

תקשורת עם האינטרנט

איור 24-1.

הטכנולוגיה הניידת והטבע הנפוצה של האינטרנט שינו את העולם בו אנו חיים. כעת אתה יכול לשבת בפארק ולעשות את הבנקאות שלך, לחפש Amazon.com בכדי למצוא ביקורות על הספר שאתה קורא, ולבדוק בטוויטר כדי לראות מה אנשים בכל פארק אחר בעולם חושבים על. טלפונים ניידים עברו הרבה מעבר להתקשרות והודעות טקסט -כעת, יש לך גם גישה מיידית לנתוני העולם.



אתה יכול להשתמש בדפדפן של הטלפון שלך כדי להגיע לאינטרנט, אבל לעתים קרובות המסך הקטן והמהירות המוגבלת של מכשיר נייד יכולים להפוך את זה לבעייתי.

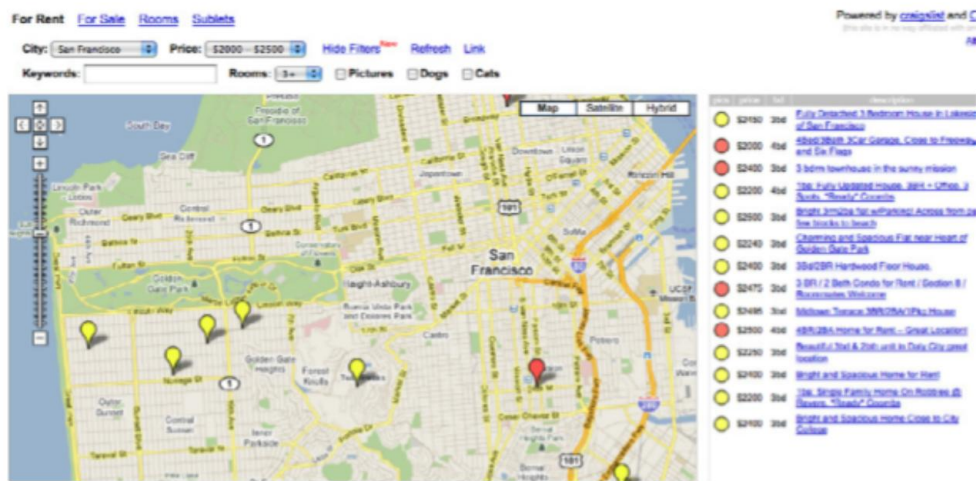
אפליקציות מותאמות אישית, שתוכננו במיוחד למשוך גושים קטנים של מידע מתאים במיוחד מהרשת, יכולות לספק אלטרנטיבה אטרקטיבית יותר לדפדפן הנייד.

בפרק זה, נסקור את רכיבי App Inventor שניגשים למידע מהאינטרנט. תלמד כיצד להציג דף אינטרנט בממשק המשתמש של האפליקציה שלך, ותלמד על ממשקי API וכיצד לגשת למידע משירות אינטרנט.

יצירתיות היא לערבב מחדש את העולם, שילוב (מעיקה) של רעיונות ותכנים קיימים בדרכים חדשות מעניינות. אמינם הוא בין אמנים רבים בעשורים האחרונים שהפכו את המוזיקה לפופולרית כאשר הציב את הקול שלו Slim Shady על רצועות AC/DC i-Vanilla Ice. של "דגימה" נפוץ כיום, ואמנים רבים, כולל Girl Talk i-Negativland מתמקדים בעיקר ביצירת רצועות חדשות משילוב של תוכן ישן.

עולם האינטרנט והמובייל אינם שונים: אתרים ואפליקציות לערבב מחדש תוכן מ מקורות נתונים שונים, ורוב האתרים מעוצבים כעת עם מחשבה על יכולת פעולה הדדית כזו. דוגמה להמחשה mashup-לבאינטרנט היא Housing Maps, בתמונה באיור 24-1, שלוקחת מידע על השכרת דירה מ-Craigslist ומרסקת אותו עם API-השל מפות Google.

