

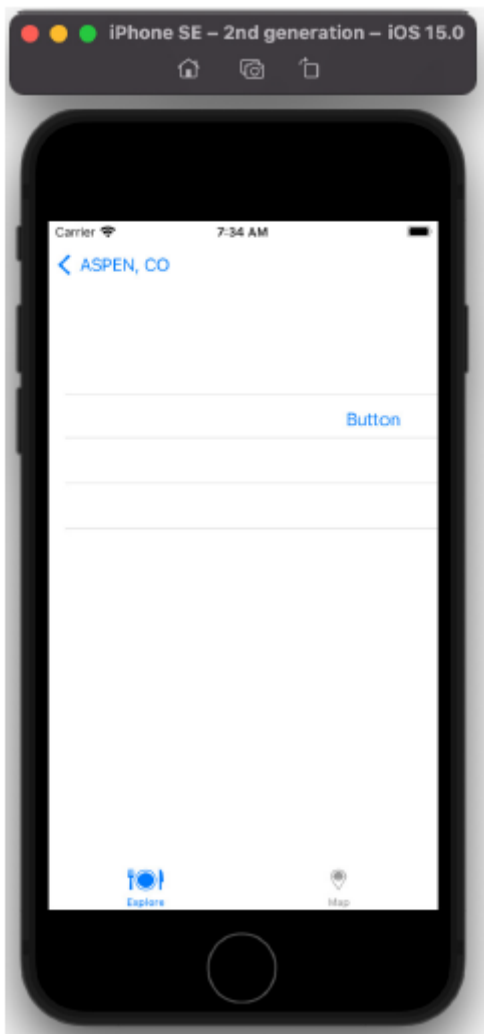
Wyświetlanie danych w statycznym widoku tabeli

Przebyłeś długą drogę, a Twoja aplikacja zawiera dane na wszystkich ekranach z wyjątkiem ekranu szczegółów restauracji. Tu skonfigurujesz klasę `RestaurantDetailViewController` do zarządzania widokami na ekranie Szczegóły restauracji. Następnie dodasz metody do `viewDidLoad()`, aby wypełnić widok tabeli, gdy wyświetlany jest ekran szczegółów restauracji. Na koniec przekażesz odpowiednie wystąpienie `RestaurantItem` z wystąpień `RestaurantListViewController` i `MapViewController` do wystąpienia `RestaurantDetailViewController`, które wyświetli dane z tego wystąpienia `RestaurantItem` na ekranie szczegółów restauracji. Pod koniec dowiesz się, jak tworzyć widoki tabel ze statycznymi komórkami wyświetlającymi dane oraz jak tworzyć własne obrazy mapy. W ten sposób będziesz mógł zaimplementować te funkcje we własnych aplikacjach. Tu zostaną omówione następujące tematy:

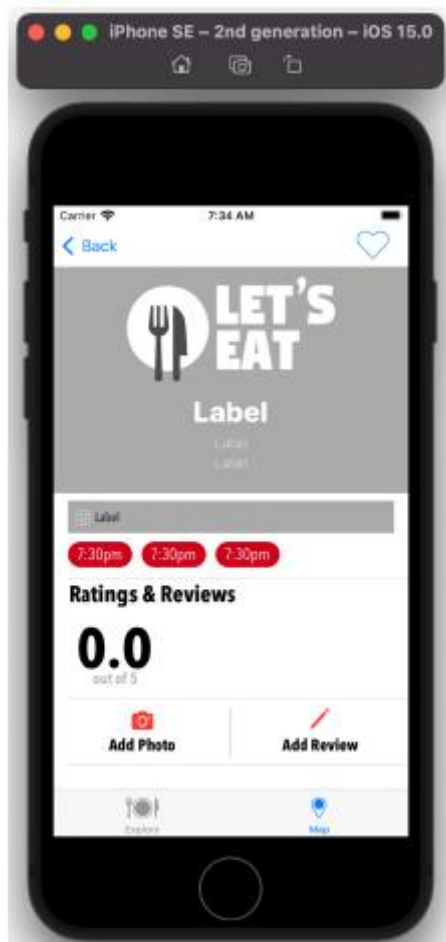
- Konfigurowanie gniazd dla klasy `RestaurantDetailViewController`
- Wyświetlanie danych w statycznym widoku tabeli
- Przekazywanie danych do instancji `RestaurantDetailViewController`

Konfigurowanie gniazd dla klasy `RestaurantDetailViewController`

Twoja aplikacja zawiera dane na wszystkich ekranach z wyjątkiem ekranu szczegółów restauracji. Dostęp do tego ekranu można uzyskać, stukając restaurację na ekranie listy restauracji lub stukając przycisk dymka objaśnienia widoku adnotacji restauracji na ekranie mapy. Jeśli utworzysz i uruchomisz swoją aplikację, dotknięcie restauracji na ekranie Lista restauracji spowoduje wyświetlenie zastępczego ekranu Szczegóły restauracji:



Kliknięcie przycisku w dymku objaśnienia widoku adnotacji restauracji na ekranie mapy powoduje wyświetlenie rzeczywistego ekranu szczegółów restauracji, ale nie zawiera on żadnych danych restauracji:



