

אמזון בחנות הספרים

איור 13-1

נניח שאתה מעיין בספרים בחנות הספרים האהובה עליך ורוצה לדעת כמה א

עלויות הזמנה באתר Amazon.com עם האמזון ב

אפליקציית חנות הספרים, אתה יכול לסרוק ספר או להזין ISBN, והאפליקציה תגיד לך את המחיר הנמוך ביותר הנוכחי של הספר באתר Amazon.com. אתה יכול גם לחפש ספרים בנושא מסוים.

אמזון בחנות הספרים מדגים איך



ממשק המשתמש - הממשק ליצור אפליקציות שמדברות לשירותי אינטרנט (המכונה ממשקי תכנות יישומים או ממשקי API). אפליקציה זו תקבל נתונים משירותי אינטרנט שנוצרו על ידי אחד ממחברי הספר הזה. עד סוף פרק זה, תוכל ליצור אפליקציה מותאמת אישית משלך לשיחה עם אמזון. לאפליקציה יש ממשק משתמש פשוט שבעזרתו המשתמש יכול להזין מילות מפתח או ISBN של ספר (מספר ספר סטנדרטי בינלאומי - קוד בן 10 או 13 ספרות המזהה ספר באופן ייחודי) ולאחר מכן מפרט את הכותרת, ISBN והמחיר הנמוך ביותר עבור עותק חדש באמזון. הוא גם משתמש ברכיב BarcodeScanner כך שהמשתמש יכול לסרוק ספר כדי להפעיל חיפוש במקום להזין טקסט (טכנית, הסורק פשוט מזין עבורך את ISBN-השל הספר).

מה תלמד

באפליקציה זו (מוצגת באיור, (13-1 תלמדו:

• כיצד להשתמש בסורק ברקוד בתוך

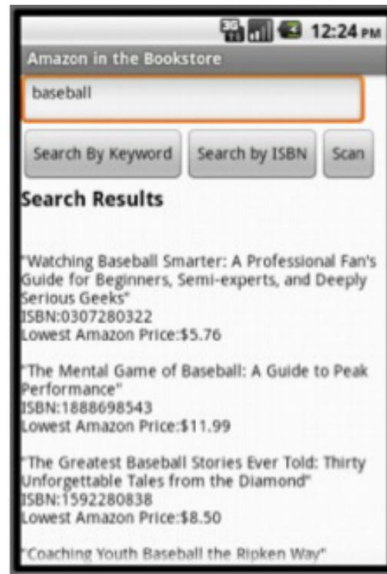
אפליקציה.

• כיצד לגשת למקור מידע אינטרנטי

TinyWebDB (של אמזון) דרך ה-API

רכיב.

• כיצד לעבד נתונים מורכבים המוחזרים מאותו מקור מידע
אינטרנטי. בפרט, תלמד כיצד לעבד רשימה של ספרים שבה
כל ספר הוא עצמו רשימה של שלושה פריטים (כותרת, מחיר
ו-ISBN).



איור 13-2. אמזון בחנות הספרים
פועלת באמולטור

תכיר גם את קוד המקור שבו תוכל להשתמש כדי ליצור אינטרנט משלך
שירות API עם שפת התכנות Python I-App Engine של גוגל.

מהו API?

לפני שתתחיל לעצב את הרכיבים שלך ולתכנת את האפליקציה, בואו נסתכל מקרוב על מה זה ממשק מתכנת יישומים (API) וכיצד הוא עובד. API הוא כמו אתר אינטרנט, אבל במקום לתקשר עם בני אדם, הוא מתקשר עם תוכנות מחשב אחרות. ממשקי API נקראים לעתים קרובות תוכניות "שרת" מכיוון שהם בדרך כלל מגישים מידע לתוכניות "לקוח" שמתממשקות למעשה עם בני אדם - כמו אפליקציית App Inventor. גם השתמשת באפליקציית Facebook בטלפון שלך, אתה משתמש בתוכנת לקוח שמתקשרת עם אפליקציית שרת API-השל Facebook.

בפרק זה, תיצור אפליקציית לקוח אנדרואיד המתקשרת עם ממשק API של אמזון. האפליקציה שלך תבקש מידע על ספר Amazon API מ-ISBN ו-IPA יחזיר רישומים מעודכנים לאפליקציה שלך. לאחר מכן האפליקציה תציג את נתוני הספר למשתמש.

API-השל Amazon שתשתמש בו מוגדר במיוחד לשימוש עם App Inventor. לא יכנס כאן לפרטים העגומים, אבל כדאי לדעת שכתוצאה מהקונפוגרציה הזו, אתה יכול להשתמש ברכיב TinyWebDB כדי לתקשר עם אמזון. החדשות הטובות הן שאתה כבר יודע איך לעשות את זה! אתה תתקשר ל- `TinyWebDB.GetValue` כדי לבקש מידע ולאחר מכן תעבד את המידע המוחזר במטפל האירועים, `TinyWebDB.GotValue`, בדיוק כפי שאתה עושה כשאתה משתמש במסד נתונים אינטרנטי. (תוכל לחזור ליישומי `MakeQuiz` ו-`TakeQuiz` בפרק 10 כדי לרענן את הזיכרון שלך, במידת הצורך).