פרק 362: 24 תקשורת עם האינטרנט

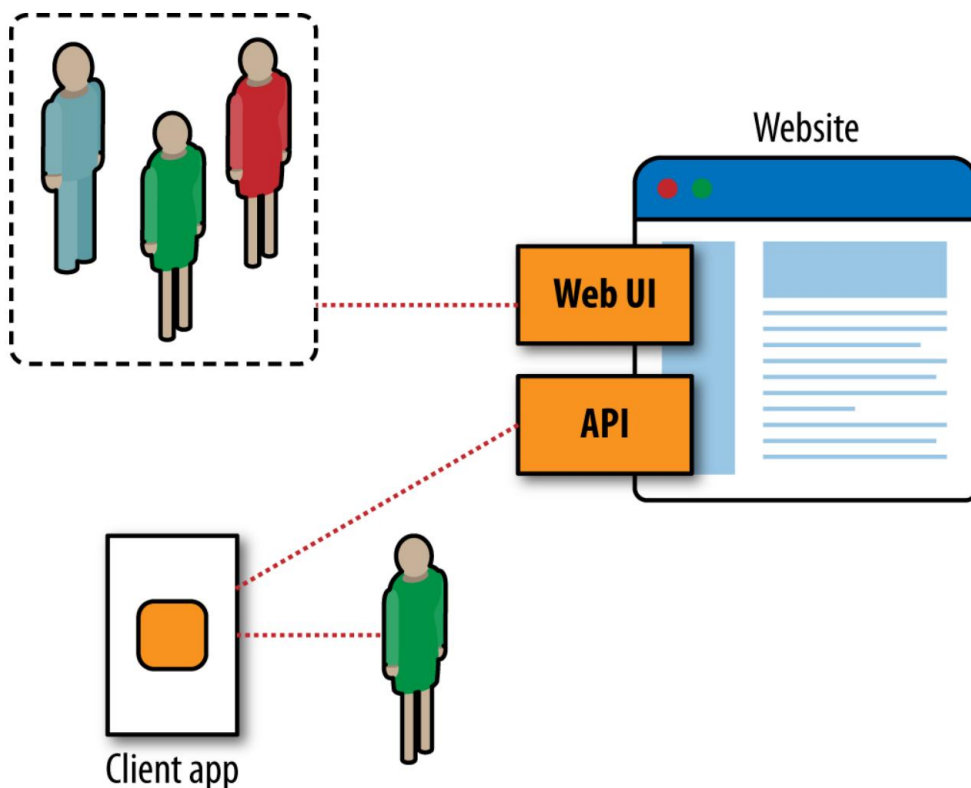


איור 24-2. מפות דיור מעבדות מידע i-Craigslist ו-Google Maps

אפשרויות משימות הדומות למפות דיור בגלל שירותים כמו מפות גוגל לספק גם אתר וגם ממשק API של שירות אינטרנט מתאים. אנו, בני האדם, מבקרים ב-<http://maps.google.com/> בדפדפן, אך אפליקציות כגון Housing Maps מתקשרות בין מכוונה למכוונה באמצעות ממשק API-השל מפות Google. Mashups מעבדים את הנתונים, משלבים אותם עם נתונים מאתרים אחרים (למשל, (Craigslist) ואז מציגים אותם בחדשים ומעניינים דרכים.

כמעט כל אתר פופולרי מספק כעת את החלופה הזו, גישה ממכוונה למכוונה. התוכנית המספקת את הנתונים נקראת שירות אינטרנט, והפרוטוקול לאופן שבו אפליקציית לקוח צריכה לתקשר עם השירות נקרא ממשק מתכנת יישומים, או API. בפועל, המונח API משמש גם להתייחסות לשירות האינטרנט.

שירות האינטרנט של אמזון (AWS) היה אחד משירותי האינטרנט הראשונים, שכן אמזון הבינה שפתיחת הנתונים שלה לשימוש על ידי ישויות צד שלישי תוביל בסופו של דבר למכירת ספרים נוספים. כשפייסבוק השיקה את API-השלה ב-2002, אנשים רבים הרימו גבות. הנתונים של פייסבוק אינם פרסומות ספרים, אז למה שהיא תאפשר לאפליקציות אחרות "לגנוב" את הנתונים האלה ולמשוך משתמשים רבים מאתר פייסבוק (והפרסומות שלו)? עם זאת, הפתיחות שלה הובילה את פייסבוק להפוך לפלטפורמה במקום רק לאתר - כלומר תוכניות אחרות יכולות להתבסס על הפונקציונליות של פייסבוק ולהיעזר בה, ואף אחד לא יכול להתווכח עם הצלחתה כיום. עד שהושק טוויטר ב-2002, הגישה API-להייתה ציפיה, לא חידוש, וטוויטר פעלה בהתאם. כעת, כפי שמוצג באיור 24-2, רוב האתרים מציגים גם ממשק API וגם ממשק אנושי.

