגישים באופן גלובלי, אם אתה צריך את המידע בכל האפליקציה שלך, עליך להעביר אותו למשתנה גלובלי. בדוגמה, התפקיד העיקרי של GotValue הוא להעביר את הנתונים המוחזרים ב-valueFromWebDB למשתנה voterList, תשתמש במטפל אחר באירועים.

בלוק if במטפל האירועים משמש לעיתים קרובות גם יחד עם GotValue, הסיבה היא שמסד הנתונים מחזיר טקסט ריק ("") ב-valueFromWebDB אין נתונים עבור התג המבוקש. ערך ההחזרה הריק הזה מתרחש לרוב כאשר זו הפעם הראשונה שבה נעשה שימוש באפליקציה. אם תשאל אם ה-valueFromWebDB הוא רשימה, אתה מוודא שיש נתונים שהוחזרו בפועל. אם ה-valueFromWebDB הוא הטקסט הריק (מבחן ה-if הוא שקר), אתה לא מכניס אותו ל-voterList.

דוגמה מורכבת יותר של GETVALUE/GOTVALUE

הבלוקים באיור 22-7 הם מודל טוב לאחזור נתונים באפליקציה פשטנית למדי. עם זאת, בדוגמה שלנו להצבעה, אנחנו צריכים היגיון מסובך יותר. באופן ספציפי:

- האפליקציה אמורה לבקש מהמשתמש להקליד כתובת דואר אלקטרוני עם תחילת התוכנית. אנחנו יכולים להשתמש ברכיב Notifier בשביל זה, שמקפיץ חלון. (תוכל למצוא את ה-Notifier בלוח "ממשק משתמש" (Designer)-ב-  
כאשר המשתמש מקליד אימייל, נאחסן אותו במשתנה.

- רק לאחר קביעת האימייל של המשתמש יש להתקשר ל- GetValue כדי לאחזר את רשימת הבוחרים. אתה יכול להבין למה?

איור 22-8 מציג את הבלוקים עבור תכנית מסובכת יותר זו לבקשת נתוני מסד הנתונים.



איור 22-9 בסכימה מורכבת יותר זו, `GetValue` נקרא לאחר קבלת האימייל של המשתמש במקום `B-Screen.Initialize`.

עם ההפעלה (`Screen1.Initialize`), רכיב `Notifier` מבקש מהמשתמש לבצע הקלד כתובת אימייל. כאשר המשתמש עושה זאת, והמטפל באירוע `Notifier.AfterTextInput` מופעל, הערך מוכנס למשתנה ולתווית, ואז נקרא `GetValue` כדי לקבל את רשימת הבוחרים. שים לב ש- `GetValue` לא נקרא ישירות ב- `Screen.Initialize`, מכיוון שאנו צריכים שכתובת האימייל של המשתמש תוגדר תחילה.

לכן, עם החסימות הללו, כאשר האפליקציה מאתחלת, היא מבקשת מהמשתמש להקליד כתובת דוא"ל ואז קורא ל- `GetValue` עם תג "רשימת בוחרים". כאשר הרשימה מגיעה מהאינטרנט, `GotValue` מופעלת. הנה מה שצריך לקרות:

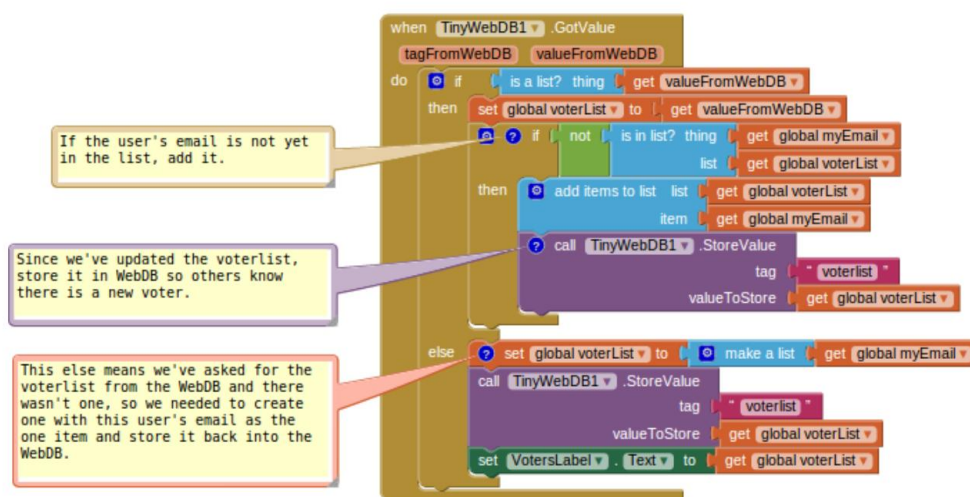
• צריכה לבדוק אם הנתונים שמגיעים אינם ריקים (מישהו השתמש באפליקציה ויזם את רשימת הבוחרים). אם יש נתונים (רשימת בוחרים), `GotValue` צריכה לבדוק אם כתובת האימייל של המשתמש המסוים שלנו כבר נמצאת ברשימת הבוחרים. אם לא, יש להוסיף אותו לרשימה, ואת הרשימה המעודכנת יש לאחסן בחזרה למסד הנתונים.