לפני יצירת האפליקציה, תצטרך להבין את הפרוטוקול של Amazon API, המפרט את הפורמט עבור הבקשה שלך ואת הפורמט של הנתונים המוחזרים. בדיוק כפי שלתרבויות אנושיות שונות יש פרוטוקולים שונים (כשאת פוגש מישהו, האם אתה לוחץ ידיים, מרכין או מהנהן?), גם למחשבים שמדברים זה עם זה יש פרוטוקולים. API-השל Amazon שבו תשתמש כאן מספק ממשק אינטרנט לבחינת אופן הפעולה של API-הלפני שתתחיל להשתמש בו. למרות שה-IPA נועד לדבר עם מחשבים אחרים, ממשק אינטרנט זה מאפשר לך לראות איך התקשורת הזו תתרחש. בעקבות השלבים הללו, תוכלו לנסות אילו קריאות `GetValue` מסוימות יחזרו דרך האתר, ולדעת שממשק API-היתנהג בדיוק אותו הדבר כשתבקשו ממנו נתונים דרך רכיב App Inventor ב-TinyWebDB ובואו נתחיל:

1. פתח דפדפן ועבור אל <http://aiamazonapi.appspot.com/> אתה תראה את

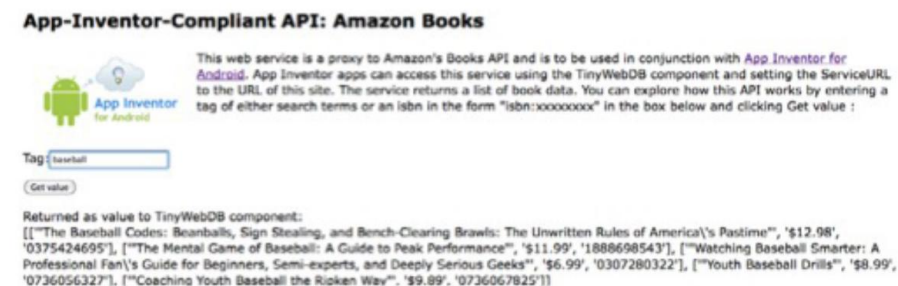
אתר אינטרנט המוצג באיור 13-2.

222 פרק: 13אמזון בחנות הספרים



אזור 13-3 מממשק האינטרנט של App Inventor Amazon API

2. בדף אינטרנט זה, אתה יכול לנסות את הפונקציה האחת שאתה יכול לקרוא עם API-הזהה: `getValue`. (למשל, "בייסבול") בשדה התגים ולאחר מכן לחץ על "קבל ערך".
דף האינטרנט יציג רשימה של חמשת הספרים המובילים שהוחזרו מאמזון, כפי שמוצג באזור 13-3.



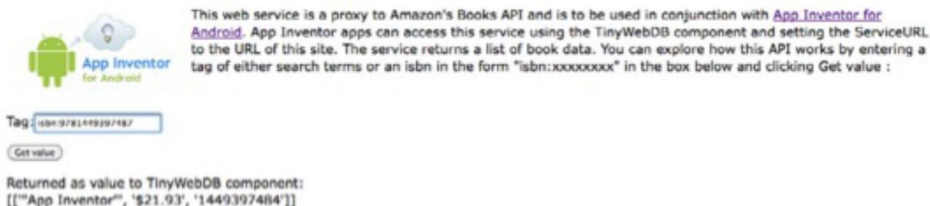
אזור 13-4 קריאה ל-API nozamA כדי לחפש ספרים הקשורים לתג (או מילת המפתח) "בייסבול"

הערך המוחזר הוא רשימה של ספרים, כל אחד מוקף בסוגריים [ככה] ומספק את הכותרת, העלות וה-NBSI.
אם תסתכל מקרוב, תראה שכל ספר מיוצג למעשה כרשימה משנה של רשימה ראשית אחרת. הרשימה הראשית (על בייסבול) מוקפת בסוגריים, וכל תת-רשימה (או ספר) מוקפת בסוגריים משלו בתוך הסוגריים הראשיים. אז, ערך ההחזר API-מזה הוא למעשה רשימה של רשימות, כאשר כל תת-רשימה מספקת את המידע עבור ספר אחד. בואו נסתכל על זה קצת יותר מקרוב. כל סוגר שמאלי ([]) בנתונים מצוין את תחילתה של רשימה. הסוגר השמאלי הראשון של התוצאה מצוין את תחילת הרשימה החיצונית (רשימת הספרים). מימין לה התחלה של רשימת המשנה הראשונה, הספר הראשון, כפי שמוצג כאן:

["קודי הבייבול: כדורי כדורים, גניבת שלטים וקטטות ספסל:
 החוקים הבלתי כתובים של הבילוי של אמריקה, "['0375424695', '\$12.98'
 לרשימה המשנה שלושה חלקים: שם, המחיר הנוכחי הנמוך ביותר עבור הספר באמזון
 ISBN-ושל הספר. כאשר אתה מקבל את המידע הזה לאפליקציית App Inventor שלך, תוכל
 לגשת לכל חלק באמצעות פריט רשימה נבחר, עם אינדקס 1 עבור הכותרת, אינדקס 2 עבור
 המחיר ואינדקס 3 עבור ISBN. (כדי לרענן את הזיכרון בעבודה עם אינדקס ורשימות, בקר שוב
 באפליקציית MakeQuiz בפרק 10.)

