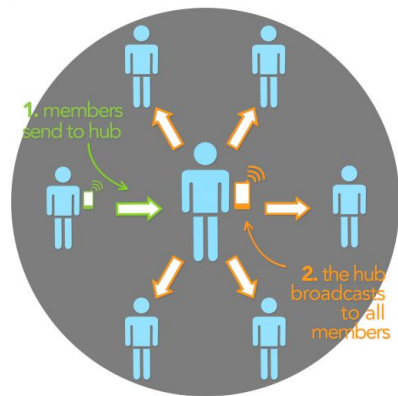


רכזת שידור

איור 11-1.



FrontlineSMS הוא כלי תוכנה המשמש ב

מדינות מתפתחות כדי לפקח על בחירות, לשדר שינויי מזג אוויר ולחבר אנשים שאין להם גישה לאינטרנט אבל יש להם

טלפונים וקישוריות סלולרית. זוהי פרי מוחו של קן בנקס, חלוץ בשימוש בטכנולוגיה ניידת כדי לעזור לאנשים במצוקה.

התוכנה משמשת כמרכז לטקסט SMS

תקשורת בתוך קבוצה. אנשים שולחים קוד מיוחד להצטרפות לקבוצה, ולאחר מכן הם מקבלים הודעות שידור מהרכזת. עבור מקומות ללא גישה לאינטרנט, רכזת השידור יכולה לשמש כחיבור חיוני לעולם החיצון.

בפרק זה, תיצור אפליקציית רכזת שידור הפועלת בדומה ל-FrontlineSMS- אבל פועל בטלפון אנדרואיד. עצם הרכזת במכשיר נייד פירושה שהמנהל יכול להיות בתנועה, דבר שחשוב במיוחד במצבים שנויים במחלוקת, כגון מעקב אחר בחירות ומשא ומתן על בריאות.

בפרק זה, תבנה אפליקציית רכזת שידור עבור ריקוד FlashMob המעשי צוות (FMDT) לקבוצה שמשמשת ברכזת כדי לארגן ריקודי אספסוף בכל מקום ובכל זמן. אנשים ירשמו לקבוצה באמצעות הודעת טקסט "joinFMDT" לרכזת, וכל מי שנרשם יכול לשדר הודעות לכל השאר בקבוצה. האפליקציה שלך תעבד הודעות טקסט שהתקבלו באופן הבא:

1. אם הודעת הטקסט נשלחה ממישהו שעדיין לא נמצא ברשימת השידורים, האפליקציה מגיבה בטקסט שמזמין את האדם להצטרף לרשימת השידורים.

2. אם מתקבלת הודעת הטקסט עם הקוד המיוחד, "joinFMDT" האפליקציה מוסיפה השולח לרשימת השידורים.

3. אם הודעת הטקסט נשלחת ממספר שכבר נמצא ברשימת השידורים, ההודעה משודרת לכל המספרים ברשימה.

האפליקציה הזו מסובכת יותר מהאפליקציה ללא טקסט בזמן נהיגה בפרק 4, אבל תבנה אותה פונקציונליות אחת בכל פעם, החל מהתגובה האוטומטית הראשונה

הודעה שמזמינה אנשים להצטרף. עד שתסיים את זה, יהיה לך מושג די טוב כיצד לכתוב אפליקציות תוך שימוש בטקסט SMS כממשק המשתמש. האם אתה רוצה לכתוב אפליקציית הצבעה לפי טקסט כמו אלה המשמשות בתוכניות כישרונות בטלוויזיה, או את אפליקציית ההודעות הקבוצתית הנהדרת הבאה? אתה תלמד איך כאן!

מה תלמד

המדריך מכסה את המושגים הבאים של ממציא האפליקציות, שסביר להניח שחלקם כבר מכירים:

- רכיב הטקסט לשליחת טקסטים ועיבוד טקסטים שהתקבלו.
- רשימת משתנים ונתונים דינמיים—במקרה זה, כדי לעקוב אחר רשימת הטלפון מספרים.
- עבור כל בלוק כדי לאפשר לאפליקציה לחזור על פעולות ברשימת נתונים. במקרה זה, תשתמש עבור כל אחד מהם כדי לשדר הודעות לרשימת מספרי הטלפון.
- רכיב TinyDB לאחסון נתונים באופן מתמשך. המשמעות היא שאם תסגור את האפליקציה ואז תפעיל אותה מחדש, רשימת מספרי הטלפון עדיין תהיה שם.

מתחילים

תזדקק לטלפון עם שירות SMS כדי לבדוק או להפעיל את האפליקציה הזו. תצטרך גם לגייס כמה חברים כדי לשלוח לך הודעות טקסט כדי לבדוק את האפליקציה במלואה. התחבר לאתר App Inventor והתחל פרויקט חדש. תן לזה שם "BroadcastHub" וגם הגדר את כותרת המסך "Broadcast Hub"-ללאחר מכן, חבר את המכשיר או האמולטור שלך לבדיקה חיה.

עיצוב הרכיבים

Broadcast Hub מאפשר תקשורת בין טלפונים ניידים. טלפונים אלה אינם צריכים להתקין את האפליקציה, או אפילו להיות סמארטפונים; הם יתקשרו באמצעות טקסט עם האפליקציה שלך. אז, במקרה זה, ממשק המשתמש של האפליקציה שלך מיועד רק למנהל הקבוצה.

ממשק המשתמש של המנהל הוא פשוט: הוא מציג את השידור הנוכחי רשימה; כלומר, רשימת מספרי הטלפון שנרשמו לשירות, וכל הטקסטים שהוא מקבל ומשדר.

כדי לבנות את הממשק, הוסף את הרכיבים המפורטים בטבלה 11-1.

טבלה 11-1. רכיבי ממשק משתמש עבור Broadcast Hub

מַטְרָה	סוג רכיב קבוצת צבעים איך תקרא לזה	
זוהי הכותרת "מספרי טלפון רשומים" למעלה את רשימת מספרי הטלפון.	תווית	תווית ממשק משתמש 1
ממשק משתמש BroadcastListLabelהצג את מספרי הטלפון הרשומים.	תווית	
זוהי הכותרת "יומן פעילות" מעל היומן מידע.	תווית	תווית ממשק משתמש 2
הצג יומן של הטקסטים שהתקבלו ושודרו.	תווית	ממשק משתמש LogLabel
עבדו את הטקסטים.	תווית	ממשק משתמש LogLabel
אחסן את רשימת מספרי הטלפון הרשומים.	תווית	ממשק משתמש TinyDB1

בעת הוספת הרכיבים, הגדר את המאפיינים הבאים:

1. הגדר את הרוחב של כל תווית ל"מלא הורה" כך שהיא משתרעת על פני הטלפון אופקית.