• אם עדיין אין רשימת בוחרים במאגר, עלינו ליצור אחת עם רשימת הבוחרים של המשתמש כתובת דואר אלקטרוני כפריט היחיד.

איור 22-9 מציג את הבלוקים להתנהגות זו.

## 342 פרק: 22 עבודה עם מסדי נתונים

הבלוקים שואלים תחילה אם רשימת בוחרים לא ריקה חזרה מהמאגר באמצעות קריאה היא רשימה? אם כן, הנתונים מוכנסים לרשימת הבוחרים המשתנה. זכור, לרשימת הבוחרים יהיו כתובות דוא"ל לכל מי שהשתמש באפליקציה הזו. עם זאת, אנחנו לא יודעים אם המשתמש הספציפי הזה עדיין ברשימה, אז אנחנו צריכים לבדוק. אם המשתמש עדיין לא ברשימה, כתובת הדוא"ל של המשתמש מתווספת עם הוסף פריט לרשימה, והרשימה המעודכנת נשמרת במסד הנתונים האינטרנטי.



איור 10-22 שימוש בלוקים של `GotValue` כדי לעבד את הנתונים המוחזרים ממסד הנתונים ולבצע פעולות שונות על סמך מה שמוחזר

ה- `else` של החסימה `if else` מופעל אם רשימה לא הוחזרה מהאינטרנט

מאגר מידע; זה קורה אם אף אחד לא השתמש באפליקציה עדיין. במקרה זה, נוצרת רשימת בוחרים חדשה עם כתובת האימייל של המשתמש הנוכחי בתור הפריט הראשון. הבוחר בסעיף אחד הזה הרשימה מאוחסנת לאחר מכן במסד הנתונים האינטרנטי (בתקווה שגם אחרים יצטרפו!).

## בקשת נתונים עם תגים שונים

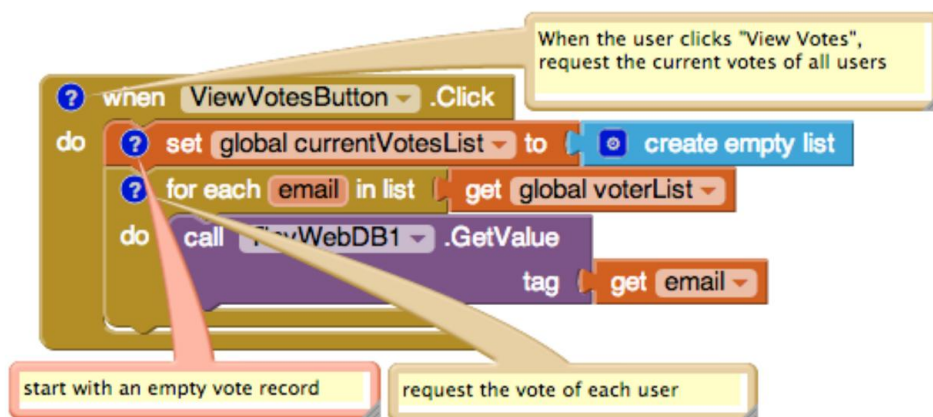
אפליקציית ההצבעה מנהלת עד כה רשימה של משתמשי אפליקציה. כל אדם יכול לראות את כתובות האימייל של כל שאר המשתמשים, אך עדיין לא יצרנו חסימות לאחזור והצגת הצבעה של כל משתמש.

