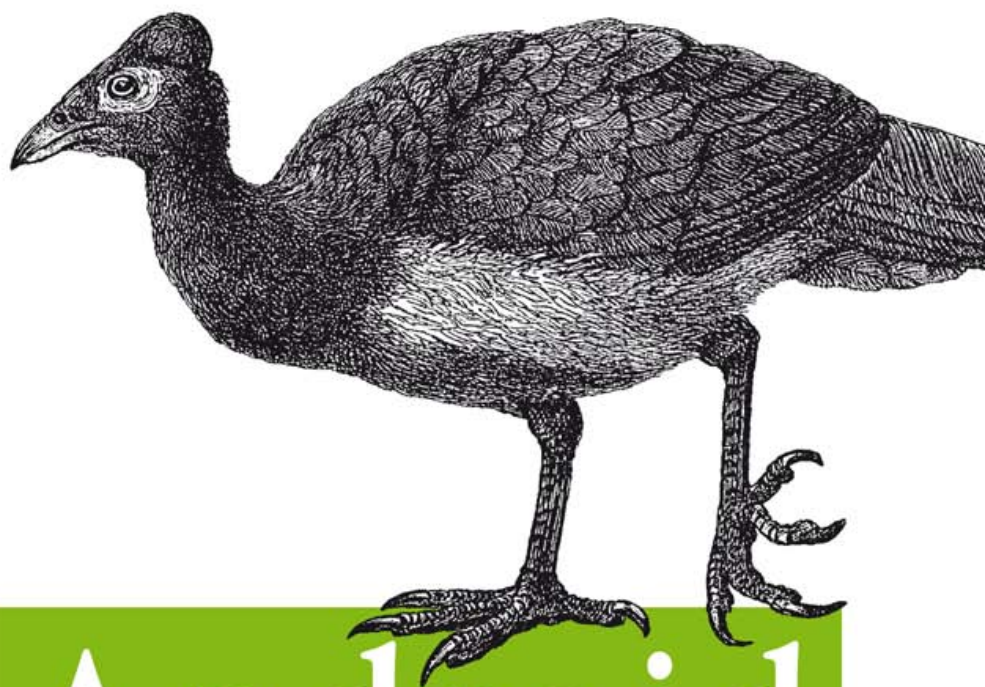


*Twórz natywne aplikacje za pomocą
standardowych technologii internetowych*



Android

*Tworzenie aplikacji w oparciu
o HTML, CSS i JavaScript*



HELION

O'REILLY®

Jonathan Stark, Brian Jepson

Tytuł oryginału: Building Android Apps with HTML, CSS,
and JavaScript: Making Native Apps with Standards-Based Web Tools

Tłumaczenie: Aleksander Lamża (rozdz. 0 – 2, 4 – 8, dodatki), Michał Nowak (rozdz. 3)

ISBN: 978-83-246-4968-6

© 2013 Helion S.A.

Authorized Polish translation of the English edition Building Android Apps with HTML,
CSS, and JavaScript, 2nd Edition ISBN 9781449316419 © 2012 Jonathan Stark.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns
or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any form
or by any means, electronic or mechanical, including photocopying, recording or
by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu
niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą
kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym,
magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź
towarowymi ich właścicieli.

Wydawnictwo HELION dołożyło wszelkich starań, by zawarte w tej książce informacje
były kompletne i rzetelne. Nie bierze jednak żadnej odpowiedzialności ani za ich
wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub
autorskich. Wydawnictwo HELION nie ponosi również żadnej odpowiedzialności za
ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:
<ftp://ftp.helion.pl/przyklady/andrta.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/andrta>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wprowadzenie	9
1. Zaczynamy	13
Aplikacje internetowe kontra aplikacje natywne	13
Czym jest aplikacja internetowa?	13
Czym jest aplikacja natywna?	14
Wady i zalety	14
Które podejście jest odpowiednie dla Ciebie?	15
Ekspresowy kurs technologii internetowych	15
Wprowadzenie do języka HTML	15
Wprowadzenie do CSS	19
Wprowadzenie do JavaScriptu	23
2. Praca ze stylami	27
Nie masz własnej witryny?	27
Pierwsze kroki	29
Osobny arkusz stylów dla Androida	33
Kontrolowanie skalowania strony	35
Arkusz stylów dla Androida	36
Wygląd i zachowanie w stylu Androida	39
Proste zachowania z jQuery	41
Czego się nauczyłeś?	47
3. Zaawansowane stylizowanie	49
Szczypota Ajaksa	49
Kierowanie ruchem	50
Przygotowanie treści	52
Przekierowywanie żądań za pomocą JavaScriptu	53

Kilka wodotrysków	55
Wskaźnik postępu	56
Ustawianie tytułu strony	59
Obsługa długich tytułów	61
Automatyczne przewijanie do początku strony	62
Przechwytywanie odsyłaczy lokalnych	64
Dodawanie własnego przycisku „wstecz”	64
Dodawanie ikony do pulpitu	72
Czego się nauczyłeś?	73
4. Animacje	75
Z niewielką pomocą naszych przyjaciół	75
Przewijanie	76
Ekran Terminarz	79
Ekran Dzień	82
Ekran Nowy wpis	83
Ekran Ustawienia	86
Integracja całości	89
Dostosowywanie biblioteki jQuery	92
Czego się nauczyłeś?	94
5. Składowanie danych po stronie klienta	95
Web Storage	95
Zapisywanie ustawień aplikacji w localStorage	97
Zapisywanie daty w sessionStorage	100
Web SQL Database	102
Tworzenie bazy danych	104
Wstawianie wierszy	106
Wybieranie wierszy i obsługa otrzymanych danych	110
Usuwanie wierszy	115
Kody błędów Web SQL Database	118
Czego się nauczyłeś?	118
6. Tryb offline	121
Podstawy aplikacji działających offline	121
Sekcje NETWORK i FALLBACK	126
Dynamiczne tworzenie pliku manifestu	132

Debugowanie	136
Konsola JavaScriptu	137
Czego się nauczyłeś?	140
7. W stronę natywności	141
Wprowadzenie do PhoneGap	141
Tworzenie aplikacji z wykorzystaniem Eclipse i Android SDK	143
Pobieranie i instalowanie środowiska Eclipse Classic	143
Pobieranie i instalowanie Android SDK	144
Instalowanie dodatku ADT w Eclipse	144
Dodawanie platform Android i innych składników	145
Pobieranie najnowszej wersji PhoneGap	146
Tworzenie nowego projektu	146
Uruchamianie Kilo jako aplikacji dla Androida	148
Sterowanie telefonem z poziomu JavaScriptu	150
Dźwięki, wibracje i ostrzeżenia	151
Geolokalizacja	155
Akcelerometr	162
Czego się nauczyłeś?	166
8. Publikowanie aplikacji w sklepie Google Play	167
Przygotowanie aplikacji do dystrybucji	167
Usuwanie kodu na potrzeby debugowania	167
Nadawanie aplikacji numeru wersji	168
Kompilowanie i podpisywanie aplikacji	169
Przesyłanie aplikacji do sklepu Google Play	170
Bezpośrednie dystrybuowanie aplikacji	172
Dalsza lektura	173
Dodatek. Wykrywanie przeglądarek za pomocą WURFL	177

Animacje

W aplikacjach dla Androida można się spotkać z wieloma charakterystycznymi elementami animacji, które często są dodatkową informacją dla użytkownika. Przykładem może być przewijanie ekranów w lewo podczas nawigowania w głąb struktury aplikacji, a w prawo podczas powrotów. W tym rozdziale dowiesz się, jak do aplikacji internetowej dodać tego typu zachowania, dzięki czemu w jeszcze większym stopniu będzie przypominała aplikację natywną.

Z niewielką pomocą naszych przyjaciół

Prawda jest taka, że stworzenie animacji wyglądającej na natywną na stronie internetowej jest bardzo trudne. Na szczęście mamy do dyspozycji bibliotekę jQTouch autorstwa Davida Kanedy, dzięki której da się to osiągnąć znacznie prościej. Jest to rozprowadzana na zasadach open source wtyczka do jQuery, która rozwiązuje większość problemów przedstawionych do tej pory, a także wiele innych, dużo bardziej złożonych.



Najnowszą dostępną wersją biblioteki jQTouch jest 1.0b4, którą możesz pobrać ze strony <http://www.jqtouch.com/>. W kolejnych wersjach są planowane istotne zmiany, więc — mimo że w chwili czytania tej książki jest zapewne dostępna nowsza wersja biblioteki — pozostań lepiej przy 1.0b4. Kiedy będziesz się już czuć na tyle pewnie, by zmierzyć się z ewentualnymi problemami, możesz zaktualizować bibliotekę do najnowszej wersji.