2. הגדר את גודל הגופן של תוויות הכותרת (Label1) ו- (Label2) ל-81 ובדוק את קופסאות FontBold.

3. הגדר את הגובה של BroadcastListLabel ו- LogLabel ל-002 פיקסלים. הם יראו שורות מרובות.

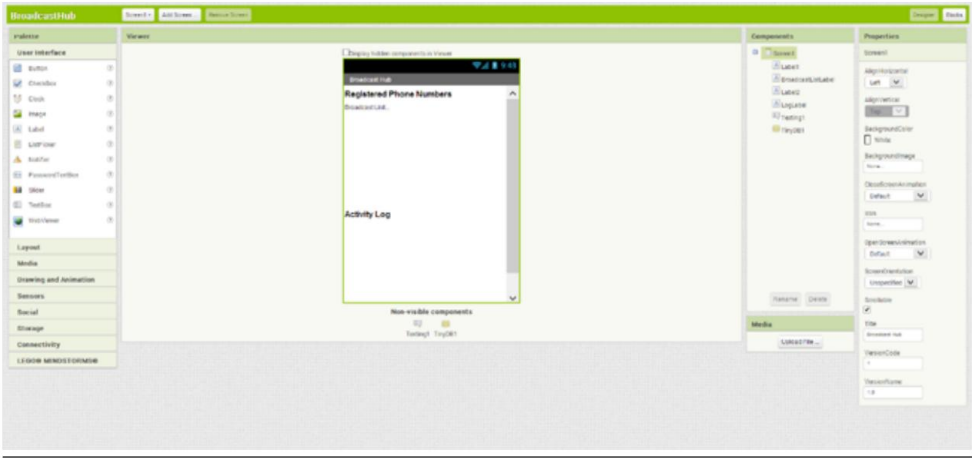
4. הגדר את המאפיין Text של "Broadcast List..." ל- BroadcastListLabel.

5. הגדר את המאפיין Text של LogLabel לריק .

6. הגדר את המאפיין Text של Label1 ל"מספרי טלפון רשומים".

7. הגדר את המאפיין Text של "Activity Log" ל- Label2.

איור 11-1 מציג את פריסת האפליקציה. Component Designer ב-



איור 11-2. Broadcast Hub ב-Components Designer

הוספת התנהגויות לרכיבים

הפעילות עבור Broadcast Hub אינה מופעלת על ידי המשתמש בהקלדת מידע או לחיצה על כפתור; במקום זאת, מדובר בטקסטים המגיעים מטלפונים אחרים. לעבד את הטקסטים הללו ולאחסן את מספרי הטלפון ששלחו אותם ברשימה, תזדקק לדברים הבאים: התנהגויות:

- כאשר הודעת הטקסט נשלחת ממשיהו שלא נמצא כבר ברשימת השידורים, האפליקציה מגיבה בטקסט שמזמין את השולח להצטרף.
- כאשר הודעת הטקסט "joinFMDT" מתקבלת, רשום את השולח במסגרת רשימת השידורים.
- כאשר הודעת הטקסט נשלחת ממספר שכבר נמצא ברשימת השידורים, ההודעה משודרת לכל המספרים ברשימה.

תגובה לטקסטים נכנסים

תתחיל ביצירת ההתנהגות הראשונה: כאשר אתה מקבל הודעת טקסט, שלח הודעה חזרה לשולחת המזמין אותה להירשם על ידי הודעת טקסט "joinFMDT" בחזרה אליך. אתה תעשה צריך את הבלוקים המפורטים בטבלה 11-2

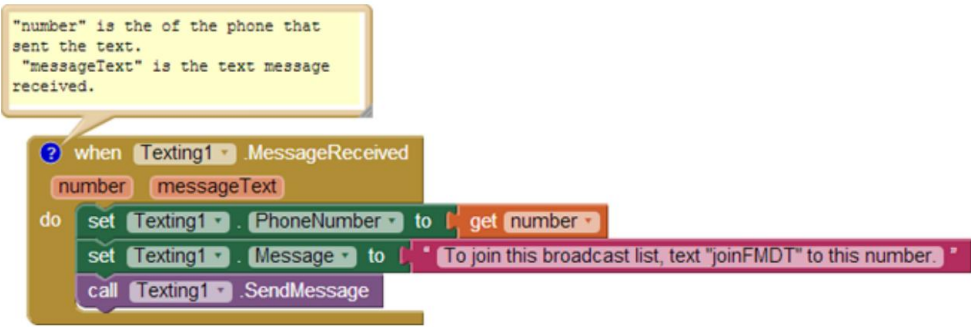
טבלה 11-2. חסימות להוספת הפונקציונליות להזמנת אנשים לקבוצה באמצעות טקסט

מטרה	מגרה	סוג בלוק
מופעל כאשר הטלפון מקבל הודעת טקסט.	שליחת הודעת טקסט	הודעת טקסט 1. התקבלה הודעה
הגדר את המספר עבור טקסט ההחזרה.	שליחת הודעת טקסט	הגדר את Texting1.PhoneNumber

מטרה	מגרה	סוג בלוק
גורם הטיעון של MessageReceived, אירוע שהתקבל בהודעה מספר הטלפון של השולח.		לקבל מספר
הגדר את הודעת ההזמנה לשליחה.	שליחת הודעת טקסט	הגדר SMS1.הודעה
הודעת ההזמנה.	טקסט	טקסט ("כדי להצטרף לרשימת השידורים הזו, שלח טקסט 'joinFMDT' למספר הזה")
שלח את זה!	שליחת הודעת טקסט	שליחת הודעות. 1.שלח הודעה

איך הבלוקים עובדים

אם השלמת את האפליקציה ללא הודעות טקסט בזמן נהיגה בפרק 4,הבלוקים האלה צריכים נראה מוכר. MessageReceived.SMS1מופעל כאשר הטלפון מקבל כל הודעת טקסט. איור 2-11מראה כיצד הבלוקים בתוך מטפל האירוע מגדירים את מספר טלפון והודעה של רכיב SMS1ולאחר מכן שלח את ההודעה.