נזכיר שאירוע `VoteButton.Click` הגיש הצבעה עם צמד תג-ערך של הטופס "אימייל: הצבעה". אם שני אנשים היו משתמשים באפליקציה ומצביעים, ערכי מסד הנתונים הרלוונטיים היו נראים בערך כמו טבלה 3-22

טבלה 22-3. צמד התג-ערך המאוחסנים במסד הנתונים

תג	ערך
	[bill@zmail.com, joe@zmail.com]
	bill@zmail.com
	joe@zmail.com

כאשר המשתמש לוחץ על כפתור `ViewVotes`, האפליקציה אמורה לאחזר את כל ההצבעות ממסד הנתונים ולהציג אותן. נניח שרשימת המצביעים כבר אוחרה ל-variant voterList; יכולים להשתמש ב-a עבור כל אחד כדי לבקש את ההצבעה של כל אדם ברשימה, כפי שמוצג באיור 22-10.



איור 22-11. שימוש ב-a עבור כל בלוק כדי לבקש את ההצבעה של כל אדם ברשימה

כאן אנו מאתחלים משתנה `currentVotesList` לרשימה ריקה, כי המטרה שלנו היא כדי להוסיף את ההצבעות המעודכנות מהמאגר לרשימה זו. לאחר מכן, אנו משתמשים עבור כל אחד להתקשר ל- `TinyWebDB1.GetValue` עבור כל כתובת דוא"ל ברשימה, שולחים את הפריט הנוכחי של עבור כל אחד, ששמו שונה ל"אימייל", כתגית הבקשה. שימו לב שההצבעות לא יתווספו למעשה ל- `currentVotesList` עד שיגיעו דרך סדרה של `GotValue` אירועים.

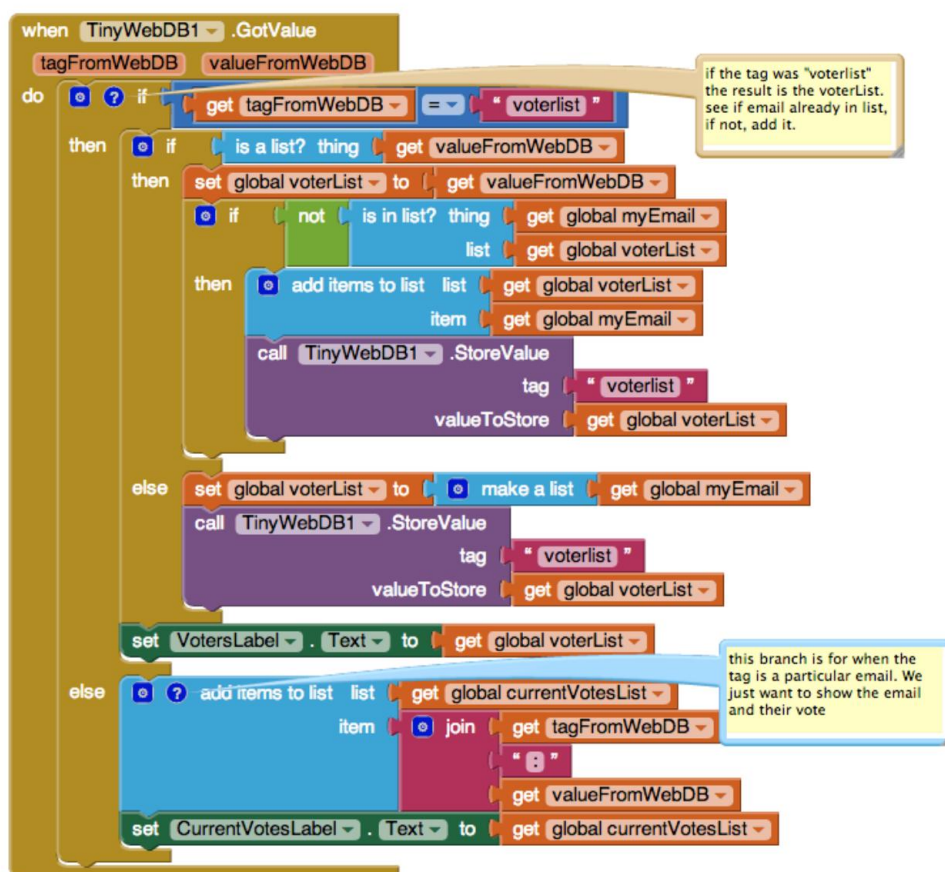
כעת, כשאנחנו רוצים להציג את ההצבעות באפליקציה שלנו, הדברים הופכים קצת יותר מסובכים שוב. עם הבקשות מ- `TinyWebDB.GetValue` תחזיר כעת נתונים הקשורים לכל תגי האימייל, כמו גם תג "רשימת הבוחרים" המשמש לאחזור רשימת כתובות האימייל של המשתמש. כאשר האפליקציה שלך מבקשת יותר מפריט אחד ממסד הנתונים עם תגים שונים, עליך לקודד את `TinyWebDB.GetValue` כדי לטפל בכל הבקשות האפשריות. (אתה עשוי לחשוב שאתה יכול לנסות לקודד מרובים

## פרק 22: עבודה עם מסדי נתונים

מטפלי אירועים של , `GotValue` אחד עבור כל בקשה למסד נתונים -האם אתה יכול להבין מדוע זה לא יעבוד?)