3. במקום לחפש לפי מילת מפתח, אתה יכול לחפש ספר על ידי הזנת ISBN. כדאי
 לבצע חיפוש כזה, אתה מזין תג בצורת "isbn: 00000000000", כאשר רשימת ה-0-
 מייצגת מספר ISBN בפועל (ראה איור 13-4). (הסוגריים הכפולים ({})) בתוצאה
 [][["App Inventor", "\$21.93", "1449397484"]]] מציגים רשימה של רשימות
 עדיין מוחזרת, למרות שיש רק ספר אחד. זה אולי נראה קצת מוזר עכשיו, אבל זה
 יהיה חשוב כשאנחנו ניגשים למידע עבור האפליקציה שלנו.

App-Inventor-Compliant API: Amazon Books



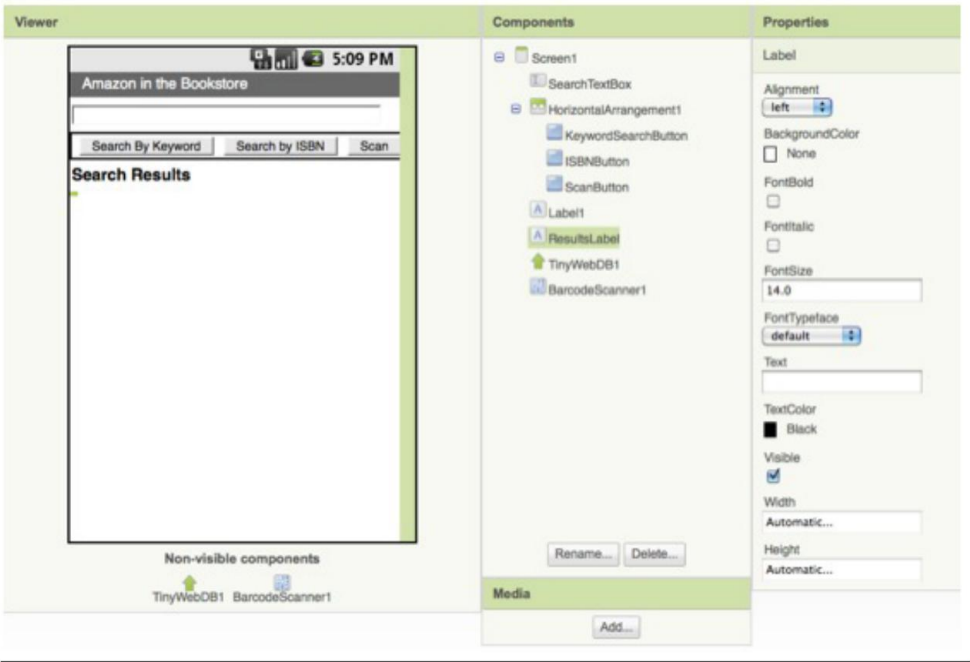
איור 13-5. שאילתה של Amazon API באמצעות ISBN במקום מילת מפתח

מתחילים

התחבר לאתר App Inventor והתחל פרויקט חדש. אז תן לזה "AmazonBooks", והגדר את כותרת
 המסך ל-"nozaMA at the Bookstore". לאחר מכן, חבר את המכשיר או האמולטור שלך לבדיקה
 חיה.

עיצוב הרכיבים

ממשק המשתמש של אפליקציית הספרים של אמזון פשוט יחסית: תן לה Textbox להזנת מילות
 מפתח או ISBN, שני כפתורים לתחילת שני סוגי החיפושים (מילת מפתח או ISBN) וכפתור שלישי
 לאפשר למשתמש לסרוק ספר (אנחנו נגיע לזה עוד מעט). לאחר מכן, הוסף תווית כותרת ותווית
 נוספת לרשימת התוצאות שה- nozaMA API מחזיר, ולבסוף שני רכיבים שאינם גלויים: TinyWebDB
 ו- BarcodeScanner בדוק את התוצאות שלך מול איור 13-5.



איור 13-6.ממשק המשתמש של אמזון בחנות הספרים המוצג ב-Designer

טבלה 13-1 מפרטת את כל הרכיבים שתצטרך לבניית ממשק המשתמש המוצג באיור 13-5

טבלה 13-1.רשימת רכיבים עבור אמזון באפליקציית חנות הספרים

מטרה	קבוצת צבעים איך תקרא לזה	סוג רכיב
המשתמש מוזן מילות מפתח או ISBN כאן.	ממשק משתמש SearchTextBox	תיבת טקסט
פונקציה שמאפשרת למשתמש לחפש את הכתובים בשורה.		
ממשק המשתמש המאפשר למשתמש לחפש מילות מפתח		לחצן
ממשק המשתמש המאפשר למשתמש לחפש ISBN		לחצן
לחצן שמאפשר למשתמש לחפש סדרה		לחצן
הכותרות והמחירים משתנים		תווית
היכן תציג את ממשק המשתמש		תווית
דבר ש-Amazon		TinyWebDB
סרוק ספרים		סרוק ברקוד

הגדר את המאפיינים של הרכיבים בצורה הבאה:

1.הגדר את הרמז של SearchTextBox ל"הזן מילות מפתח או ISBN".

2.הגדר את המאפיינים של הכפתורים והתוויות כך שהם יופיעו כפי שמוצג ב
איור 5-13.

3.הגדר את המאפיין ServiceURL של רכיב TinyWebDB ל-
<http://aiamazonapi.appspot.com/>.