איור 3-11שליחת הודעת ההזמנה חזרה לאחר קבלת הודעת טקסט



בדוק את האפליקציה שלך תצטרך טלפון שני כדי לבדוק התנהגות זו; אתה לא רוצה לשלוח הודעה לעצמך, כי זה יכול לנצח! אם אין לך טלפון אחר, אתה יכול להירשם בגוגל קול או שירות דומה ושלח הודעות SMSמאותו שירות לטלפון שלך. מהטלפון השני, שלח את הטקסט "שלום" אל הטלפון שמריץ את האפליקציה. הטלפון השני צריך אז לקבל טקסט שמזמין אותו להצטרף לקבוצה.

הוספת מספרים לרשימת השידורים

הגיע הזמן ליצור את החסימות להתנהגות השנייה: כאשר הודעת הטקסט "joinFMDT"מתקבל, השולח מתווסף לרשימת השידורים. ראשית, תצטרך הגדר משתנה רשימה, BroadcastList,כדי לאחסן את מספרי הטלפון שנרשמים. מ

את מגירת המשתנים, גרור החוצה בלוק גלובלי אתחול ותן לו שם "רשימת שידורים". אתחול אותו לרשימה ריקה באמצעות צור בלוק רשימה מתוך מגירת רשימות, כפי שמוצג באיור 11-3 (נוסיף את הפונקציונליות לבניית רשימה זו בקרוב).



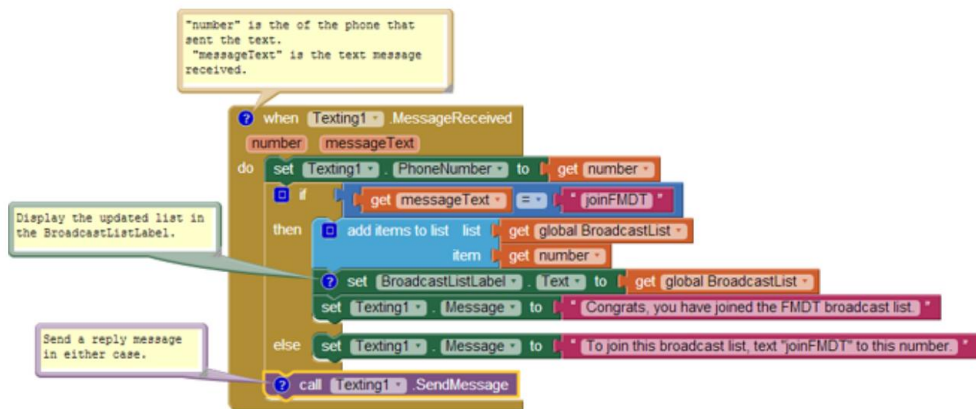
איור 11-4. המשתנה BroadcastList לאחסון רשימת המספרים הרשומים

לאחר מכן, שנה את המטפל באירוע Texting1.MessageReceived כך שיוסיף את מספר הטלפון של השולח ל- BroadcastList אם ההודעה שהתקבלה היא "joinFMDT." תזדקק לחסום אם אחרת כדי לבדוק את ההודעה, ולחסום הוספת פריט לרשימה הוסף את המספר החדש לרשימה. ערכת הבלוקים המלאה שתזדקק לה רשומה בטבלה 11-3. לאחר הוספת המספר לרשימה, הצג את הרשימה החדשה ב- BroadcastListLabel. טבלה 11-3. חסימות לבדיקת הודעת טקסט והוספת השולח לרשימת השידורים

מטרה	מגרה	סוג בלוק
בהתאם להודעה שהתקבלה, עשה אחרת דברים.	לשלוט	אחרת
קבע אם messageText שווה ל "הצטרף" FMDT.	מתמטיקה	=
גרור החוצה אירוע שהתקבל בהודעה חבר את זה לבלוק. = מטפל		קבל הודעה טקסט
חבר את זה לבלוק. =	טקסט	טקסט ("joinFMDT")
הוסף את מספר השולח BroadcastList-7	רשימות	הוסף פריטים לרשימה
הרשימה.	גרור החוצה מהמשתנה בלוק אתחול.	קבל את רשימת השידור העולמית
חבר את זה כפריט של הוסף פריטים לרשימה.	גרור החוצה אירוע שהתקבל בהודעה מטפל	לקבל מספר
הצג את הרשימה החדשה.	BroadcastListLabel	הגדר את BroadcastListLabel.7
גרור החוצה מהמשתנה BroadcastListLabel.7 טקסט לחסימה.		רשימת השידור העולמית
הכן הודעות טקסט כדי לשלוח הודעה בחזרה אל שולח.	שלוח הודעות טקסט	הגדר ל-SMS.1 הודעה ל
ברך את השולח על הצטרפותו לקבוצה.	טקסט	טקסט ("מזל טוב, אתה...")

איך הבלוקים עובדים

שורת הבלוקים הראשונה המוצגת באיור 4-1 מגדירה את Texting1.PhoneNumber למספר הטלפון של ההודעה שזוהי התקבלה; אנחנו יודעים שאנחנו הולכים להגיב לשולח, אז זה קובע את זה. האפליקציה שואלת אם ההודעה txeTהיה הקוד המיוחד, "joinFMDT" אם כן, מספר הטלפון של השולח מתווסף ל- BroadcastList, ונשלחת הודעת מזל טוב. אם ההודעה txeTהיא משהו אחר מאשר, "joinFMDT" הודעת התשובה חוזרת על הודעת ההזמנה. לאחר החסימה אם אחרת, נשלחת הודעת התשובה (השורה התחתונה של הבלוקים).



איור 5-11 אם ההודעה הנכנסת היא "joinFMDT" הוסף את השולח ל- BroadcastList



בדוק את האפליקציה שלך מטלפון שלא מריץ את האפליקציה, שלח את הודעת הטקסט "joinFMDT" לטלפון שמריץ את האפליקציה. אתה אמור לראות את מספר הטלפון הרשום בממשק המשתמש תחת "מספרי טלפון רשומים". הטלפון ה"שולח" אמור לקבל גם את הודעת הברכה כטקסט בתשובה. נסה לשלוח גם הודעה שאינה "joinFMDT", כדי לבדוק אם הודעת ההזמנה עדיין נשלחת כהלכה.