כדי לטפל במורכבות הזו, למטפל האירועים `GotValue` יש ארגומנט `tagFromWebDB` שמודיע לך איזו בקשה הגיעה זה עתה. במקרה זה, אם התג הוא "רשימת בוחרים", עלינו להמשיך לעבד את הבקשה כפי שעשינו בעבר. אם התג הוא משהו אחר, אנו יכולים להניח שזהו האימייל של מישהו ברשימת המשתמשים, הנובע מהבקשות שהופעלו במטפל האירועים . `ViewVotesButton.Click` כאשר בקשות אלו מגיעות, אנו רוצים להוסיף את הנתונים הנכנסים -הבוחר וההצבעה -לרשימת ההצבעות הנוכחית כדי שנוכל להציג אותם למשתמש.

איור 22-11 מציג את כל מטפלי האירועים . `TinyWebDB.GotValue`



איור 22-12 המטפל באירועים `TinyWebDB.GotValue`

## הקמת מסד נתונים אינטרנטי

כפי שהזכרנו קודם לכן בפרק, מסד הנתונים האינטרנטי המוגדר כברירת מחדל בכתובת <http://appinventivwebdb.appspot.com> מיועד למטרות אב טיפוס ובדיקות בלבד. לפני שאתה פורסם אפליקציה עם משתמשים אמיתיים, עליך ליצור מסד נתונים ספציפי עבור האפליקציה שלך.

אתה יכול ליצור מסד נתונים אינטרנטי באמצעות ההוראות בכתובת <http://appinventorapi.com/create-a-web-database-python-2-7>. (Wolber) ומכיל קוד לדוגמה והוראות להגדרת מסדי נתונים וממשקי API באינטרנט של Inventor. App ההוראות מכוונות אותך לאיזה קוד שתוכל להוריד ולהשתמש בו רק בשינוי מינורי של תצורה. הורדת הקוד זהה לזו המשמשת עבור מסד הנתונים האינטרנטי המוגדר כברירת מחדל שהוגדר על ידי App Inventor. App הוא פועל על App Engine של גוגל, שירות מחשוב ענן שיארח את מסד הנתונים האינטרנטי שלכם בשרתי גוגל בחינם (טוב, לפחות עד שהאתר יקבל מספר מסוים של כניסות). על ידי ביצוע ההוראות, תוכל להפעיל מסד נתונים אינטרנט פרטי משלך התואם לפרוטוקולים של Inventor App תוך דקות ולהתחיל ליצור אפליקציות סולריות התומכות באינטרנט המשתמשות בו.

כאשר אתה יוצר ופורסם מסד נתונים אינטרנט מותאם אישית משלך, הכלי App Engine מספק לך כתובת אתר שבה השרת שלך שוכן. אתה יכול לכוון את האפליקציה שלך להשתמש בשרת מסד הנתונים המותאם אישית שלך במקום ברירת המחדל, [appinventivwebdb.appspot.com](http://appinventivwebdb.appspot.com), על ידי שינוי המאפיין ServiceURL ברכיב TinyWebDB. לאחר שינוי המאפיין הזה, כל הקריאות ל- `TinyWebDB.StoreValue` ו- `TinyWebDB.GetValue` יתממשקו עם שירות האינטרנט החדש.

## סיכום

App Inventor מקל על אחסון נתונים בהתמדה באמצעות רכיבי TinyDB ו- TinyWebDB שלו. הנתונים מאוחסנים תמיד כזוג תג-ערך, כאשר התג מזהה את הנתונים לצורך אחזור מאוחר יותר. השתמש TinyDB בכאשר מתאים לאחסון נתונים ישירות במכשיר. כאשר יש צורך לשתף נתונים בין טלפונים (למשל, עבור משחק מרובה משתתפים או אפליקציית הצבעה), תצטרך להשתמש ב- TinyWebDB במקום זאת. TinyWebDB מסובך יותר מכיוון שאתה צריך להגדיר הליך התקשרות חזרה ( המטפל באירועים ) `GotValue` כמו גם שירות מסד נתונים אינטרנטי.