תכנות התנהגות האפליקציה

עבור אפליקציה זו, תציין את ההתנהגויות הבאות בעורך הבלוקים:

חיפוש לפי מילת מפתח

המשתמש מזין מונחים מסוימים ולוחץ על כפתור KeywordSearch כדי להפעיל
חיפוש באמזון. אתה תתקשר ל- TinyWebDB1.GetValue כדי שזה יקרה.

חיפוש לפי ISBN

המשתמש מזין ISBN ולוחץ על כפתור ISBNB. אתה תארוז את הקידומת "isbn:"
עם המספר שהוזן ותפעיל את החיפוש של אמזון.

סריקת ברקוד

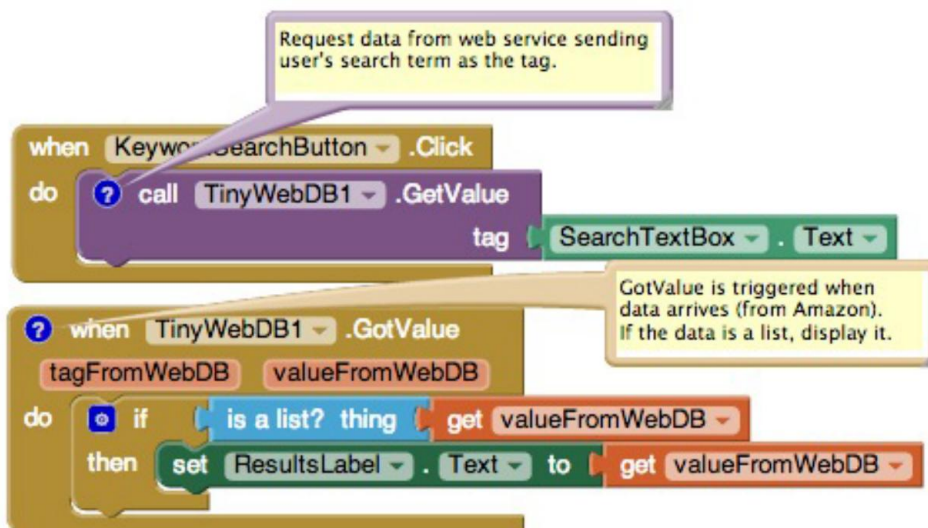
המשתמש לוחץ על כפתור והסורק מופעל. כאשר המשתמש סורק ISBN מתוך
ספר, האפליקציה שלך תפעיל חיפוש לפי ISBN.

עיבוד רשימת הספרים

בהתחלה, האפליקציה שלך תציג את הנתונים שהוחזרו מאמזון בצורה בסיסית.
מאוחר יותר, תשנה את הבלוקים כך שהאפליקציה תחליף את הכותרת, המחיר
וה-NBSI מכל ספר שהוחזר ותציג אותם בצורה מסודרת.

חיפוש לפי מילת מפתח

כאשר המשתמש לוחץ על כפתור KeywordSearch, אתה רוצה לתפוס את הטקסט מ-
SearchTextBox ולשלוח אותו בתור התג בבקשה שלך ל-API. nozamA אתה תשתמש בגוש
TinyWebDB.GetValue כדי לבקש את החיפוש באמזון. כאשר התוצאות יחזרו מאמזון, מטפל
האירועים TinyWebDB.GotValue יופעל. לעת עתה, פשוט הצג את התוצאה שמוחזרת
ישירות לתווית התוצאות, כפי שמוצג באיור 6-13 בהמשך, לאחר שתראה שהנתונים אכן
מאוחרים, תוכל להציג את הנתונים בצורה מתוחכמת יותר.



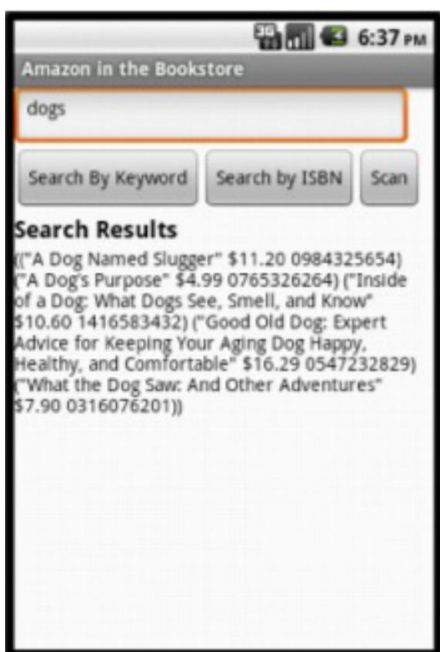
איור 13-7. שלח את בקשת החיפוש API-לושם תוצאות ResultsLabel-ב

איך הבלוקים עובדים

כאשר המשתמש לוחץ על כפתור KeywordSearch, מתבצעת בקשת TinyWebDB1.GetValue. התג שנשלח עם הבקשה הוא המידע שהמשתמש הזין ב- SearchTextBox. השלמת את אפליקציית MakeQuiz (פרק 10), יודע שבקשות TinyWebDB1.GetValue לא נענות מיד. במקום זאת, כאשר הנתונים מגיעים מהממשק TinyWebDB1.GotValue, ה-API המופעל. ב- GotValue, הבלוקים בודקים את הערך המוחזר כדי לראות אם זו רשימה (זה לא יהיה אם ה-API השל אמזון לא מקוון או שאין נתונים עבור מילות המפתח). אם מדובר ברשימה, הנתונים ממוקמים ב- ResultsLabel.