שידור ההודעות

לאחר מכן, תוסיף את ההתנהגות כך שהאפליקציה תשדר הודעות שהתקבלו למספרים ב- BroadcastList, אך רק אם ההודעה מגיעה ממספר שכבר מאוחסן ברשימה זו. המורכבות הנוספת הזו תדרוש יותר בלוקי בקרה, כולל עוד בלוקי בקרה ואחד עבור כל אחד מהם. תזדקק לחסימה נוספת אם אחרת כדי לבדוק אם המספר נמצא ברשימה, ולכל חסימה כדי לשדר את ההודעה לכל מספר ברשימה. תצטרך גם להעביר את הבלוקים 'אם אחרת' מהקודם

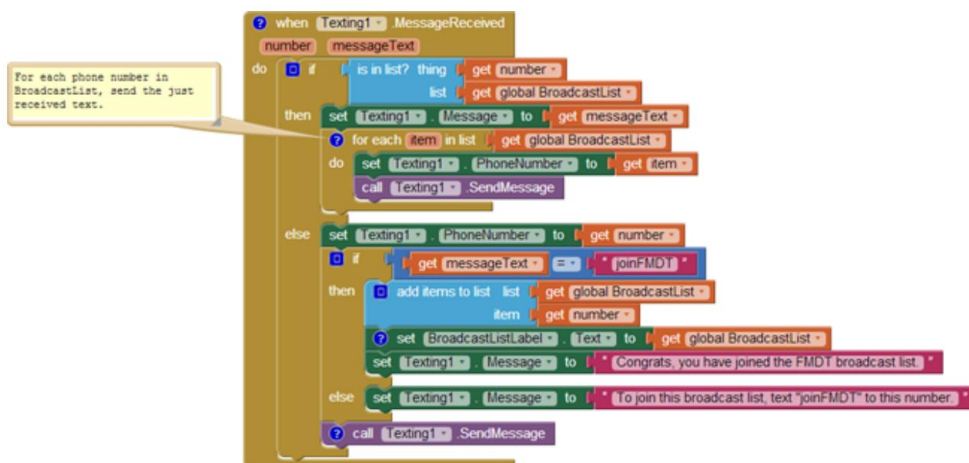
התנהגות ושקע אותם לחלק האחר של החדש אם כן. כל התוספת בלוקים שתזדקק להם מפורטים בטבלה 4-11

טבלה 4-11 חסימות לבדיקה אם השולח כבר בקבוצה

מטרה	מגרה	סוג בלוק
תלוי אם השולח כבר נמצא ברשימה, לעשות דברים שונים.	לשלוט	אחרת
בדוק אם יש משהו ברשימה.	רשימות	נמצא ברשימה?
לחבר את זה לשקע "רשימה" של הוא ברשימה?	גרור החוצה מהמשתנה בלוק אתחול	קבל את רשימת השידור העולמית
לחבר את זה לשקע "דבר" של הוא ברשימה?	גרור החוצה אירוע שהתקבל בהודעה מטפל	לקבל מספר
שלח שוב ושוב הודעה לכל החברים ברשימה.	לשלוט	לכל אחד
חבר את זה לשקע "רשימה" של כל אחד.	גרור החוצה מהמשתנה בלוק אתחול	קבל את רשימת השידור העולמית
הגדר את ההודעה.	שליחת הודעות טקסט 1	הגדר ל-SMS1. הודעה ל
ההודעה שהתקבלה ותהיה משדר.	גרור החוצה מה אירוע שהתקבל בהודעה	קבל הודעה טקסט
הגדר את מספר הטלפון.	שליחת הודעות טקסט 1	הגדר את Texting1.PhoneNumber ל
החזק את הפריט הנוכחי ברשימת השידורים; זה מספר (טלפון).	גרור החוצה עבור כל בלוק	לקבל פריט

איך הבלוקים עובדים

האפליקציה הפכה למורכבת מספיק כדי שהיא דורשת בלוק מקוון אם אחרת, אשר אתה יכול לראות באיור 5-11 בלוק מקוון אם אחר הוא אחד שמחובר ל- שקע של or else חילוק של אחר, חיצוני אם אחר. במקרה זה, החיצוני אם אחר סניף בודק אם מספר הטלפון של ההודעה שהתקבלה כבר נמצא ב- רשימה. אם כן, ההודעה מועברת לכל מי שברשימה. אם המספר אינו ברשימה, הבדיקה המקוננת מתבצעת: הבלוקים בודקים אם ההודעה txeT שווה ל "הצטרף FMDT" לוסעף אחת משתי דרכים בהתבסס על התשובה.



איור 6-11 החסימות בודקות אם השולח כבר בקבוצה ומשדרים את ההודעה אם כן

באופן כללי, אם ואם עוד בלוקים ניתן לקנן לרמות שרירותיות, מה שנותן לך את כוח לתכנת התנהגויות מורכבות יותר ויותר (ראה פרק 18 למידע נוסף על בלוקים מותנים).

ההודעה משודרת באמצעות `עבור כל אחד` (בתוך הסעיף החיצוני ואז). עבור כל אחד חוזר דרך ברשימת השידור ושולח את ההודעה לכל פריט. כאשר עבור כל חוזר, כל מספר טלפון עוקב מ- `BroadcastList` מאוחסן בפריט (פריט הוא מציין מיקום משתנה עבור הפריט הנוכחי המעובד בכל אחד). הבלוקים בתוך עבור כל קבוצה `Texting.PhoneNumber` לפרט הנוכחי ולאחר מכן לשלוח את ההודעה. למידע נוסף על אופן הפעולה של כל אחת מהן, ראה פרק 20.



בדוק את האפליקציה שלך ראשית, בקש שני טלפונים שונים להירשם על ידי הודעת טקסט "joinFMDT" לטלפון שמריץ את האפליקציה. לאחר מכן, הודעת טקסט נוספת מאחד הטלפונים. שני הטלפונים צריכים לקבל את הטקסט (כולל זה ששלח אותו).