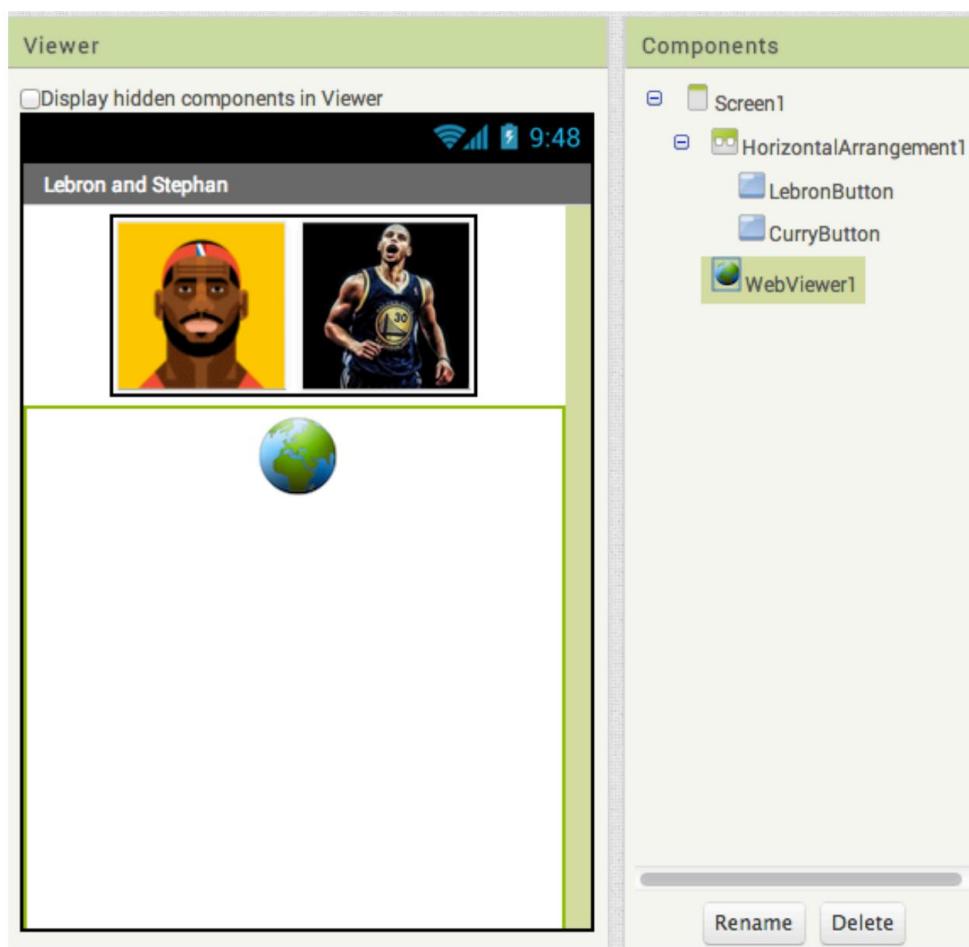


איור 3-24. רוב האתרים מספקים גם ממשק אנושי וגם ממשק API עבור אפליקציות לקוח

לפיכך, הרשת היא דבר אחד עבורנו בני האדם הממוצעים (אוסף של אתרים לביקור). ל מתכנתים, זהו מסד הנתונים הגדול והמגוון ביותר בעולם של מידע.

רכיב WebView-ה

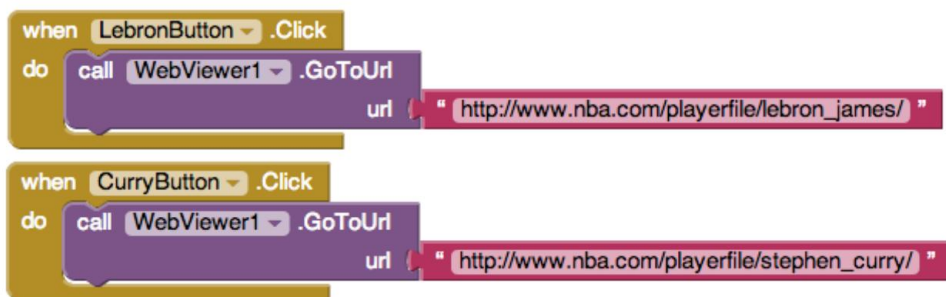
רכיב WebView מאפשר לך להציג דף אינטרנט בתוך האפליקציה שלך. אתה יכול להציג דף מפות Google המציג את מיקומו הנוכחי של המשתמש, דף טוויטר המציג את הנושאים הפופולריים ביותר הקשורים לאפליקציה שלך, או דף nba.com מהמציג את הנתונים הסטטיסטיים של השחקנים האהובים עליך. WebView (ראה איור 3-24) דומה לרכיב Canvas שהוא מגדיר תת-פאנל של המסך. אבל בעוד Canvas משמש לציורים והנפשות, WebView מציג דף אינטרנט.