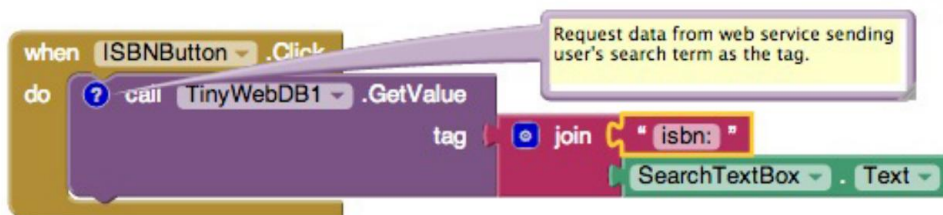
בדוק את האפליקציה שלך הזן מונח בתיבת החיפוש ולחץ על חפש לפי מילות מפתח. אתה אמור לקבל רשימה דומה למה שמוצג באיור 13-7. (זה לא ממש יפה למראה, אבל נטפל בזה בקרוב.)



איור 8-13. תוצאת חיפוש של מילות מפתח עבור "כלבים"

חיפוש לפי ISBN

הקוד לחיפוש לפי ISBN דומה, אבל במקרה זה API-השל Amazon מצפה שהתג יהיה בצורת (זהו) "isbn:xxxxxxxxxxxxxxxx" הפרוטוקול ש-IPA מצפה לחיפוש לפי (ISBN. אינך רוצה לאלץ את המשתמש לדעת את הפרוטוקול הזה; המשתמש צריך רק להיות מסוגל להזין את ה-ISBN-ה בתיבת הטקסט וללחוץ על חפש לפי ISBN, והאפליקציה צריכה להוסיף את הקידומת "isbn:" מאחורי הקלעים עם טקסט עשה. איור 8-13 מציג את הבלוקים לעשות זאת.



איור 9-13. האפליקציה מקדימה "isbn:" לחיפוש כך שהיא תחפש ספר מסוים

איך הבלוקים עובדים

בלוק הצטרפות משרשר את הקידומת "isbn:" עם המידע שהמשתמש הזין ב- SearchTextBox ושולח את התוצאה כתג ל- TinyWebDB1.GetValue.

228 פרק: 13 אמזון בחנות הספרים

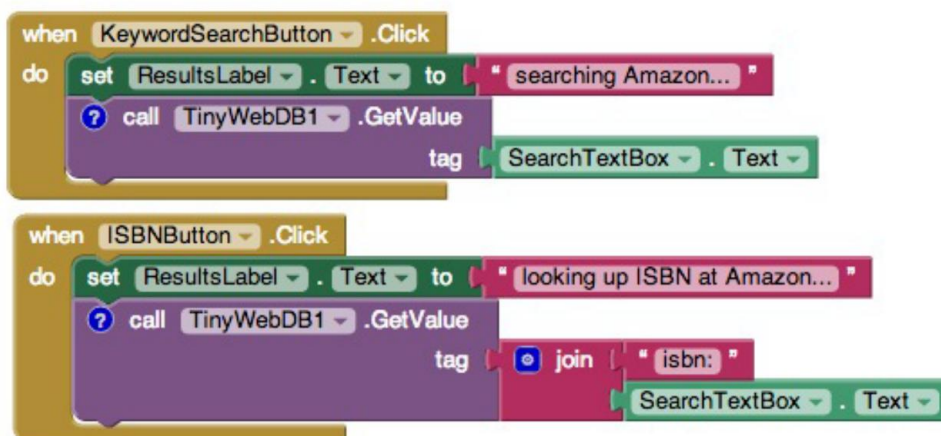
בדיוק כמו בחיפוש מילות מפתח, API-השולח בחזרה תוצאת רשימה עבור חיפוש ISBN-in במקרה זה, רשימה של הפריט היחיד שה-NBSI שלו תואם בדיוק את הקלט של המשתמש. מכיוון שמטפל האירועים TinyWebDB.GetValue כבר מוגדר לעבד רשימה של ספרים (אפילו רשימה עם פריט אחד בלבד), לא תצטרך לשנות את מטפל האירועים שלך כדי שזה יעבוד.



בדוק את האפליקציה שלך הן ISBN (למשל, 9781449397487) ולחץ על SearchTextBox האם פרטי הספר מופיע

אל תשאר את המשתמשים שלך תלויים

כאשר אתה מתקשר לשירות אינטרנט (API) עם TinyWebDB1.GetValue יכול להיות עיכוב לפני הגעת הנתונים ו-TinyWebDB1.GetValue מופעל. בדרך כלל מומלץ ליידע את המשתמשים שהבקשה מעובדת כדי להרגיע אותם שהאפליקציה לא נתקעה. עבור אפליקציה זו, אתה יכול לשים הודעה ב- ResultsLabel בכל פעם שאתה מבצע את הקריאה ל- TinyWebDB1.GetValue, כפי שמוצג באיור 9-13.