מיפוי תצוגת הרשימה

האפליקציה יכולה כעת לשדר הודעות, אבל ממשיך המשתמש של מנהל האפליקציה צריך קצת עבודה. ראשית, רשימת מספרי הטלפון מוצגת בצורה לא אלגנטית. באופן ספציפי, כאשר אתה מציב משתנה רשימה בתווית, הוא מציג את הרשימה עם רווחים בין הפריטים, תוך התאמה ככל האפשר בכל שורה. אז, `BroadcastListLabel` עשוי להציג את `BroadcastList` כך:

(+1415111-1111 +1415222-2222 +1415333-3333 +1415444-4444)

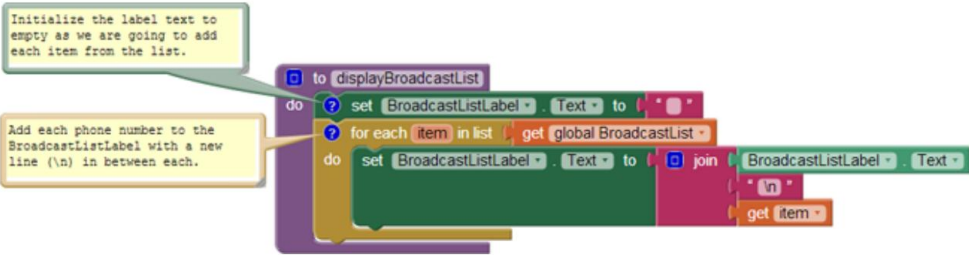
כדי לשפר עיצוב זה, צור הליך בשם `displayBroadcastList` באמצעות הבולקים המפורטים בטבלה 11-5. הליך זה מציג את הרשימה עם כל טלפון מספר בשורה נפרדת. הקפד לקרוא להליך מלמטה כדי להוסיף פריטים בלוק רשימה כך שהרשימה המעודכנת תוצג.

טבלה 11-5. חסימות לניקוי תצוגת מספרי הטלפון ברשימה שלך

מטרה	מגרה	סוג בלוק
צור את ההליך (אל תבחר לבצע את התוצאה).	להליך ("displayBroadcastList")	
הצג את הרשימה כאן.	הגדר את BroadcastListLabel.Text ל-BroadcastListLabel	
לחץ על טקסט ולאחר מכן לחץ על מחק כדי ליצור אובייקט טקסט ריק.	טקסט ("") טקסט	
חזור על המספרים.	לכל אחד לשלוט	
החבר את התוצאה לשקשוקה "in" של כל הבלוק. אתחול.	קבל את רשימת השידור העולמית	
שנה זאת עם כל אחד מהמספרים.	הגדר את BroadcastListLabel.Text ל-BroadcastListLabel	
בנה אובייקט טקסט ממספר חלקים.	טקסט	
הוסיף את זה לתווית בכל איטרציה של לכל אחד.	BroadcastListLabel	BroadcastListLabel.Text
הוסיף תו קו חדש כך הבא המספר נמצא בשורה הבאה.	טקסט	טקסט ("\\n")
המספר הנוכחי מהרשימה.	גרור החוצה עבור כל בלוק.	לקבל פריט

איך הבולקים עובדים

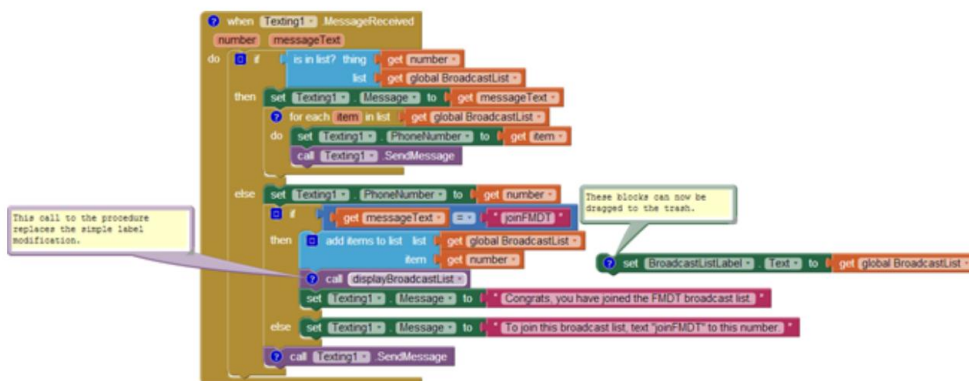
עבור כל אחד ב- `displayBroadcastList` מוסיף ברציפות מספר טלפון לסוף של התווית, כפי שמוצג באיור 11-6. הצבת תו חדש (\\n) בין כל פריט על מנת להציג כל מספר בשורה חדשה.



איור 11-7. הצגת מספרי הטלפון עם תו חדש בין כל אחד מהם

כמובן, הליך `displayBroadcastList` לא יעשה שום דבר אלא אם תקרא לזה. התקשר אליו במטפל האירועים, `Texting1.MessageReceived`, במטפל האירועים, כדי להוסיף פריט לרשימה. הקריאה צריכה להחליף את הבלוקים שפשוט מגדירים את `BroadcastListLabel.Text` ל- `BroadcastList`. אתה יכול למצוא את השיחה `displayBroadcastList` במגירה של נהלים.

איור 7-11 מראה כיצד החסימות הרלוונטיות בתוך ה- `Texting1.MessageReceived` מטפל באירוע צריך להסתכל.



איור 8-11 קריאה לנוהל `displayBroadcastList`

למידע נוסף על השימוש עבור כל אחד מהם כדי להציג רשימה, ראה פרק 20. למידע נוסף מידע על נהלי יצירה והתקשרות, ראה פרק 21.



בדוק את האפליקציה שלך הפעל מחדש את האפליקציה כדי לנקות את הרשימה ולאחר מכן תרשום לפחות שני טלפונים שונים (שוב). האם מספרי הטלפון מופיעים בקווים נפרדים?

רישום הטקסטים המשודרים

כאשר הודעת טקסט מתקבלת ומשודרת לטלפונים האחרים, האפליקציה צריכה לתעד את ההתרחשויות הזו כדי שמנהל המערכת יוכל לעקוב אחר הפעילות. במעצב הרכיבים הוספת את התווית `LogLabel` לממשק המשתמש למטרה זו. כעת, תקודד כמה בלוקים שמשנים את `LogLabel` בכל פעם שמגיע טקסט חדש.

אתה צריך לבנות טקסט שאומר משהו כמו "הודעה מ-+115141-2222 שודרה." המספר +1415111-2222 אינו נתונים קבועים; במקום זאת, זהו הערך של מספר הארגומנט שמגיע עם האירוע. `MessageReceived` כדי לבנות את הטקסט, תשרש את החלק הראשון, "הודעה מאת", עם בלוק קבל מספר ולבסוף עם החלק האחרון של ההודעה, הטקסט "שידור".