Przewijanie

Tym razem przygotujemy prostą aplikację Kilo wspomagającą stosowanie diety. Jej podstawową funkcjonalnością ma być dodawanie (i usuwanie) informacji o posiłkach do poszczególnych dni. Aplikację podzielimy na pięć ekranów: *Główny*, *Ustawienia*, *Terminarz*, *Dzień* oraz *Nowy wpis*. Zaczniemy od dwóch ekranów, a pozostałe dodamy później.



Do elementów dokumentu HTML będziemy stosować różne klasy CSS (takie jak `toolbar`, `edgetoedge`, `arrow`, `button` czy `back`). Wszystkie one odpowiadają predefiniowanym klasom pochodzącym z domyślnego arkusza CSS biblioteki jQTouch. Możesz oczywiście tworzyć i stosować własne klasy oraz modyfikować szablony jQTouch, ale na potrzeby tego przykładu pozostaniemy przy domyślnych klasach.

Pracę nad tą aplikacją zaczynamy od zera, więc pliki tworzone we wcześniejszych rozdziałach nie będą nam potrzebne. Przede wszystkim musisz utworzyć plik *index.html* i wpisać do niego kod HTML przedstawiony na listingu 4.1 (zawiera on główny ekran i informacje o aplikacji).

Listing 4.1. Kod HTML pliku *index.html*

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Kilo</title>
  </head>
  <body>
    <div id="home"> ❶
      <div class="toolbar"> ❷
        <h1>Kilo</h1>
      </div>
      <ul class="edgetoedge"> ❸
        <li class="arrow"><a href="#about">0 Kilo</a></li> ❹
      </ul>
    </div>
    <div id="about">
      <div class="toolbar">
        <h1>0 Kilo</h1>
        <a class="button back" href="#">Wstecz</a> ❺
      </div>
      <div>
        <p>Aplikacja Kilo pomaga stosować dowolną dietę.</p>
      </div>
    </div>
  </body>
</html>
```


W dokumencie HTML znajduje się tylko sekcja nagłówka z tytułem oraz dwa bloki `div`:

- ❶ Ten blok `div` (podobnie jak drugi — `about`) stanie się ekranem aplikacji. Wykorzystamy tu fakt, że oba bloki są bezpośrednimi potomkami elementu `body`.
- ❷ W każdym bloku `div` znajduje się element `div` klasy `toolbar` (jest ona zdefiniowana w arkuszu biblioteki `jQTouch`). Dzięki temu element będzie wyglądał jak standardowy pasek narzędzi w aplikacjach mobilnych.
- ❸ Ten znacznik nienumerowanej listy (`ul`) ma przypisaną klasę `edgetoedge`, dzięki której lista zostanie rozciągnięta w poziomie na całą szerokość ekranu.
- ❹ Ten element `li` zawiera odsyłacz (w atrybucie `href`) do ekranu *O Kilo*. Przypisanie do elementu `li` klasy `arrow` jest opcjonalne, ale jej zastosowanie spowoduje dodanie strzałki z prawej strony elementu.
- ❺ We wszystkich blokach z przypisaną klasą `toolbar` znajduje się jeden element `h1`, który będzie wyświetlany jako tytuł ekranu. Znacznik `a` ma przypisane klasy `button` oraz `back`, dzięki czemu będzie wyglądał jak przycisk „wstecz”.



Atrybut `href` przycisku „wstecz” został ustawiony na `#`. Standardowym zachowaniem przeglądarki po kliknięciu takiego odsyłacza jest przejście na początek bieżącego dokumentu, jednak w przypadku biblioteki `jQTouch` spowoduje to przejście do poprzedniego ekranu. Zamiast ogólnego łącza `#` możesz zastąpić konkretny punkt powrotu, na przykład `#home`, dzięki czemu wciśnięcie przycisku spowoduje wyświetlenie wskazanego ekranu, a nie poprzedniego.

Mamy już kod HTML podstawowej wersji strony, pora dołączyć `jQTouch`. Po zainstalowaniu `jQTouch` w tym samym katalogu co dokument HTML (patrz ramka „Instalowanie biblioteki `jQTouch`”) dodaj do sekcji `head` dokumentu kilka wierszy kodu (listing 4.2).

Instalowanie biblioteki `jQTouch`

W tym i wielu innych przykładach zaprezentowanych w książce będzie używana biblioteka `jQTouch`. Ze strony <http://www.jqtouch.com> pobierz plik ZIP i rozpakuj go. W miejscu, w którym znajduje się dokument HTML aplikacji *Kilo*, utwórz podkatalog `jqtouch` i umieść w nim katalogi `src` oraz `themes` z wypakowanego archiwum ZIP.

Listing 4.2. Aby skorzystać z *jQueryTouch*, w sekcji *head* dokumentu dopisz poniższe wiersze