איור 10-13 הוספת הודעה כדי ליידע את המשתמש מה קורה

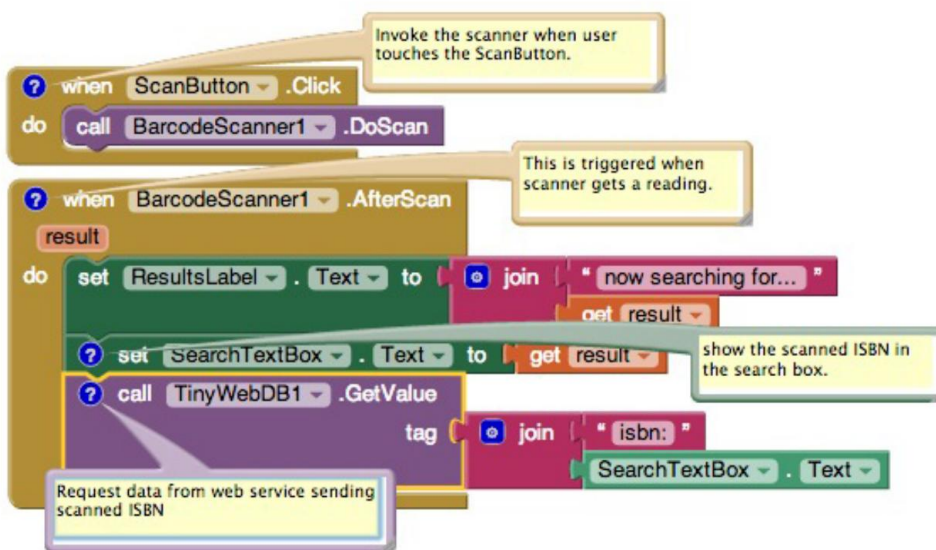
איך הבלוקים עובדים

הן עבור מילת המפתח והן עבור חיפוש ISBN, הודעת "חיפוש באמזון..." ממוקמת ב- ResultsLabel כאשר הנתונים מתבקשים. שימו לב שכאשר GetValue מופעלת, הודעה זו נמחקת עם התוצאות בפועל מאמזון.

סריקת ספר

בואו נודה בזה: הקלדה בפלאפון היא לא תמיד הדבר הכי קל, ואתם נוטים לטעות פה ושם. זה בהחלט יהיה קל יותר (ויגרם לפחות טעויות) אם משתמש יוכל פשוט להפעיל את האפליקציה שלך ולסרוק את הברקוד של הספר. זוהי עוד תכונת טלפון אנדרואיד מובנית נהדרת שתוכלו להתחבר אליה בקלות עם App Inventor.

הפונקציה BarcodeScanner.DoScan מפעילה את הסורק. אתה תרצה לקרוא לזה כאשר לוחצים על כפתור הסריקה. המטפל באירועים BarcodeScanner.AfterScan הוא מופעל ברגע שמשוהו נסרק. יש לו ארגומנט אחד, תוצאה, שמכיל את המידע שנסרק. במקרה זה, אתה רוצה להתחיל חיפוש ISBN באמצעות תוצאה זו, כפי שמוצג באיור 10-13.



איור 10-13. חסימות לתחילת חיפוש ISBN לאחר סריקה של משתמש

איך הבלוקים עובדים

כאשר המשתמש לוחץ על לחצן הסריקה, DoScan מפעיל את הסורק. כאשר משוהו נסרק, AfterScan מופעל. תוצאת הטיעון מכילה את תוצאת הסריקה - במקרה זה, ISBN של ספר. המשתמש מקבל הודעה על כך שנעשתה בקשה, התוצאה (מספר ISBN-ההסרוק) ממוקמת ב- SearchTextBox, TinyWebDB1.GetValue נקרא להתחיל את החיפוש. שוב, מטפל האירועים TinyWebDB1.GetValue יעבד את פרטי הספר המוחזרים.



בדוק את האפליקציה שלך לחץ על לחצן הסריקה וסרוק את הברקוד של ספר. האם האפליקציה מציגה את המידע על הספר?

שיפור התצוגה

אפליקציית לקוח כמו זו שאתה יוצר יכולה לעשות מה שהיא רוצה עם הנתונים שהיא מקבלת - אתה יכול להשוות את פרטי המחיר לזה של חנויות מקוונות אחרות, או להשתמש בפרטי הכותרת כדי לחפש ספרים דומים מספרייה אחרת. כמעט תמיד, תרצה לטעון את מידע API-הלמשתנים שתוכל לעבד הלאה. במטפל האירועים TinyWebDB.GotValue שיש לך עד כה, אתה פשוט מכניס את כל המידע שהוחזר מאמזון ל- ResultsLabel במקום זאת, בואו נעבד את הנתונים על ידי (1) הוספת הכותרת, המחיר וה-NBSI של כל ספר שהוחזר למשתנים נפרדים, ו- (2) הצגת הפריטים הללו בצורה מסודרת. אם השלמת חלק מהפרקים הקודמים, כנראה שאתה מבין בהגדרת משתנים ושימוש בהם בתצוגה שלך, אז נסה לבנות את המשתנים שאתה חושב שתזדקק להם ואת הבלוקים להצגת כל תוצאת חיפוש בה. קו נפרד משלו. לאחר מכן, השווה את מה שעשית עם איור 13-11.

Each bookItem in the list is itself a list. The title, cost, and isbn are the 1st, 2nd, and 3rd items of each sublist. Extract them, then display on separate lines (\n is new line).