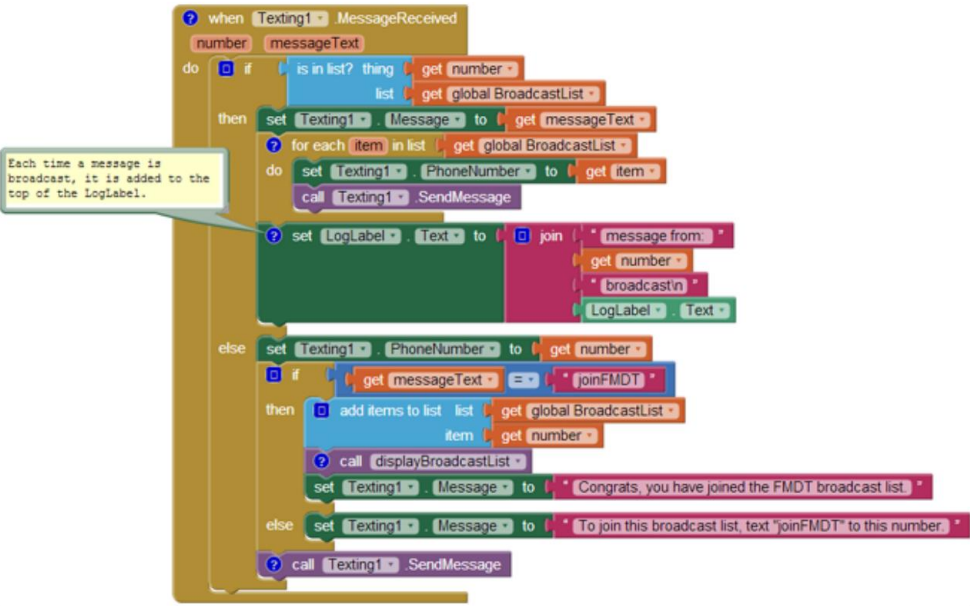
כפי ששנית בפרקים הקודמים, השתמש ב- `join` כדי לשרשר את החלקים באמצעות שימוש
הבלוקים המפורטים בטבלה 11-6.

טבלה 11-6. חסימות לבניית יומן ההודעות המשודרות שלך

מטרה	הבלוק	הגדר את
הצג את היומן כאן.	הגדר את	<code>LogLabel.Text</code>
בנה אובייקט טקסט ממספר חלקים.	הצג	<code>LogLabel</code>
זו הודעת הדיווח.	הודעה ("הודעה מאת")	
מספר הטלפון של השולח.	גרור החוצה אירוע שהתקבל בהודעה מטפל	לקבל מספר
הוסף את החלק האחרון של "הודעה משידור 111-2222" ולכלול קו חדש.	טקסט	טקסט ("שידור\ח")
הוסף יומן חדש לקודמים.	<code>LogLabel</code>	<code>LogLabel.Text</code>

איך הבלוקים עובדים

לאחר שידור ההודעה שהתקבלה לכל המספרים ב- `BroadcastList`, האפליקציה משנה כעת את `LogLabel` כדי להוסיף דוח של הטקסט ששודר זה עתה, כפי שמוצג באיור 11-8. שימו לב שההודעה מתווספת לתחילת הרשימה במקום הסוף. בדרך זו, ההודעה האחרונה שנשלחה לקבוצה מופיעה בראש.



איור 11-9. הוספת הודעת שידור חדשה ליומן

בלוק ההצטרפות יוצר ערכים חדשים של הטופס:

הודעה מ: 111-2222 שידור

בכל פעם שטקסט משודר, רשומת היומן מתווספת (נוסף מקדימה) ל-
`LogLabel.Text` כך שהערכים האחרונים יופיעו למעלה. האופן שבו אתה מארגן את בלוק
 ההצטרפות קובע את סדר הערכים. במקרה זה, ההודעה החדשה מתווספת עם שלושת השקעים
 העליונים של הצטרפות, ו- `LogLabel.Text` שמכיל את הערכים הקיימים - מחובר לשקע האחרון.

ה-"\n" בטקסט "broadcast\n" הוא תו השורה החדשה שגורם לכל יומן
 ערך להצגה בשורה נפרדת:

הודעה מ: 112222 שידור שידור מ: 555-6666 שידור

למידע נוסף על שימוש עבור כל אחד מהם כדי להציג רשימה, ראה פרק 20.

אחסון רשימת השידור במסד נתונים

האפליקציה שלך די עובדת, אבל אם השלמת חלק מהמדריכים הקודמים, בטח ניחשת שיש בעיה:
 אם המנהל יסגור את האפליקציה ויפעיל אותה מחדש, רשימת השידורים תאבד וכולם יצטרכו
 להירשם שוב.

כדי לתקן זאת, תשתמש ברכיב `TinyDB` כדי לאחסן ולאחזר את `BroadcastList` למסד נתונים
 וממנו.

אתה תשתמש בסכימה דומה לזו שבה השתמשת באפליקציית `MakeQuiz`
 (פרק: 10)

•אחסן את הרשימה במסד הנתונים בכל פעם שמתווסף פריט חדש.

•כאשר האפליקציה מופעלת, טען את הרשימה ממסד הנתונים למשתנה.

התחל בקידוד הבלוקים המפורטים בטבלה 7-11 כדי לאחסן את הרשימה במסד הנתונים. עם
 רכיב `TinyDB`, משמש לזיהוי הנתונים ולהבדיל ביניהם מנתונים אחרים המאוחסנים במסד
 הנתונים. במקרה זה, אתה יכול לתייג את הנתונים. "broadcastList" -כאתה תוסיף את החסימות
 באירוע `Texting1.MessageReceived` מתחת לבלוק הוסף פריטים לרשימה.