```
<link type="text/css" rel="stylesheet" media="screen"
href="jqtouch/themes/css/jqtouch.css" /> ❶
<script type="text/javascript" src="jqtouch/src/lib/zepto.min.js"></script> ❷
<script type="text/javascript" src="jqtouch/src/jqtouch.min.js"></script> ❸
<script type="text/javascript"> ❹
    var jQT = $.jqTouch({
        icon: 'kilo.png'
    });
</script>
```

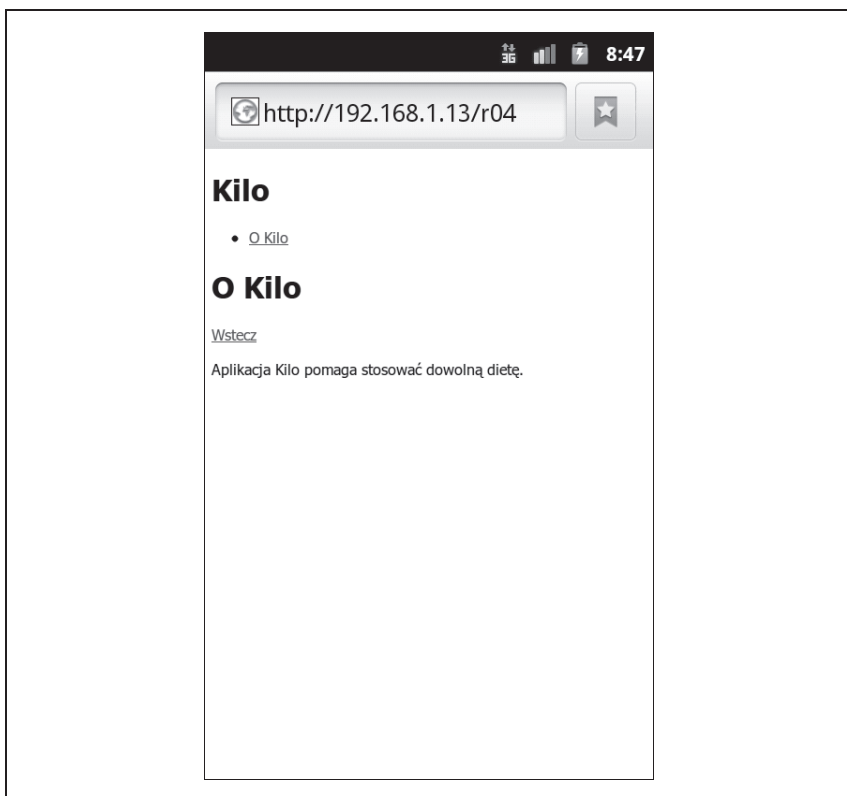
- ❶ W tym wierszu dołączamy arkusz stylów *jqtouch.css*, który jest domyślnym tematem graficznym stosowanym przez *jQueryTouch*. Są w nim zdefiniowane klasy, których używamy w dokumencie HTML. Nie powinienś wprowadzać zmian w tym arkuszu. Poza domyślnym tematem w *jQueryTouch* w wersji 1.0b4 dostępne są jeszcze dwa: *apple.css* oraz *vanilla.css*. Aby sprawdzić, jak wyglądają, zmodyfikuj w kodzie HTML nazwę dołączanego pliku CSS.
- ❷ *jQueryTouch* wymaga biblioteki *jQuery* bądź *Zepto* (obie zostały dołączone do paczki ZIP). Rozwiązaniem sugerowanym przez twórców biblioteki *jQueryTouch* jest zastosowanie *Zepto* (biblioteki bazującej na *jQuery*, która jest polecana dla aplikacji mobilnych ze względu na zminimalizowany rozmiar¹), jednak w omawianym przykładzie użyjemy biblioteki *jQuery* (z uwagi na nieobecność w *Zepto* potrzebnych nam funkcji). W takim przypadku konieczne jest dołączenie pliku *jqtouch-jquery.min.js*, który tworzy „most” między bibliotekami *jQueryTouch* i *jQuery*.
- ❸ W tym miejscu dołączamy skrypt biblioteki *jQueryTouch*. Bardzo ważna jest kolejność dołączania skryptów — najpierw *jQuery* (bądź *Zepto*), później *jQueryTouch*. W przeciwnym razie aplikacja by nie zadziałała.
- ❹ W bloku script tworzymy obiekt *jQueryTouch* i ustawiamy jego opcję *icon*.

Obiekt *jQueryTouch* udostępnia wiele właściwości, dzięki którym można spersonalizować zachowanie i wygląd aplikacji. Kilka z nich omówimy w dalszej części książki. Co prawda wszystkie te właściwości są opcjonalne, ale w większości przypadków i tak trzeba użyć co najmniej kilku.

Zastosowana w przykładzie właściwość *icon* odpowiada za ikonę wyświetlaną na ekranie domowym.

Na rysunku 4.1 widać aplikację przed zastosowaniem biblioteki *jQueryTouch*, a na rysunku 4.2 — po jej dołączeniu. Różnica jest niesamowita, ale naprawdę zadziwiające jest to, że wystarczyło kilka linii kodu, by uzyskać efekt przewijania stron. Robi wrażenie, prawda? A to dopiero początek!

¹ Szczegółowe informacje na temat biblioteki *Zepto* można znaleźć na stronie <http://zeptojs.com/> — przyp. tłum.



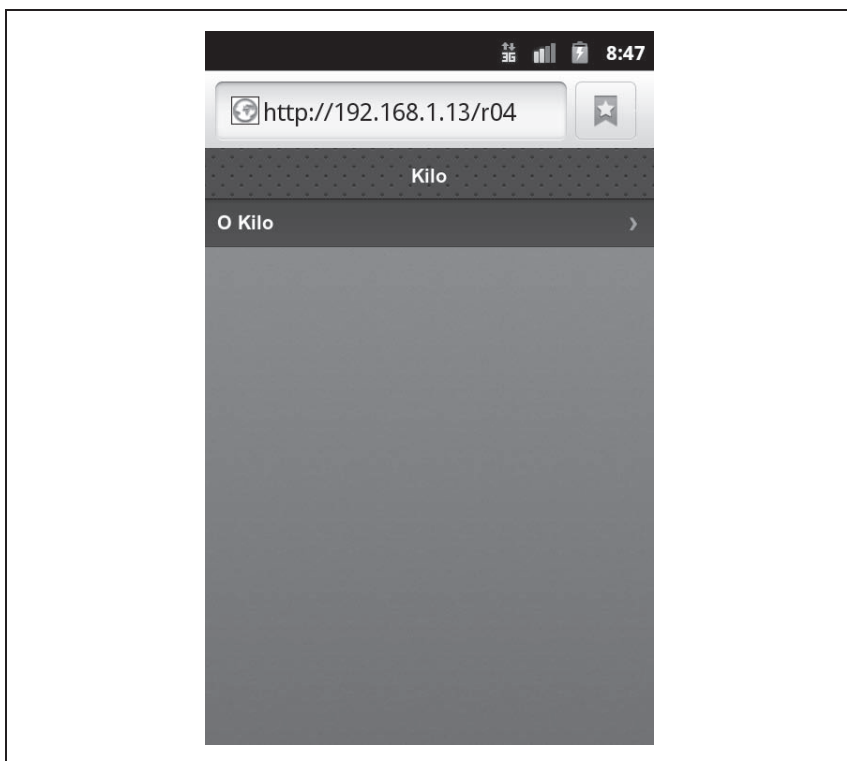
Rysunek 4.1. Aplikacja Kilo bez jQTouch...

Ekran Terminarz

Teraz zajmiemy się ekranem *Terminarz*. Ma się na nim znaleźć lista dni zaczynająca się dzisiaj, a kończąca pięć dni wcześniej. W dokumencie HTML po kodzie ekranu O Kilo (ale przed zamykającym znacznikiem `</body>`) wpisz kod z listingu 4.3. Trzeba będzie jeszcze uzupełnić nawigację na ekranie głównym, ale tym zajmiemy się za moment.

Listing 4.3. Kod HTML ekranu Terminarz

```
<div id="dates">
  <div class="toolbar">
    <h1>Terminarz</h1>
    <a class="button back" href="#">Wstecz</a>
```



Rysunek 4.2. ...i po dołączeniu biblioteki

```

</div>
<ul class="edgetoedge">
  <li class="arrow"><a id="0" href="#date">Dzisiaj</a></li>
  <li class="arrow"><a id="1" href="#date">Wczoraj</a></li>
  <li class="arrow"><a id="2" href="#date">2 dni temu</a></li>
  <li class="arrow"><a id="3" href="#date">3 dni temu</a></li>
  <li class="arrow"><a id="4" href="#date">4 dni temu</a></li>
  <li class="arrow"><a id="5" href="#date">5 dni temu</a></li>
</ul>
</div>

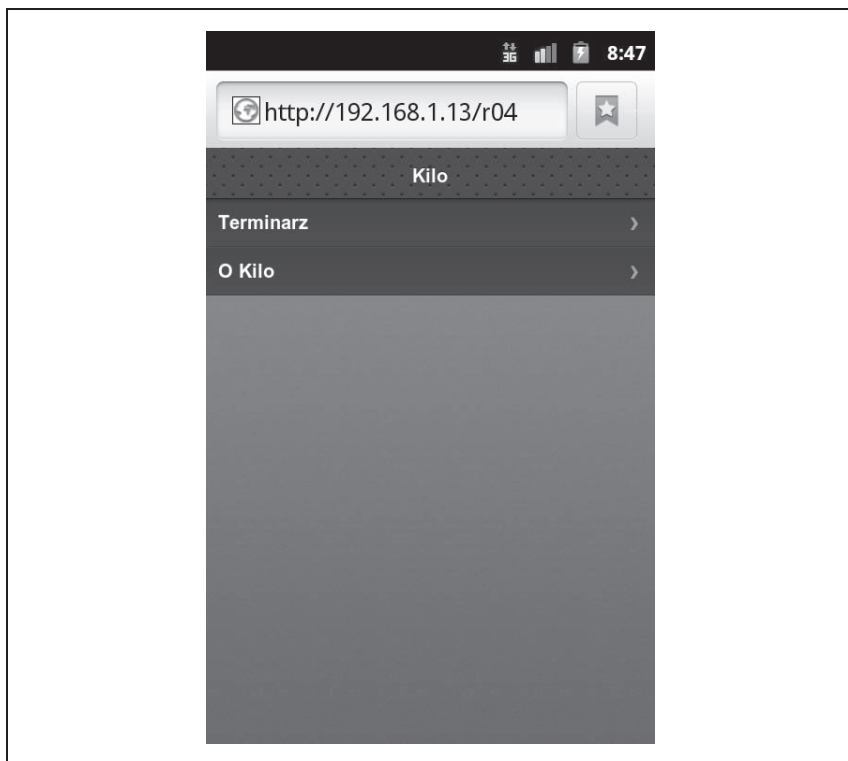
```

Podobnie jak na ekranie *O Kilo*, tu również ma się znaleźć pasek z tytułem i przycisk *Wstecz*. Dodatkowo umieściliśmy nienumerowaną listę odsyłaczy z przypisaną klasą *edgetoedge*. Zwróć uwagę, że wszystkie odsyłacze mają unikalne identyfikatory (od 0 do 5), ale tę samą wartość atrybutu *href* (*#date*) — dlatego tak jest, dowiesz się niebawem.

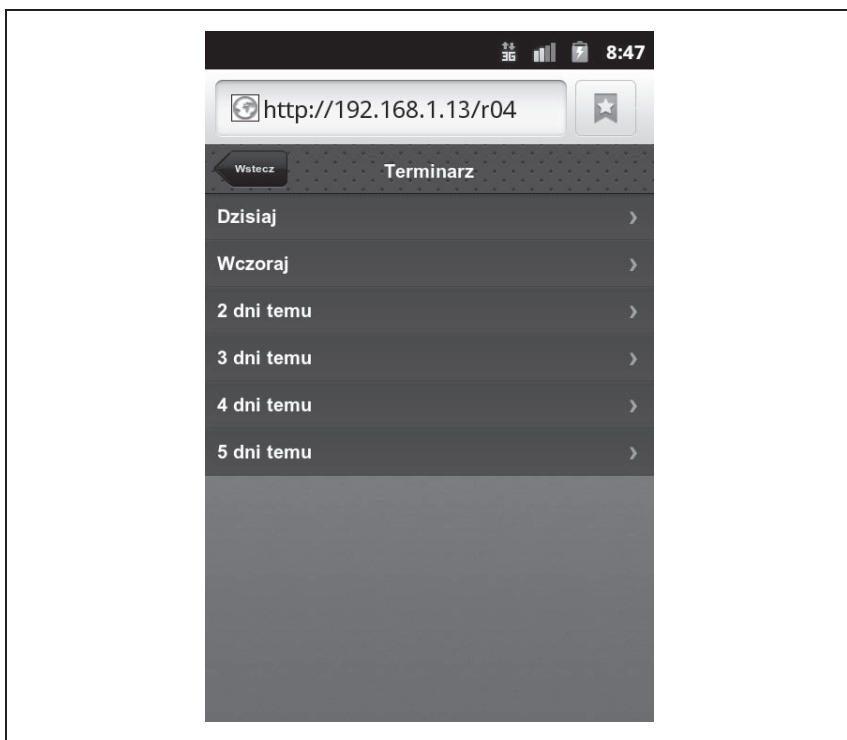
Następnie w kodzie głównego ekranu musimy umieścić odsyłacz do ekranu *Terminarz* (nowy kod jest wyróżniony pogrubieniem):