איור 13-12. חילוץ הכותרת, העלות וה-NBSI של כל ספר, ולאחר מכן הצגתם בשורות נפרדות

איך הבlookים עובדים

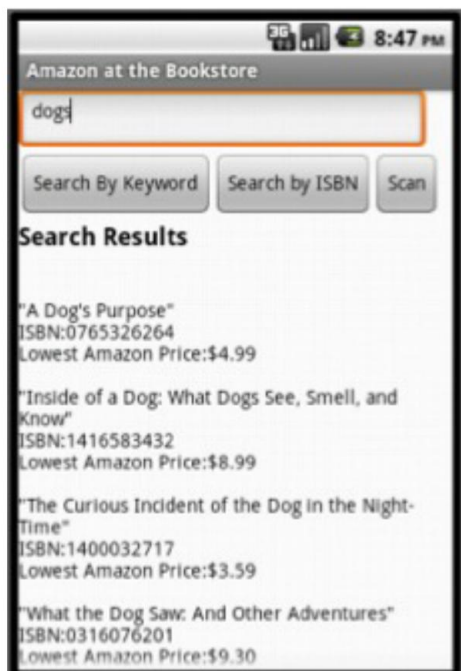
ארבעה משתנים -תוצאות רשימה, כותרת, עלות ו- isbn מוגדרים כך שיכילו כל פיסת נתונים כפי שהיא מוחזרת מהממשק API-ההתוצאה מה-IPA, valueFromWebDB ממקמת במשתנה resultList. האפליקציה זו הייתה יכולה לעבד את הארגומנט valueFromWebDB ישירות, אבל באופן כללי, תכניס אותו למשתנה למקרה שתצטרך לעבד את הנתונים מחוץ למטפל באירועים. (ארגומנטים של אירוע כמו valueFromWebDB מחזיקים את הערך שלהם רק בתוך המטפל באירועים.)

עבור כל לולאה משמש כדי לחזור על כל פריט של התוצאה. נזכיר כי הנתונים המוחזרים מאמזון הם רשימה של רשימות, כאשר כל תת-רשימה מייצגת את המידע עבור ספר. אז, שם placeholder-השל עבור כל אחד שונה לפריט ספר, והוא מחזיק את פריטי הספר הנוכחיים (רשימה) בכל איטרציה. כעת עלינו להתמודד עם העובדה שהפריט המשתנה הוא רשימה - הפריט הראשון היא הכותרת; השני, המחיר; והשלישי, ISBN-הלפיקר, אנו משתמשים בlookים של פריטי רשימה נבחרים כדי לחלץ פריטים אלה ולהציב אותם במשתנים המתאימים שלהם (כותרת, מחיר ו- isbn).

לאחר שהנתונים אורגנו למשתנים, תוכל לעבד אותם איך שתצטרך. האפליקציה הזו פשוט משתמשת במשתנים כחלק מבלוק הצטרפות שמציג את הכותרת, המחיר וה-NBSI בשורות נפרדות.



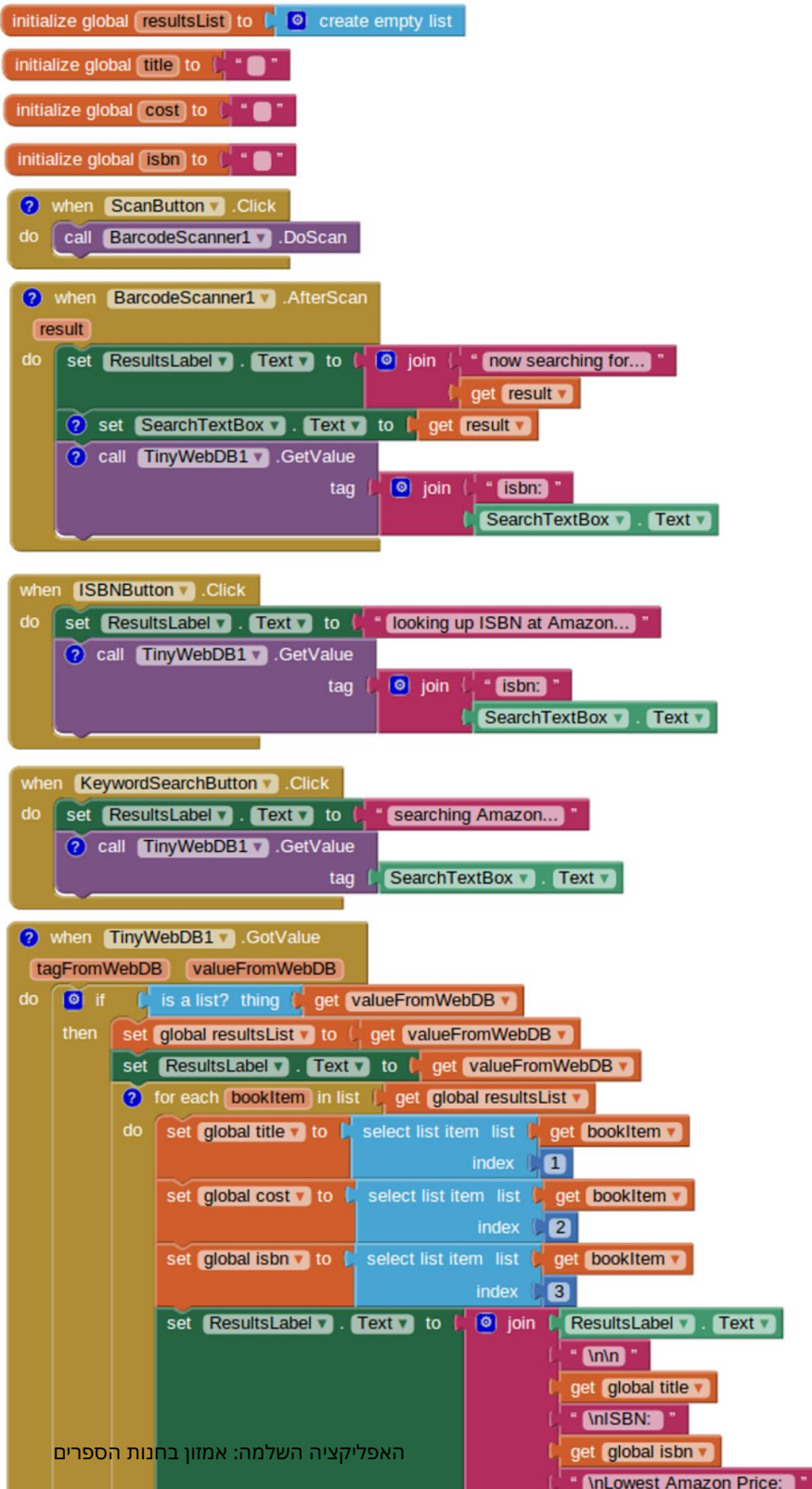
בדוק את האפליקציה שלך נסה חיפוש נוסף ובדוק כיצד המידע על הספר מוצג. זה צריך להיראות דומה לתמונה 12-13