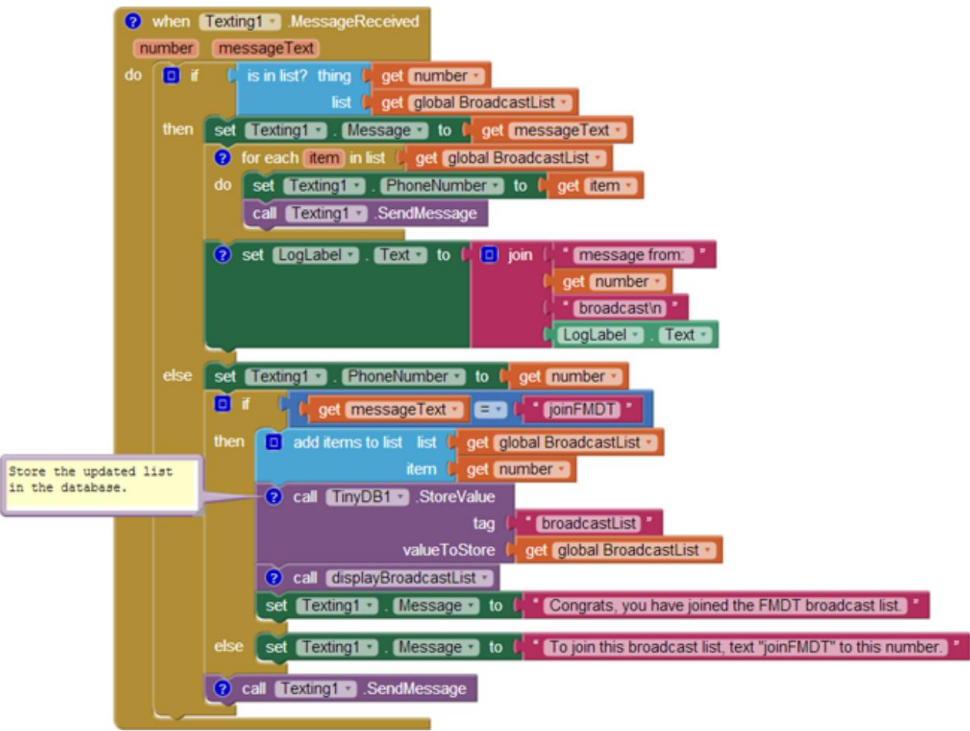
טבלה 7-11 חסימות לאחסון הרשימה עם `TinyDB`

מְטָרָה	מְגֵרָה	סוג בלוק
אחסן את הנתונים במסד הנתונים.	<code>TinyDB1</code>	<code>TinyDB1.StoreValue</code>
חבר את זה לחריץ "תג" של <code>StoreValue</code> .	טקסט	טקסט ("רשימת שידור")
חבר את זה לחריץ "ערך" של <code>StoreValue</code> .	גרור החוצה מבלוק אתחול משתנה	קבל את רשימת השידור העולמית

פרק 198: 11 רכות שידור

איך הבלוקים עובדים

כאשר נכנס טקסט "joinFMDT" ומספר הטלפון של החבר החדש מתווסף אליו הרשימה, TinyDB1.StoreValue נקראת כדי לאחסן את BroadcastList במסד הנתונים. ה תג (אובייקט טקסט בשם broadcastList) משמש כדי שתוכל מאוחר יותר לאחזר את נתונים. איור 9-11 ממחיש שהערך שנקרא על ידי StoreValue הוא המשתנה רשימת שידורים.



איור 10-11 קורא TinyDB-לכדי לאחסן את רשימת השידורים

טוען את רשימת השידור מבסיס נתונים

הוסף את הבלוקים המפורטים בטבלה 8-11 לטעינת הרשימה חזרה בכל פעם שהאפליקציה השקות.

טבלה. 8-11 חוסם לטעינת רשימת השידורים חזרה לאפליקציה כאשר היא מופעלת

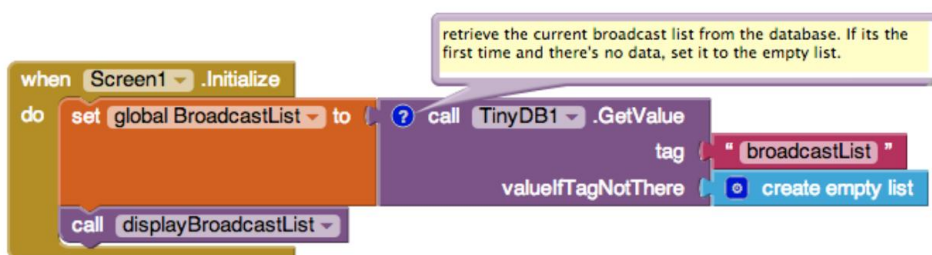
מטרה	מגרה	סוג בלוק
מופעל כאשר האפליקציה מופעלת.	מסך 1	מסך. 1. אתחול
בקש את הנתונים ממסד הנתונים.	TinyDB1	TinyDB1.GetValue
חבר את זה לשקע "תג" של GetValue.	טקסט	טקסט ("רשימת שידור")

מגרה	סטנדרט
call	displayBroadcastList
ללאחר טעינת הנתונים, הצג אותם.	

כאשר האפליקציה מתחילה, אירוע Screen1.Initialize מופעל, כך שאתה חוסם יכנס למטפל באירוע הזה.

איך הבלוקים עובדים

כאשר האפליקציה מתחילה, האירוע Screen1.Initialize מופעל. הבלוקים המוצגים באיור 11-10 מבקשים את הנתונים ממסד הנתונים עם TinyDB1.GetValue.