איור 4-24: Web-WebView. Designer-ב

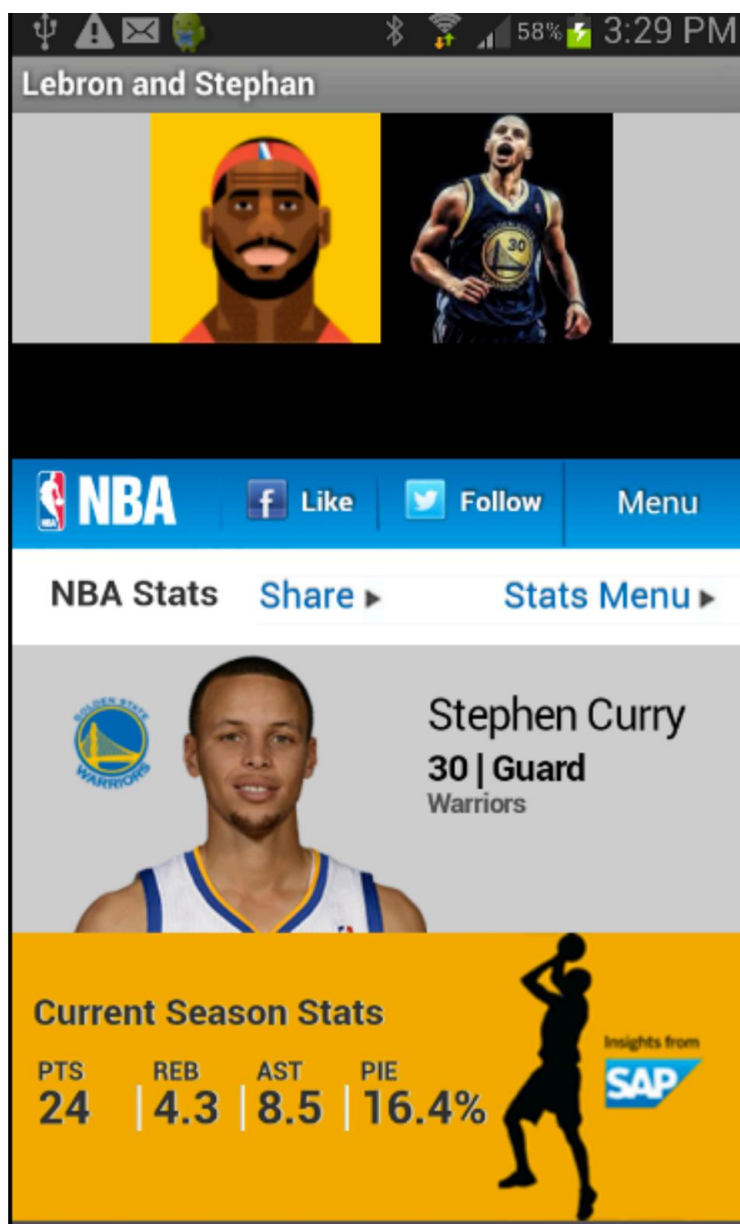
אתה יכול לגרור `WebView` במהמגירה של ממשק המשתמש. לאחר מכן תוכל לשנות באופן דינמי את כתובת האתר המופיעה, כמו באיור 4-24, שמתארת בלוקים מאפליקציה המציגה את הנתונים הסטטיסטיים של שחקני NBA-הלברון ג'יימס וסטיבן קארי:

רכיב 365 WebView-ה



איור 24-5. חסימות כדי להציג את דף האינטרנט עבור השחקנים שנבחרו

אם המשתמש יקיש על התמונה של סטיבן קארי, האפליקציה תציג את הדף שלו
24-5. WebViewer, כמו באיור 24-5. nba.com-מב-



איור 24-6. WebViewer באפליקציה

רכיב האינטרנט

בעוד WebViewer-שמציג דף אינטרנט, רכיב האינטרנט, רכיב חדש יחסית, Inventor, App-במאפשר לאפליקציה לתקשר עם שירות אינטרנט באמצעות פרוטוקול ההעברה הסטנדרטי של HTTP (Hypertext פרוטוקול זה מספק Get, Put

שיטות פרסום להכנסת מידע לאפליקציה שלך. המידע מגיע לא כדף הניתן לתצוגה, אלא כנתונים שתוכל להציג או לעבד כרצונך.