```
<div id="home">
  <div class="toolbar">
    <h1>Kilo</h1>
  </div>
  <ul class="edgetoedge">
    <li class="arrow"><a href="#dates">Terminarz</a></li>
    <li class="arrow"><a href="#about">0 Kilo</a></li>
  </ul>
</div>
```

W ten oto sposób dodaliśmy do aplikacji kolejny ekran (rysunek 4.3). Po kliknięciu odsyłacza *Terminarz* zostanie wyświetlony ekran *Terminarz* (rysunek 4.4). Odsyłacze znajdujące się na tym ekranie na razie nie działają. Poradzimy sobie z tym problemem, tworząc ekran wyświetlający dni (czyli ekran *Dzień*).



Rysunek 4.3. Na głównym ekranie pojawił się odsyłacz *Terminarz*



Rysunek 4.4. Ekran Terminarz zawiera pasek z tytułem i przyciskiem Wstecz oraz listę odsyłaczy do kolejnych dni

Ekran Dzień

Ekran *Dzień* przypomina poprzednie ekrany z kilkoma drobnymi różnicami (spójrz na listing 4.4). Dopisz przedstawiony kod HTML pod kodem ekranu *Dates*, ale przed znacznikiem zamykającym `</body>`.

Listing 4.4. Kod HTML ekranu *Dzień*

```
<div id="date">
  <div class="toolbar">
    <h1>Dzień</h1>
    <a class="button back" href="#">Wstecz</a>
    <a class="button slideup" href="#createEntry">+</a> ❶
  </div>
  <ul class="edgetoedge">
    <li id="entryTemplate" class="entry" style="display:none"> ❷
```

```

        <span class="label">Etykieta</span>
        <span class="calories">000</span>
        <span class="delete">Usuń</span>
    </li>
</ul>
</div>

```

- 1 Na pasku umieściliśmy dodatkowy przycisk. Po jego kliknięciu zostanie wyświetlony ekran *Nowy wpis* (którego jeszcze nie przygotowaliśmy). Odsyłaczowi jest przypisana klasa `slideup`, dzięki której podczas zmiany ekranów zostanie zastosowany efekt przewijania w górę, a nie, jak dotąd, z lewej lub z prawej strony.
- 2 Kolejnym elementem odróżniającym ten ekran od pozostałych jest ustawienie stylu `display:none` w pozycji listy, co powoduje jej ukrycie.

Jak wkrótce zobaczysz, użyjemy tej niewidocznej pozycji listy jako szablonu dla wyświetlanych wpisów. Na razie nie ma jednak żadnych wpisów, więc lista jest pusta.

Po dodaniu kodu kliknięcie dowolnej pozycji na ekranie *Terminarz* spowoduje wyświetlenie ekranu *Dzień* (rysunek 4.5).

Ekran Nowy wpis

Na listingu 4.5 znajduje się kod ekranu *Nowy wpis*. Umieść go na końcu pliku *index.html* (oczywiście przed zamykającym znacznikiem `</body>`).

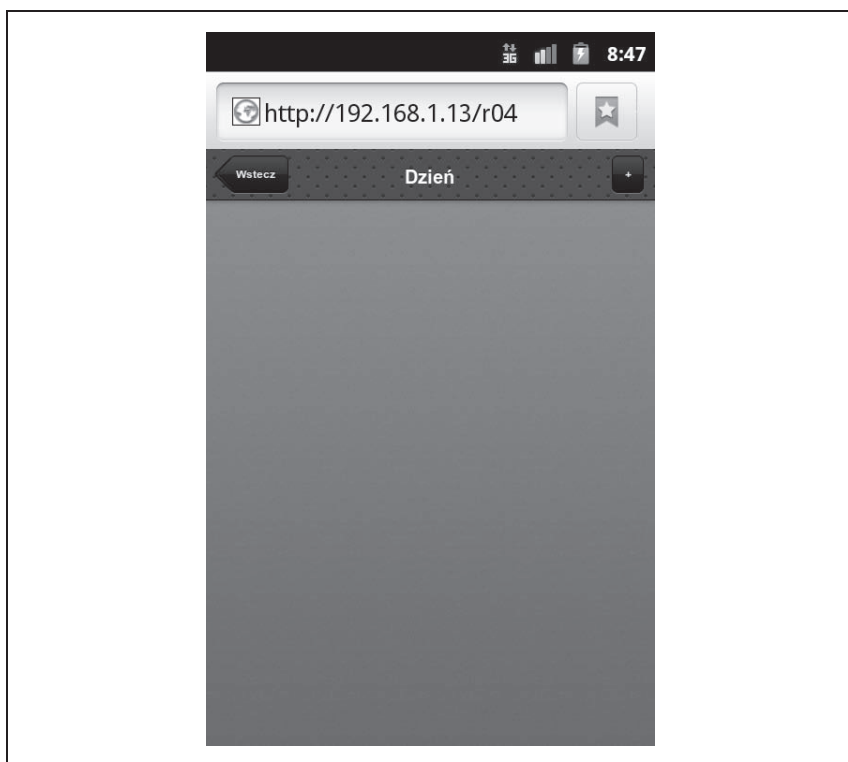
Listing 4.5. Kod HTML ekranu *Nowy wpis*