איור 11-10 טעינת רשימת השידורים ממסד הנתונים

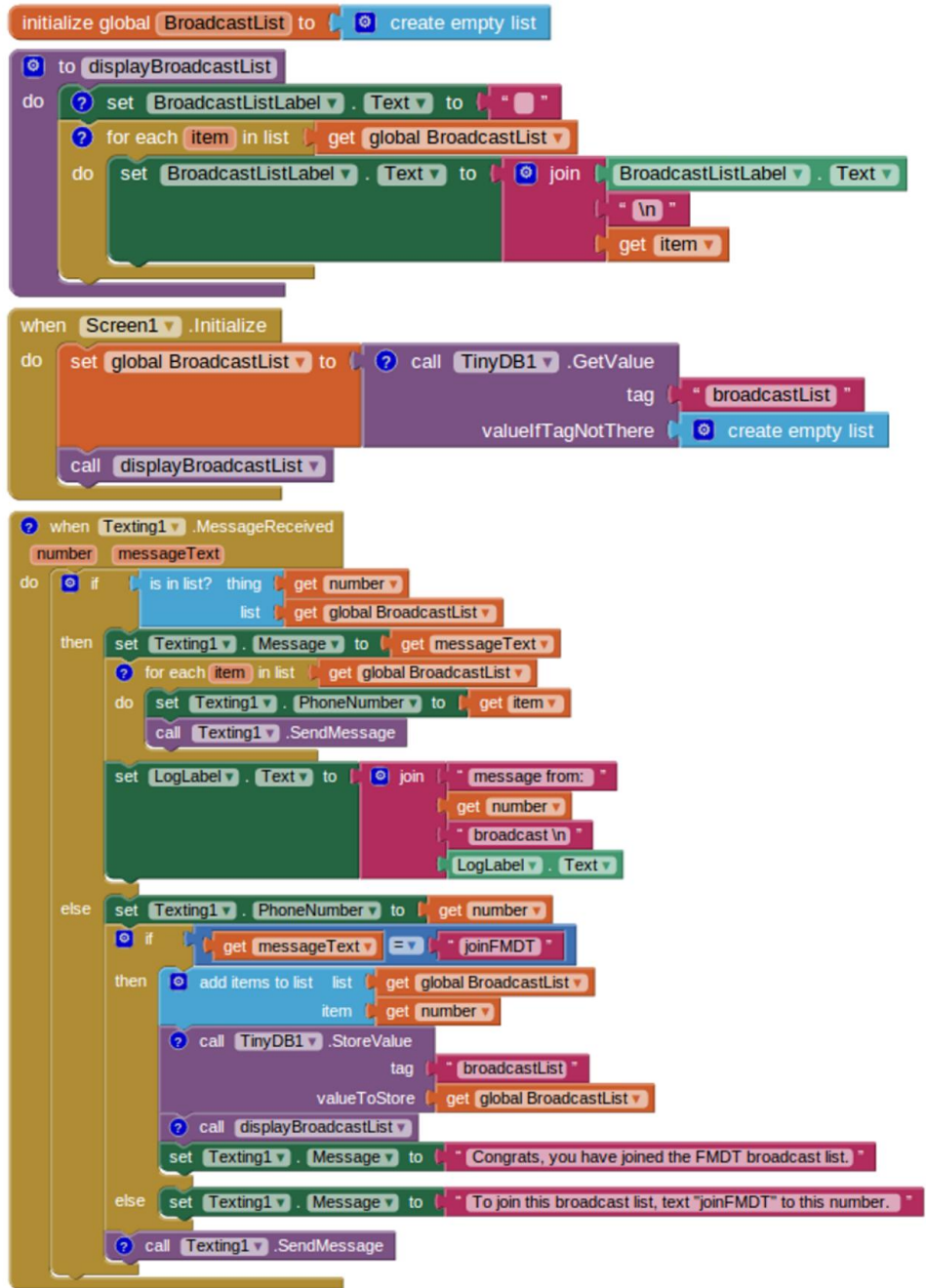
אתה קורא TinyDB.GetValue לבאמצעות אותו תג שבו השתמשת לאחסון הרשימה (broadcastList). במקרה הכללי, רשימת מספרי הטלפון המאוחסנת בעבר תוחזר ותמוקם במשתנה BroadcastList. אבל TinyDB.GetValue מספק שקע, valueIfTagNotThere, לציון מה הבלוק צריך להחזיר אם עדיין אין נתונים במסד הנתונים של התג הזה, כפי שיקרה בפעם הראשונה שהאפליקציה הזו תופעל. במקרה זה, מוחזרת רשימה ריקה.



בדוק את האפליקציה שלך אתה יכול להשתמש בבדיקה חיה עבור אפליקציות שמשנות את מסד הנתונים, אך בצע זאת בזריזות. במקרה זה, הוסף טקסט לאפליקציה באמצעות טלפון אחר כדי להוסיף מספרים לרשימת השידורים ולאחר מכן הפעל מחדש את האפליקציה. אתה יכול להפעיל מחדש במצב בדיקה חי על ידי מעבר למעצב ושינוי מאפיין כלשהו, אפילו משהו כמו שינוי הגופן של תווית. שים לב שכדי לבדוק אפליקציות מסד נתונים במלואן, עליך לארוז ולהוריד את האפליקציה לטלפון (בחר "בנה" >אפליקציה (שמור apk למחשב שלי)). לאחר שהורדת את האפליקציה שלך, השתמש בטלפונים האחרים שלך כדי לשלוח הודעת טקסט כדי להצטרף לקבוצה ולאחר מכן לסגור את האפליקציה. אם המספרים עדיין מופיעים בעת הפעלת האפליקציה מחדש, חלק מסד הנתונים פועל.

האפליקציה השלמה: Broadcast Hub

איור 11-11 ממחיש את הבלוקים באפליקציית Broadcast Hub שהושלמה.



איור 11-12. האפליקציית Broadcast Hub המלאה

וריאציות

לאחר שחגגת את בניית אפליקציה כה מורכבת, אולי תרצה לחקור כמה מהווריאציות הבאות:

- האפליקציה משדרת כל הודעה לכולם, כולל הטלפון ששלח את ההודעה. שנה זאת כך שההודעה תשודר לכולם מלבד השולח.

- אפשר לטלפונים של לקוחות להסיר את עצמם מהרשימה על ידי הודעת טקסט "צא" ל-אפליקציה. תצטרך להסיר מהרשימה בלוק.

- תן למנהל הרכזת את היכולת להוסיף ולהסיר מספרים מה-רשימת שידורים דרך ממשק המשתמש.

- תן למנהל הרכזת לציין מספרים שאסור להכניס ל-
רשימה.

- התאם אישית את האפליקציה כך שכל אחד יוכל להצטרף כדי לקבל הודעות, אבל רק את מנהל יכול לשדר הודעות.

- התאם אישית את האפליקציה כך שכל אחד יוכל להצטרף כדי לקבל הודעות, אבל רק רשימה קבועה של מספרי טלפון יכולה לשדר הודעות לקבוצה.

סיכום

להלן כמה מהמושגים שכיסינו במדריך זה:

- אפליקציות יכולות להגיב לאירועים שלא יוזמו על ידי משתמש האפליקציה, כגון טקסט מתקבל. זה אומר שאתה יכול לבנות אפליקציות שבהן "המשתמשים" שלך נמצאים בטלפון אחר.

- ניתן להשתמש ב- `Nested if else` ולכלל בלוקים לקוד התנהגויות מורכבות. למידע נוסף על תנאים ולכלל איטרציה, ראה פרק 18 ופרק 20 בהתאמה.

- ניתן להשתמש בגוש החיבור כדי לבנות אובייקט טקסט ממספר חלקים.

- ניתן להשתמש ב- `TinyDB` כדי לאחסן ולאחזר נתונים ממסד נתונים. גנרל התוכנית היא לקרוא ל- `StoreValue` כדי לעדכן את מסד הנתונים בכל פעם שהנתונים משתנים ולהתקשר ל- `GetValue` כדי לאחזר את נתוני מסד הנתונים כשהאפליקציה מתחילה.