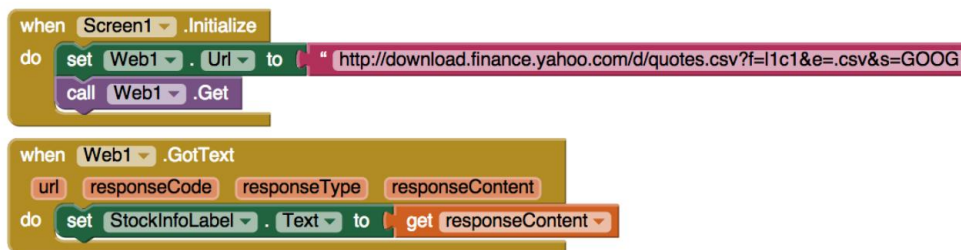
הרכיב ברמה נמוכה למדי, והשימוש בו דורש מומחיות מסוימת בתכנות. בדרך כלל אתה מגדיר את המאפיין Web.URL כדי לציין עם איזה שירות אינטרנט תתקשר, ואז אתה קורא לאחת משיטות HTTP-הכדי לבקש פעולה כלשהי. זה מסובך כי אתה צריך להבין את ה-API של שירות האינטרנט (הפרוטוקול לתקשורת), ואתה צריך להבין איך לעבד את המידע ששירות האינטרנט מחזיר לאפליקציה שלך. עיבוד זה ידוע בשם ניתוח, וזוהי טכניקת תכנות מתקדמת.

בפרק זה, תכירו את רכיב האינטרנט באמצעות דוגמה פשוטה יחסית הניגשת למידע על מחירי מניות פיננסיים API-מציבורי שזמין על ידי Yahoo Finance הפרוטוקול לדבר עם ה-API הזה הוא פשוט למדי, והנתונים המוחזרים נמצאים ברשימת ערכים מופרדים בפסיקים (ערכים מופרדים בפסיקים, או CSV) שהוא משמש כהקדמה נחמדה לתקשורת API.

לרוע המזל, לרוב ממשקי ה-API היש סכימות הרשאות וממשקי API מסובכים, ולעתים קרובות הם מחזירים נתונים בפורמטים כגון JavaScript Object Notation (JSON), או XML הדורשים קוד מתקדם כדי לנתח.

מדגם שוק המניות

איור 24-6 מציג את החסימות של אפליקציה שמציגה מידע על מניות Google כאשר האפליקציה מופעלת.



איור 24-7. גישה למידע על מלאי חיים באמצעות רכיב האינטרנט

לממן. כאשר Web1.Get נקרא, הבקשה מתבצעת, אך לא מוחזרים נתונים מיידית. On Screen.Initialize, Web1.Url מוגדר לכתובת ה-LRU לתקשורת עם Yahoo

במקום זאת, כאשר יאהו מחזירה את הנתונים המבוקשים לאפליקציה שלך, אירוע Web1.GetText מופעל, וכאן תוכל לעבד את הנתונים המוחזרים. פרמטר האירוע responseContent מחזיק את הנתונים. כאמור, ממשק ה-API של Yahoo Finance מחזיר נתונים בפורמט CSV. אם תבנה את האפליקציה הזו ותפעיל אותה, תראה שהנוכחי

מחיר מניית Google והשינוי במחיר היום מוצגים ב-, StockInfoLabel מופרדים בפסיקים.

אתה יכול להתאים אישית את Web.Url-הכדי לקבל את המידע עבור חברה (או חברות) שונות וכדי לקבל סוגים שונים של מידע על שוק המניות. API-השל Yahoo Finance, בכתובת <https://code.google.com/p/yahoo-finance-managed/wiki/CSVAPI>, מפרט כיצד תוכל לשנות את כתובת האתר כדי להתאים אישית את הבקשה שלך, כמו גם את פורמט הנתונים שהיא החזרת.