```

<div id="createEntry">
  <div class="toolbar">
    <h1>Nowy wpis</h1>
    <a class="button cancel" href="#">Anuluj</a> ❶
  </div>
  <form method="post"> ❷
    <ul class="rounded">
      <li><input type="text" placeholder="Posiłek" name="food" id="food"
        ↳autocapitalize="off" autocorrect="off" autocomplete="off" /></li>
      <li><input type="text" placeholder="Kalorie" name="calories" id="
        ↳calories" autocapitalize="off" autocorrect="off" autocomplete="off" /></li>
      <li><input type="submit" class="submit" name="action" value="Zapisz"
        ↳/></li> ❸
    </ul>
  </form>
</div>

```

- 1 Pierwszym, na co należy zwrócić uwagę na ekranie *Nowy wpis*, jest inna nazwa przycisku — zamiast *Wstecz* jest *Anuluj*.



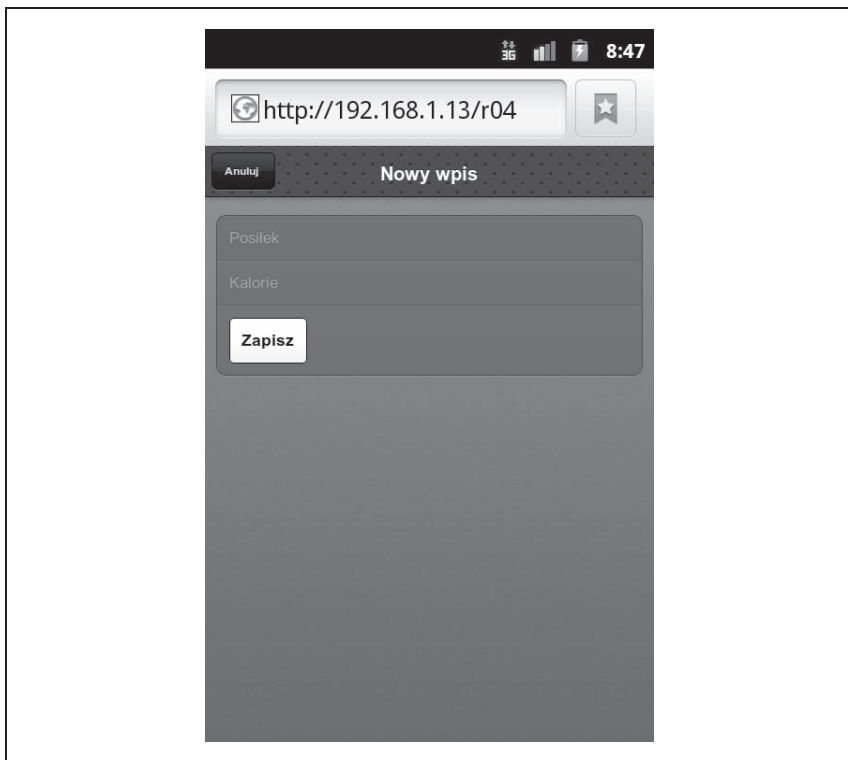
Rysunek 4.5. Na razie ekran Dzień jest pusty (poza paskiem z tytułem i dwoma przyciskami)



Przycisk *Anuluj* (klasa `cancel`) zachowuje się w `jQueryTouch` tak samo jak przycisk *Wstecz* (klasa `back`), czyli usuwa bieżący ekran z zastosowaniem animacji odwrotnej do tej, która jest stosowana podczas pojawiania się ekranu. Można jednak zauważyć różnicę w wyglądzie — przycisk *Anuluj* nie ma charakterystycznego kształtu strzałki skierowanej w lewo.

Zastosowaliśmy tu przycisk *Anuluj* ze względu na to, że ekran *Nowy wpis* jest wyświetlany z efektem wjazdu z góry, więc jego ukrywaniu powinien towarzyszyć ekran zjazdu w dół. Gdybyśmy użyli przycisku *Wstecz*, doprowadzilibyśmy do niekonsekwencji, ponieważ po naciśnięciu przycisku ze strzałką skierowaną w lewo ekran zjechałby w dół (a to mogłoby zmylić użytkowników).

- ❷ Formularz zawiera nienumerowaną (wypunktowaną) listę trzech elementów: dwóch pól tekstowych i przycisku potwierdzenia. Dzięki osadzeniu kontrolek formularza w elemencie `li` zostaną do nich zastosowane style jQueryTouch (rysunek 4.6).



Rysunek 4.6. Do elementów formularza zostały zastosowane style z jQueryTouch

W polach tekstowych umieściliśmy kilka atrybutów:

`type="text"`

Definiuje kontrolkę formularza jako jednowierszowe pole tekstowe.

`placeholder`

Łańcuch tekstowy wyświetlany w pustym polu formularza.

`name`

Nazwa pola, która po przesłaniu formularza zostanie powiązana z wartością wpisaną przez użytkownika.

id

Unikalny identyfikator elementu w kontekście całej strony.

autocapitalize

Steruje automatyczną zamianą wprowadzanych liter na wielkie (działa tylko w przeglądarce Mobile Safari w iOS).

autocorrect

Steruje funkcją sprawdzania pisowni w przeglądarce Mobile Safari w iOS (nie działa na Androidzie).

autocomplete

Steruje funkcją automatycznego uzupełniania w przeglądarce Mobile Safari w iOS (nie działa na Androidzie).

- ❸ Atrybut `class` przycisku przesyłającego formularz wymaga dokładniejszego wyjaśnienia. Przeglądarka Androida wyświetla klawiaturę ekranową, gdy kursor znajdzie się w polu formularza. W prawym dolnym rogu klawiatury znajduje się przycisk *Idź*, który służy do przesyłania formularza. W sytuacji, gdy przechwytujemy funkcję przesyłania (a tak się dzieje w omawianym przykładzie), wciśnięcie tego przycisku nie spowoduje usunięcia kursora z aktywnego pola, w związku z czym klawiatura nie zniknie z ekranu. Biblioteka `jQueryTouch` umożliwia rozwiązanie tego problemu poprzez automatyczne usunięcie kursora z pola po zatwierdzeniu formularza. Aby skorzystać z tej funkcji, do przycisku przesyłającego formularz trzeba przypisać klasę `submit`.

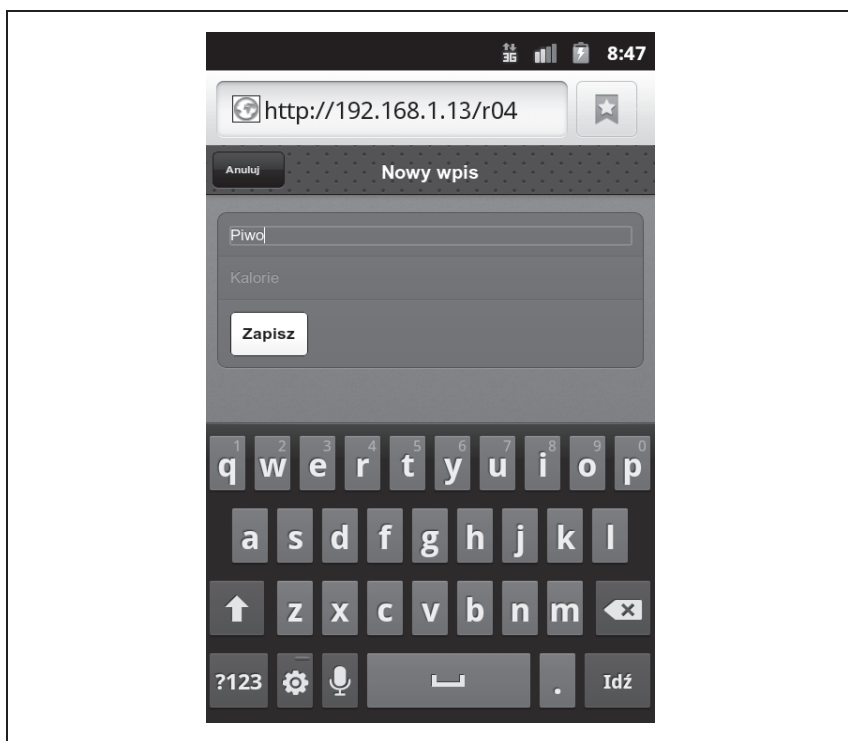
Na rysunku 4.7 widać działający formularz *Nowy wpis*. Na razie nie zajęliśmy się faktycznym zapisywaniem wprowadzonych danych — omówimy to w rozdziale 5.

Ekran Ustawienia

Do tej pory nie utworzyliśmy przycisku umożliwiającego przejście do ekranu Ustawienia. Zrobimy to teraz, dodając przycisk do paska tytułowego na głównym ekranie. W tym celu wystarczy dodać jedną linię kodu HTML (na listingu wyróżnioną pogrubieniem):

```
</head>
<body>
  <div id="home">
    <div class="toolbar">
      <h1>Kilo</h1>

    <a class="button flip" href="#settings">Ustawienia</a> ❶
    </div>
    <ul class="edgetoedge">
```



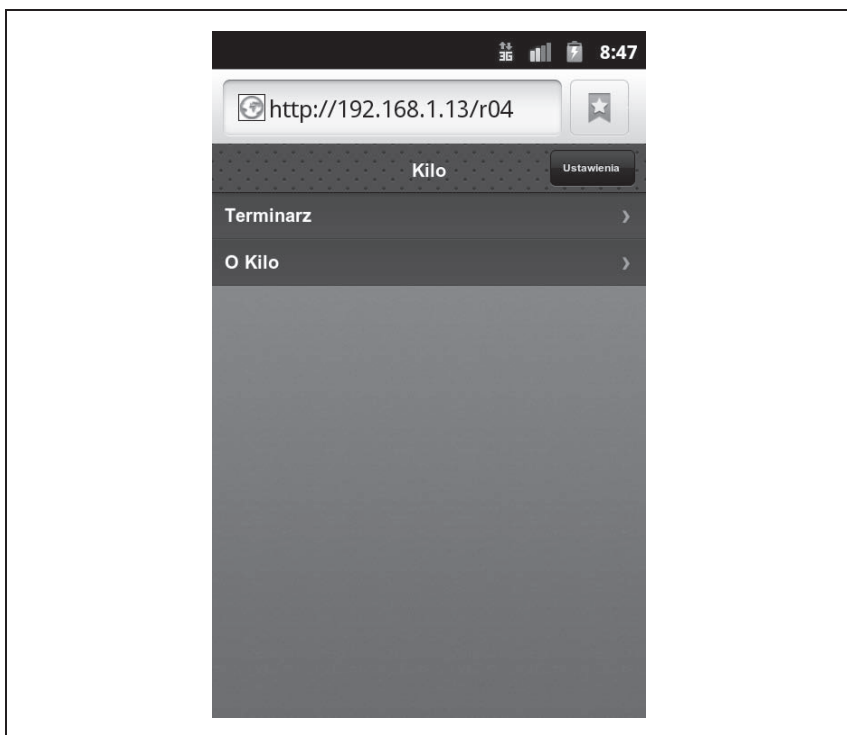
Rysunek 4.7. Klawiatura ekranowa w formularzu Nowy wpis