איור 13-13. רישום החיפוש הוצג בצורה מתוחכמת יותר

האפליקציה השלמה: אמזון בחנות הספרים

איור 13-13 מציג את תצורת הבלוק הסופי עבור אמזון בחנות הספרים.



התאמה אישית של API-ה

API-השאליו התחברת, <http://aiamazonapi.appspot.com>, נוצר עם שפת התכנות Python i-App Engine של גוגל. App Engine מאפשר לך ליצור ולפרוס אתרים ושירותים (APIs) החיים בשרתים של Google. אתה משלם עבור App Engine רק אם האתר או API-ה שלך הופכים פופולריים ומושכים הרבה כניסות.

שירות API-ההמשמש במדריך זה מספק גישה חלקית בלבד ל-API nozama והמלא ומחזיר לכל היותר חמישה ספרים לכל חיפוש. אם תרצה לספק גמישות רבה יותר - לדוגמה, בקש ממנו לחפש פריטים שאינם ספרים - תוכל להוריד את קוד המקור של השירות מ- <http://amazon/appinventorapi.com/> ולתאים אותו אישית. התאמה אישית כזו אכן דורשת ידע בתכנות Python, אז היזהרו! אבל, אם סיימתם את האפליקציות App Inventor בספר זה, אולי אתם פשוט מוכנים להתגר. כדי להתחיל ללמוד Python, בדוק את הגרסה האינטראקטיבית של הספר *How to Think Like a Computer Scientist: Learning with Python* ולאחר מכן עיין בסעיף על בניית API-השל App Inventor בפרק 24 של ספר זה.

וריאציות

לאחר שתפעיל את האפליקציה, אולי תרצה לחקור כמה מהוריאציות הבאות:

- כפי שהיא, האפליקציה נתקעת אם החיפוש אינו מחזיר ספרים (לדוגמה, כאשר המשתמש מזין ISBN לא חוקי). שנה את החסימות כך שהאפליקציה תדווח כשאינן תוצאות.

- שנה את האפליקציה כך שתציג רק ספרים מתחת ל-\$0.1.

- שנה את האפליקציה כך שאחרי שתסרוק ספר, המחיר הנמוך ביותר באמזון יושמע בקול (השתמש ברכיב TextToSpeech שנדון באפליקציית No Text While Driving בפרק 4).

- הורד את <http://aiamazonapi.appspot.com> וקוד API ושנה אותו כך שיחזיר מידע נוסף. לדוגמה, ייתכן שתבקש ממנו להחזיר את כתובת האתר של אמזון של כל ספר, להציג את כתובת האתר יחד עם כל ספר ברשימה, ולתת למשתמש ללחוץ על כתובת האתר כדי לפתוח את הדף הזה. כפי שהוזכר קודם לכן, שינוי API-הדורש תכנות של Python וידע מסוים App Engine בשל גוגל. למידע נוסף, עיין בפרק 24.

סיכום

הנה כמה מהמושגים שכיסינו באפליקציה הזו:

•אתה יכול לגשת לאינטרנט מאפליקציה באמצעות TinyWebDB ובמיוחד ממשיקי API שנבנו. אתה מגדיר את ה- ServiceURL של רכיב TinyWebDB לכתובת API-הולאחר מכן מתקשר ל- TinyWebDB.GetValue כדי לבקש את המידע. הנתונים אינם מוחזרים מיד, אלא ניתן לגשת אליהם בתוך המטפל באירועים . TinyWebDB.GotValue

•הפונקציה BarcodeScanner.DoScan מפעילה את הסריקה. כאשר המשתמש סורק את א ברקוד, האירוע BarcodeScanner.AfterScan מופעל והנתונים הסרוקים ממוקמים בתוצאת הארגומנט.

App Inventor ב- •נתונים מורכבים מיוצגים עם רשימות ורשימות של רשימות. אם אתה יודע את הפורמט של הנתונים המוחזרים, API-מאתה יכול להשתמש עבור כל פריט ובחר ברשימה כדי לחלץ את פיסות המידע הנפרדות למשתנים, ולאחר מכן לבצע כל עיבוד או להגדיר את התצוגה איך שתמצה באמצעות המשתנים האלה .