ממשקי API תואמי TinyWebDB ו-TinyWebDB

רכיב האינטרנט מספק שיטה לגישה לממשקי API. ממשקי API הוא פשוט למדי, כמו Yahoo Finance מתכנתים מתחילים יכולים להשתמש ברכיב האינטרנט כדי לגשת אליו ישירות. אבל ממשקי API אחרים, כמו API-השל Amazon שהוצג בפרק 13, הם מסובכים יותר.

עבור ממשקי API מסובכים, מתכנת מנסה יכול להגדיר שירות אינטרנט תואם TinyWebDB שיוכל לשמש מתכנתי App Inventor פחות מנסים כדי לגשת. API-לכאשר שירות כזה מוגדר, מתכנתים אחרים יכולים לגשת לשירות האינטרנט באמצעות פרוטוקול תג-ערך הפשוט הטמון בפונקציה . TinyWebDB.GetValue אתה שולח תג מסוים כפרמטר, ואובייקט רשימה או טקסט מוחזר כערך. בדרך זו, מתכנת App Inventor מוגן מהתכנות הקשה הנדרש לניתוח (להבין ולחלץ נתונים) פורמטים סטנדרטיים של נתונים כגון XML או JSON.

"תואם TinyWebDB" פירושו רק שירות אינטרנט שעוקב אחר הפרוטוקול הצפוי של TinyWebDB: הוא מצפה לבקשה ספציפית, ומחזיר נתונים TinyWebDB-שיכולה להבין. שירות האינטרנט של Amazon API המשמש בפרק 13 הוא דוגמה לשירות אינטרנט כזה, והוא יכול לשמש כדוגמה למתכנתים שרוצים להגדיר שירות כזה (למשל, אם אתה מורה ומעוניין לספק גישה API-לכלשהו עבור התלמידים שלך).

בעבר, בניית ממשקי API הייתה קשה מכיוון שלא רק היית צריך להבין את התכנות ואת פרוטוקולי האינטרנט, אלא גם צריך להגדיר שרת שיארח את שירות האינטרנט שלך, ומסד נתונים כדי לאחסן את הנתונים. כעת, זה הרבה יותר קל מכיוון שאתה יכול למנף כלי מחשוב ענן כגון App Engine של גוגל ו-Elastic Compute Cloud של אמזון כדי לפרוס מיד את השירות שאתה יוצר. הפלטפורמות הללו לא רק יארחו את שירות האינטרנט שלך, אלא גם יאפשרו למאות משתמשים לגשת אליו לפני שיגבו ממך אגורה אחת. כפי שאתה יכול לדמיין, אתרים אלה הם ברכה גדולה לחדשנות.

הפרטים של יצירת שירות אינטרנט תואם TinyWebDB מעבר לתחום של הספר הזה. אבל אם אתה מעוניין, בדוק את התיעוד והדוגמאות בכתובת <http://appinventorapi.com/>.

סיכום

רוב אתרי האינטרנט ואפליקציות סלולריות רבות אינן ישויות עצמאיות; כדי לבצע את עבודתם, הם מסתמכים על יכולת פעולה הדדית של אתרים אחרים. עם App Inventor, אתה יכול לבנות משחקים, חידונים ואפליקציות עצמאיות אחרות, אבל בקרוב תיתקל בבעיות הקשורות לגישה לאינטרנט. האם אני יכול לכתוב אפליקציה שאומרת לי מתי האוטובוס הבא יגיע לתחנה הרגילה שלי? האם אני יכול לכתוב אפליקציה ששולחת הודעות טקסט לתת-קבוצה מיוחדת של חבריי בפייסבוק? האם אני יכול לכתוב אפליקציה ששולחת ציורים? App Inventor מספק שלושה רכיבים שיכולים לדבר עם האינטרנט: WebView להצגת דף אינטרנט חי; רכיב האינטרנט, לגישה למידע API-M; ורכיב TinyWebDB לגישה לנתונים API-באינטרנט שעוצב במיוחד.

גישה API-ליכולה להיות מסובכת; אתה צריך לדעת את הפרוטוקול לבקשת מידע, ואתה צריך לעבד (לנתח) את הנתונים המורכבים המוחזרים לעתים קרובות. אבל התגמול עבור ללמוד איך לעשות זאת הוא גדול; האפליקציות שלך יכולות ליצור אינטראקציה עם העולם!