```
<li class="arrow"><a href="#dates">Terminarz</a></li>
<li class="arrow"><a href="#about">0 Kilo</a></li>
</ul>
</div>
...pozostała część dokumentu HTML nie została zamieszczona...
```

- 1 W tym miejscu dodajemy przycisk (rysunek 4.8). Zwróć uwagę, że do odsyłacza przypisaliśmy klasę flip. Dzięki temu przejściu z ekranu głównego do ekranu *Ustawienia* będzie towarzyszył efekt obrócenia wokół osi pionowej. Dodatkowo, dla wzmocnienia efektu, strona jest lekko skalowana podczas animacji. Ładnie, prawda?



Niestety ze wsparciem dla trójwymiarowych animacji w przeglądarkach urządzeń mobilnych bywa różnie (dotyczy to również Androida). Z tego względu na niektórych urządzeniach wszelkie obroty, przejścia i inne animacje 3D mogą zostać zastąpione dwuwymiarowymi odpowiednikami.



Rysunek 4.8. Przycisk Ustawienia umieszczony w pasku na ekranie głównym

Kod ekranu *Ustawienia* (listing 4.6) jest bardzo podobny do tego, czym zajmowaliśmy się przed chwilą, czyli ekranu *Nowy wpis*. Znajduje się tu jedno pole tekstowe więcej, niektóre atrybuty są pominięte, niektóre mają inne wartości, ale sama koncepcja pozostaje podobna. Do dokumentu HTML dopisz kod z poniższego listingu, tak jak to robiłeś dla innych ekranów.

Listing 4.6. Kod HTML ekranu *Ustawienia*

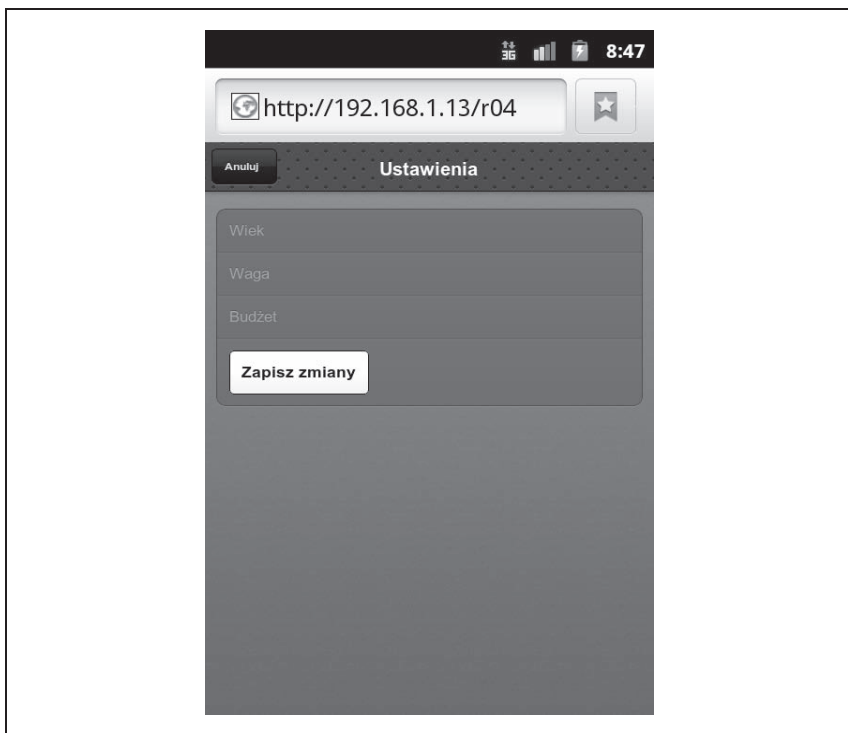
```
<div id="settings">
  <div class="toolbar">
    <h1>Ustawienia</h1>
    <a class="button cancel" href="#">Anuluj</a>
  </div>
  <form method="post">
    <ul class="rounded">
      <li><input placeholder="Wiek" type="text" name="age" id="age" /></li>
      <li><input placeholder="Waga" type="text" name="weight" id="weight" /></li>
      <li><input placeholder="Limit kalorii" type="text" name="budget"
        ↪ id="budget" /></li>
    </ul>
  </form>
</div>
```

```

<li><input type="submit" class="submit" name="action" value="Zapisz zmiany"
  ↳></li>
</ul>
</form>
</div>

```

Podobnie jak w przypadku formularza *Nowy wpis*, w formularzu *Ustawienia* też nie ma jeszcze możliwości zapisania wprowadzonych danych (patrz rysunek 4.9). Problem ten rozwiążemy w kolejnym rozdziale.



Rysunek 4.9. Ekran *Ustawienia*

Integracja całości

I to by było na tyle. Wystarczyło napisać niewiele ponad sto linii kodu HTML, by otrzymać pięcioekranową aplikację wyglądającą i zachowującą się jak natywna. Zastosowaliśmy w niej trzy różne efekty przejścia między ekranami. Pełny listing aplikacji znajdziesz na listingu 4.7. Nie wygląda to źle, prawda?

Listing 4.7. Kompletny listing pięcioekranowej aplikacji

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Kilo</title>
    <link type="text/css" rel="stylesheet" media="screen"
href="jqtouch/themes/css/jqtouch.css" />
    <script type="text/javascript" src="jqtouch/src/lib/jquery-
1.7.min.js"></script>
    <script type="text/javascript" src="jqtouch/src/jqtouch-
jquery.min.js"></script>
    <script type="text/javascript" src="jqtouch/src/jqtouch.min.js"></script>
    <script type="text/javascript">
      var jQT = $.jQTouch({
        icon: 'kilo.png'
      });
    </script>
  </head>
  <body>
    <div id="home">
      <div class="toolbar">
        <h1>Kilo</h1>
        <a class="button flip" href="#settings">Ustawienia</a>
      </div>
      <ul class="edgetoedge">
        <li class="arrow"><a href="#dates">Terminarz</a></li>
        <li class="arrow"><a href="#about">0 Kilo</a></li>
      </ul>
    </div>
    <div id="about">
      <div class="toolbar">
        <h1>0 Kilo</h1>
        <a class="button back" href="#">Wstecz</a>
      </div>
      <div>
        <p>Aplikacja Kilo pomaga stosować dowolną dietę.</p>
      </div>
    </div>
    <div id="dates">
      <div class="toolbar">
        <h1>Terminarz</h1>
        <a class="button back" href="#">Wstecz</a>
      </div>
      <ul class="edgetoedge">
        <li class="arrow"><a id="0" href="#date">Dzisiaj</a></li>
        <li class="arrow"><a id="1" href="#date">Wczoraj</a></li>
        <li class="arrow"><a id="2" href="#date">2 dni temu</a></li>
        <li class="arrow"><a id="3" href="#date">3 dni temu</a></li>
        <li class="arrow"><a id="4" href="#date">4 dni temu</a></li>
        <li class="arrow"><a id="5" href="#date">5 dni temu</a></li>
      </ul>
    </div>
  </body>
```

```

<div id="date">
  <div class="toolbar">
    <h1>Dzień</h1>
    <a class="button back" href="#">Wstecz</a>
    <a class="button slideup" href="#createEntry">+</a>
  </div>
  <ul class="edgetoedge">
    <li id="entryTemplate" class="entry" style="display:none">
      <span class="label">Etykieta</span>
      <span class="calories">000</span>
      <span class="delete">Usuń</span>
    </li>
  </ul>
</div>
<div id="createEntry">
  <div class="toolbar">
    <h1>Nowy wpis</h1>
    <a class="button cancel" href="#">Anuluj</a>
  </div>
  <form method="post">
    <ul class="rounded">
      <li><input type="text" placeholder="Posiłek" name="food" id="food"
        ↪autocapitalize="off" autocorrect="off" autocomplete="off" /></li>
      <li><input type="text" placeholder="Kalorie" name="calories"
        ↪id="calories" autocapitalize="off" autocorrect="off"
        ↪autocomplete="off" /></li>
      <li><input type="submit" class="submit" name="action"
        ↪value="Zapisz" /></li>
    </ul>
  </form>
</div>
<div id="settings">
  <div class="toolbar">
    <h1>Ustawienia</h1>
    <a class="button cancel" href="#">Anuluj</a>
  </div>
  <form method="post">
    <ul class="rounded">
      <li><input placeholder="Wiek" type="text" name="age" id="age" /></li>
      <li><input placeholder="Waga" type="text" name="weight" id="weight"
        ↪/></li>
      <li><input placeholder="Limit kalorii" type="text" name="budget"
        ↪id="budget"
        ↪/></li>
      <li><input type="submit" class="submit" name="action" value="Zapisz
        ↪zmiany" /></li>
    </ul>
  </form>
</div>
</body>
</html>

```

Dostosowywanie biblioteki jQueryTouch

Domyślne zachowanie jQueryTouch możesz dostosować do własnych potrzeb, przekazując w konstruktorze zestaw właściwości. W omawianym przykładzie skorzystaliśmy tylko z jednej — `icon` — ale dostępnych jest znacznie więcej. Z całą pewnością warto się z nimi zapoznać (tabela 4.1).

Tabela 4.1. Opcje dostosowywania jQueryTouch

Właściwość	Domyślna wartość	Oczekiwane wartości	Uwagi
<code>addGlossToIcon</code>	<code>true</code>	<code>true</code> albo <code>false</code>	Jeśli jest ustawione na <code>true</code> , do ikon na ekranie domowym zostanie dodana poświata (działa w iOS, ale nie w Androidzie)
<code>backSelector</code>	<code>'.back, .cancel, .goback'</code>	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Definiuje elementy, które po kliknięciu (bądź puknięciu) mają wyzwalać zachowanie powrotu polegające na usunięciu bieżącego ekranu z zastosowaniem przeciwnej animacji i usunięcie go z historii
<code>cacheGetRequests</code>	<code>true</code>	<code>true</code> albo <code>false</code>	Jeśli jest ustawione na <code>true</code> , wszystkie żądania GET są cachowane, więc kolejne wywołania korzystają z pobranych już danych
<code>cubeSelector</code>	<code>'.cube'</code>	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Definiuje elementy, które wyzwalają animację „cube” podczas zmiany ekranów
<code>dissolveSelector</code>	<code>'.dissolve'</code>	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Definiuje elementy, które wyzwalają animację „dissolve” podczas zmiany ekranów
<code>fadeSelector</code>	<code>'.fade'</code>	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Definiuje elementy, które wyzwalają animację „fade” podczas zmiany ekranów
<code>fixedViewport</code>	<code>true</code>	<code>true</code> albo <code>false</code>	Ustawione na <code>true</code> blokuje możliwość skalowania strony
<code>flipSelector</code>	<code>'.flip'</code>	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Definiuje elementy, które wyzwalają animację „flip” podczas zmiany ekranów

Tabela 4.1. Opcje dostosowywania jQuery — ciąg dalszy

Właściwość	Domyślna wartość	Oczekiwane wartości	Uwagi
formSelector	'form'	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Definiuje elementy, które powinny odbierać zdarzenie onSubmit
fullScreen	true	true albo false	Działa tylko w iOS. Jeśli jest ustawione na true i aplikacja jest uruchamiana z ekranu domowego, zostanie otwarta w trybie pełnoekranowym. Ta opcja nie działa, jeśli aplikacja jest uruchamiana z poziomu przeglądarki Mobile Safari
fullScreenClass	'fullscreen'	łańcuch tekstowy	Działa tylko w iOS. Nazwa klasy, która zostanie zastosowana do sekcji body po uruchomieniu aplikacji w trybie pełnoekranowym. Dzięki temu można zdefiniować style stosowane tylko w tym przypadku
icon	null	null bądź względna lub bezwzględna ścieżka do pliku PNG	Ikona aplikacji umieszczana na ekranie domowym. Ten obraz zostanie wyświetlony po dodaniu do ekranu domowego zakładki do aplikacji
popSelector	' .pop'	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Definiuje elementy, które wyzwalają animację „pop” podczas zmiany ekranów
preloadImages	false	Tablica ścieżek do obrazów	Definiuje obrazy, które mają zostać pobrane przed załadowaniem strony (na przykład: ['images/link_over.png', 'images/link_select.png'])
slideInSelector	'ul li a'	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Definiuje elementy, które wyzwalają animację „slide in” podczas zmiany ekranów
slideUpSelector	' .slideUp'	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Definiuje elementy, które wyzwalają animację „slide up” podczas zmiany ekranów

Tabela 4.1. Opcje dostosowywania jQueryTouch — ciąg dalszy

Właściwość	Domyślna wartość	Oczekiwane wartości	Uwagi
startupScreen	Null	null bądź względna lub bezwzględna ścieżka do pliku obrazu	Działa tylko w iOS. Względna lub bezwzględna ścieżka do pliku z obrazem startowym (320×460 pikseli) dla pełnoekranowych aplikacji. Jeśli opcja statusBar jest ustawiona na black-translucent, obraz powinien mieć wymiary 320×480 pikseli.
statusBar	'default'	default, black-translucent, black	Działa tylko w iOS. Definiuje wygląd 20-pikselowego paska stanu znajdującego się na górze okna aplikacji działającej w trybie pełnoekranowym
submitSelector	'.submit'	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Selektor elementu, który ma przysyłać nadrzędny formularz (i zamykać panel klawiatury ekranowej)
swapSelector	'.swap'	Dowolny poprawny selektor CSS; poszczególne wartości oddzielane przecinkami	Definiuje elementy, które wyzwalają animację „swap” podczas zmiany ekranów
useAnimations	true	true albo false	Ustawienie na false powoduje wyłączenie wszystkich animacji

Czego się nauczyłeś?

Z tego rozdziału dowiedziałeś się, jak dzięki bibliotece jQueryTouch dodać do aplikacji internetowej animacje charakterystyczne dla aplikacji natywnych. W kolejnym rozdziale nauczysz się korzystać z mechanizmu lokalnego składowania danych oraz poznasz możliwości baz danych działających po stronie klienta, by zapisywać dane wymagane przez tworzone aplikacje.

Skorowidz

A

A List Apart, 72
adres URL, 44
ADT, 144, 145
Ajax, 49
akapit, 16
akcelerometr, 141, 162
Android, 14, 15, 36, 142
Android Developer Tools, *Patrz:* ADT
Android SDK, 143, 144, 173
animacja
 cube, 92
 dissolve, 92
 fade, 92
 flip, 92
 pop, 93
 przeciwna, 92
 slide in, 93
 slide up, 93
 swap, 94
 trójwymiarowa, 87
 wyłączenie, 94
aplikacja
 debugowanie, *Patrz:* debugowanie
 dystrybucja, 167, 172
 działająca offline, 121, 126, 132, 141
 internetowa, 13, 14, 141, 142
 kompilowanie, *Patrz:*
 kompilowanie
 natywna, 13, 14, 27, 141, 142
 numer wersji, 168
 podpisywanie, 169
 pracujących offline, 136
 ustawienia, 97

 z podwyższonymi
 uprawnieniami, 142
Asynchronous JavaScript and XML,
 Patrz: Ajax
atrybut, 17
 autocapitalize, 86
 autocomplete, 86
 autocorrect, 86
 class, 20, 86
 href, 77
 id, 20, 86
 name, 85
 onclick, 25
 placeholder, 85
 type, 85

B

bada, 142
baza danych, 104
 maksymalny rozmiar, 105
 nazwa, 105
 SQLite, 102
 wersja, 105
biała lista, 126
biblioteka
 cordova.js, 150
 jQuery, *Patrz:* jQuery
 jQuery, *Patrz:* jQuery
 wurfl-php, 177, 178, 179
 Zepto, *Patrz:* Zepto
Bit Rate Throttling Media Services,
 58
BlackBerry, 142
blok div, 31

błąd, 108, 118, 137, 139, 153
 CONSTRAINT_ERR, 119
 DATABASE_ERR, 119
 QUOTA_ERR, 119
 SYNTAX_ERR, 119
 TIMEOUT_ERR, 119
 TOO_LARGE_ERR, 119
 UNKNOWN_ERR, 119
 VERSION_ERR, 119
body, *Patrz:* ciało

C

Cascading Style Sheet, *Patrz:* CSS
chrome, 19
ciało, 17
ciasteczka, 95
cień, 40
Crockford Douglas, 55
CSS, 15, 19, 21, 27, 76
 Sprite, 72

D

data, 100, 101
debugowanie, 109, 136
definicja
 max-width, 34
 min-width, 34
deklaracja, 19
Document Object Model, *Patrz:* DOM
DOM, 25
 manipulowanie strukturą, 24
domyślne zachowanie łączy, 54
dyrektywa AddType, 123
dziecko, 16
dźwięk, 141, 151

E

E Text Editor, 18
Eclipse, 143
edytor tekstu, 18

ekran
 domowy, 72, 93, 141
 przewijanie, 75
element
 html, 17
 potomny, 16
 title, 17
 ukrywanie, 42
 ul, 37, 41, 42, 46
emacs, 18
emfaza, 17

F

font
 Droid, 37
 Roboto, 37
formularz, 85, 86, 97, 106
 kontrolka, 85
 nadrzędny, 94
funkcja
 alert, 153
 ALTER TABLE, 155
 bind, 101
 errorHandler, 109
 executeSql, 110
 goBack, 98
 gt, 112
 haszująca, 135
 header, 133
 hijackLinks, 59
 jako parametr innej funkcji, 55
 loadPage, 62
 obsługi błędów, 109
 openDatabase, 104, 105
 przesyłania, 86
 setTimeout, 58
 transaction, 110
 url.match, 64
 uzupełniania automatycznego, 86
 val, 97

G

gedit, 18
geolokalizacja, 155
Google Chart Tools, 172
Google Play, 170
gradient, 40

H

head, *Patrz:* nagłówek
hiperłącze, *Patrz:* odsyłacz
HTML, 15

I

IDE, 143
identyfikator, 21, 37, 86
ikona, 72
instalowanie, 14
instrukcja sterująca, 23
Integrated Development
Environment, *Patrz:* IDE
interfejs
 API, 102, 118, 142
 API WURFL, 34
 użytkownika, 13, 15, 36
Internet Explorer, 34, 45
iPhone, 142

J

JavaScript, 13, 14, 24, 49, 55, 137
język
 skryptowy, 23, 27
jQuery, 75, 77, 78, 86, 92, 99
 konstruktor, 97
jQuery, 25, 26, 45, 46, 57, 59, 70, 75,
 78, 97, 112

K

kamera, 141
Kanedy David, 75
kaskadowy arkusz stylów, *Patrz:* CSS
klasa, 21
 arrow, 77
 back, 77, 84
 button, 77
 cancel, 84
 clicked, 70, 71
 edgetoedge, 77
 flip, 87
 hide, 42, 46
 slideup, 83
 submit, 86
 toolbar, 77
klawiatura
 ekranowa, 86, 94
kliknięcie, 71
klucz, 169
kod QR, 172
kolor, 40
kompilowanie, 14, 169
konsola
 JavaScriptu, 137
 logów, 136
kotwica, 17
krawędź szerokość, 44
kwerenda, 105, 152
 CREATE TABLE, 105

L

link, *Patrz:* odsyłacz
Linux, 18, 29, 58, 131
lista, 16
 biała, 126
 nienumerowana, 77, 85
 nieuporządkowana, 16
 wypunktowana, *Patrz:* lista
 nienumerowana
localStorage, 95, 96, 97

M

magazyn kluczy, 169
manipulowanie strukturą DOM,
 Patrz: DOM manipulowanie
 strukturą
Marcotte Ethan, 32
Microsoft Word, 18
model freemium, 15

N

nagłówek, 16, 17, 37
nano, 18
Nitobi, 141
Notepad2, 18

O

obiekt
 jQuery, 78
 klikalny, 69
 WebView, 149
 window, 96
 zdarzenia, 55
obraz
 startowy, 94
 wczesne ładowanie, 70
obsługa błędów, 108, 109, 137, 139,
 153
odsyłacz, 17, 52, 77
 lokalny, 64
 przechwytywanie, 64
okno modalne, 153
operator logiczny, 61
ostrzeżenie, 151, 153

P

padding, 38
para klucz-wartość, 95, 96
pasek stanu, 94
pętla for, 23

PhoneGap, 15, 141, 146, 151
PhoneGap Build, 143
PHP, 27, 122, 131
plik
 htaccess, 124
 manifestu, 121, 122, 125, 130, 132,
 139, 167
 obrazu, 44
polecenie ipfw, 58
poświęta, 92
potomek, 77
programowanie, 14
przeglądarka, 14, 15, 96
 Chrome, 19
 desktopowa, 35, 45
 Internet Explorer,
 Patrz: Internet Explorer
 Mobile Safari, 86, 93
 pasek tytułowy, 17
 wykrywanie, 177
przesunięcie
 w pionie, 40
 w poziomie, 40
przycisk, 42, 43, 46, 71, 83
 wstecz, 64, 77
pseudoklasa, 42
 active, 70
 hover, 70

Q

Qt, 142

R

Responsive Web Design, 32
RhoMobile, 142
rodzic, 16
rozmycie, 40

S

sekcja

- body, 17, 19, 93

- CACHE MANIFEST, 127

- FALLBACK, 126, 127, 130

- head, 21

- NETWORK, 126, 130

selektor, 19, 20, 21, 37, 42

serwer, 27

- Apache, 29, 122, 124

- IIS, 123, 131

- lokalny, 28

sessionStorage, 95, 96, 100

słowo kluczowe, 20

- this, 101, 105

SQL, 103

strona

- automatyczne przewijanie

 - do początku, 62

- główna witryny, 54

- przeladowywanie, 49

- skalowanie, 35, 92

- szerokość, 35, 45

- tytuł, 59, 61

Sublime Text, 18

Symbian, 142

symulowanie rzeczywistego

- obciążenia sieci, 58

system operacyjny, 177

T

tabela, 103, 104

tablica, 23, 66

- wartości statusu trybu offline, 138

Text Wrangler, 18

TextEdit, 18

TextMate, 18

Titanium Mobile, 142

transakcja, 103, 105

tryb

- offline, 15

- pełnoekranowy, 93, 94

V

vi, 18

W

wartość, 19

Web SQL Database, 95, 102, 118

Web Storage, 95

webOS, 142

wibracja, 141, 151, 155

Windows Phone, 142

Wireless Universal Resource File,

- Patrz:* WURFL

właściwość, 19

- addGlossToIcon, 92

- backSelector, 92

- border-width, 43

- cacheGetRequests, 92

- cubeSelector, 92

- dissolveSelector, 92

- fadeSelector, 92

- fixedViewport, 92

- flipSelector, 92

- formSelector, 93

- fullScreen, 93

- fullScreenClass, 93

- icon, 78, 93

- innerWidth, 45

- popSelector, 93

- preloadImages, 93

- slideInSelector, 93

- slideupSelector, 93

- startupScreen, 94

- statusBar, 94

- submitSelector, 94

- swapSelector, 94

- text-shadow, 40

- useAnimations, 94

- webkit-border-image, 43, 72

- webkit-border-radius, 41

- webkit-gradient, 40

WordPad, 18

wskaźnik postępu, 56, 57, 58, 60
WURFL, 34, 177
wyrażenie regularne, 64

Z

zakładka, 72, 93
zaokrąglone wierzchołki, 40
zasada DRY, 32, 33
zdarzenie
 click, 101
 kliknięcie, *Patrz:* kliknięcie
 onsubmit, 93
 pageAnimationStart, 99
 przesyłania formularza, 106
 ready, 45, 46
 touchend, 101
Zepro, 78
Zepto, 78
zintegrowane środowisko
 programistyczne, *Patrz:* IDE

zmienna, 23
 globalna, 66, 104
znacznik, 21
 a, 17, 42
 blokowy, 16
 h1, 16
 liniowy, 17
 listy nienumerowanej, 77
 listy nieuporządkowanej, 16
 meta viewport, 35
 otwierający, 16, 17
 p, 16
 zamykający, 16
znak ucieczki, 134

Ż

żądanie GET, 92

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA



Helion SA

Android

Tworzenie aplikacji w oparciu o HTML, CSS i JavaScript



Jeśli znasz HTML, CSS i JavaScript, jesteś gotowy na tworzenie aplikacji dla Androida. Ten praktyczny podręcznik pomoże Ci wejść w świat aplikacji przeznaczonych dla platformy Android, tworzonych z wykorzystaniem otwartych standardów internetowych – przede wszystkim HTML5 – oraz dodatkowych narzędzi, takich jak choćby PhoneGap.

Dowiesz się, jak budować aplikacje internetowe i jak wykorzystać bezpłatny framework Adobe PhoneGap do przekonwertowania ich na natywne aplikacje dla Androida. Na własnej skórze przekonasz się, że tak tworzone aplikacje są przyszłością mobilnego świata. Dzięki tej książce będziesz tworzyć elastyczne rozwiązania dla szerokiego grona odbiorców.

Z tą książką:

- nauczysz się wykorzystywać HTML5, CSS i JavaScript do tworzenia aplikacji mobilnych
- stworzysz dynamiczny, pełny animacji interfejs użytkownika
- rozwiążesz najczęstsze problemy dzięki PhoneGap
- opublikujesz aplikację w sklepie Google Play!

Stwórz aplikację, opublikuj ją w Google Play i odnieś sukces!

helion.pl
księgarnia
internetowa

Nr katalogowy: 12548



Księgarnia internetowa:
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900
0 601 339900



Helion

Sprawdź najnowsze promocje:

👉 <http://helion.pl/promocje>

Książki najchętniej czytane:

👉 <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

👉 <http://helion.pl/novowosci>

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel.: 32 230 98 63

e-mail: helion@helion.pl

<http://helion.pl>

sięgnij po **WIECEJ**



KOD KORZYŚCI

ISBN 978-83-246-4968-6



Cena 34,90 zł