Aby to naprawić, skonfigurujmy punkty sprzedaży dla klasy RestaurantDetailViewController. Kliknij plik RestaurantDetailViewController w nawigátorze projektów. Dodaj następujące punkty sprzedaży po deklaracji klasy i przed deklaracją właściwości selectedRestaurant:

```
// Nav Bar
@IBOutlet var heartButton: UIBarButtonItem!

// Cell One
@IBOutlet var nameLabel: UILabel!
@IBOutlet var cuisineLabel: UILabel!
@IBOutlet var headerAddressLabel: UILabel!

// Cell Two
@IBOutlet var tableDetailsLabel: UILabel!

// Cell Three
@IBOutlet var overallRatingLabel: UILabel!

// Cell Eight
@IBOutlet var addressLabel: UILabel!
```

// Cell Nine

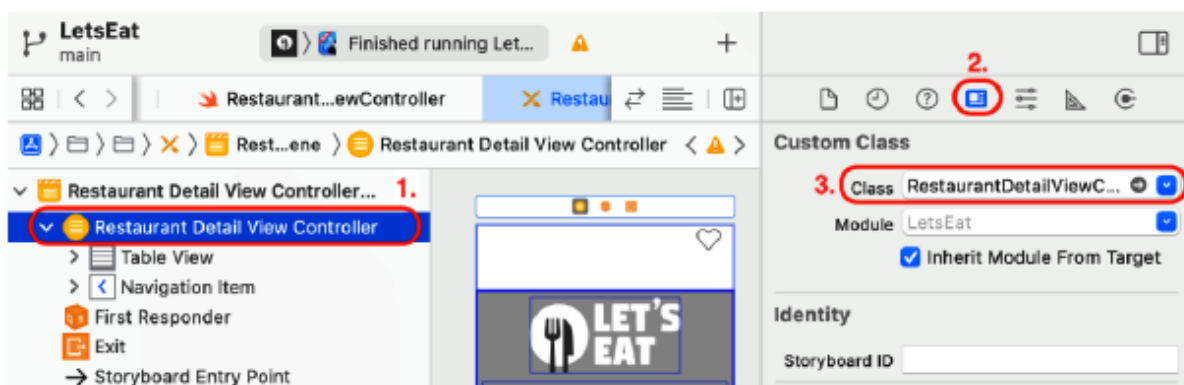
@IBOutlet var locationManagerMapView: UIImageView!

Skonfigurowane przez Ciebie gniazda są następujące:

- heartButton to wyjście dla przycisku w kształcie serca na pasku nawigacyjnym. Nie będziesz go używać w tej książce, ale jest to coś, nad czym możesz później popracować samodzielnie.
- nameLabel to miejsce, w którym etykieta wyświetla nazwę restauracji w pierwszej komórce.
- cuisineLabel jest miejscem, w którym etykieta wyświetla kuchnie oferowane przez restaurację w pierwszej komórce.
- headerAddressLabel to wyjście dla etykiety, która wyświetla adres restauracji w pierwszej komórce.
- tableDetailsLabel to wyjście dla etykiety, która wyświetla szczegóły tabeli restauracji w drugiej komórce.
- generalRatingLabel to miejsce, w którym etykieta wyświetla ogólną ocenę restauracji w trzeciej komórce. Obliczysz i ustawisz tę wartość później.
- addressLabel to miejsce dla etykiety, która wyświetla adres restauracji w ósmej komórce.
- locationManagerMapView to wyjście dla widoku obrazu, który wyświetla mapę lokalizacji restauracji w dziewiątej komórce. Metody generowania tej mapy napiszesz w dalszej części tego rozdziału.

Teraz, po utworzeniu gniazd, połączysz je z elementami interfejsu użytkownika w scenie kontrolera widoku szczegółów restauracji w pliku scenorysu RestaurantDetail. Wykonaj następujące kroki:

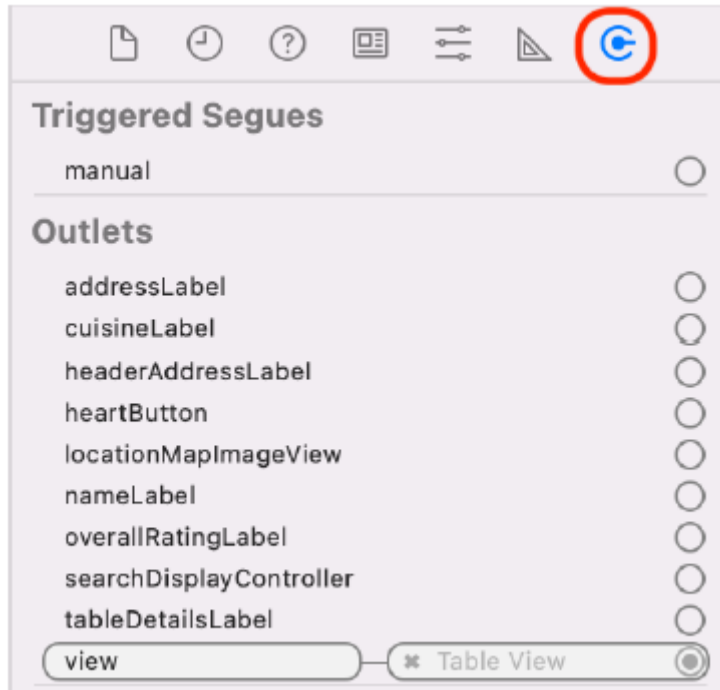
1. Rozwiń folder RestaurantDetail w nawigatory projektu. Kliknij plik scenorysu RestaurantDetail. Następnie kliknij ikonę kontrolera widoku w scenie kontrolera widoku szczegółów restauracji. Następnie kliknij przycisk Inspektor tożsamości. W sekcji Custom Class potwierdź, że Class została ustawiona na klasę RestaurantDetailViewController:



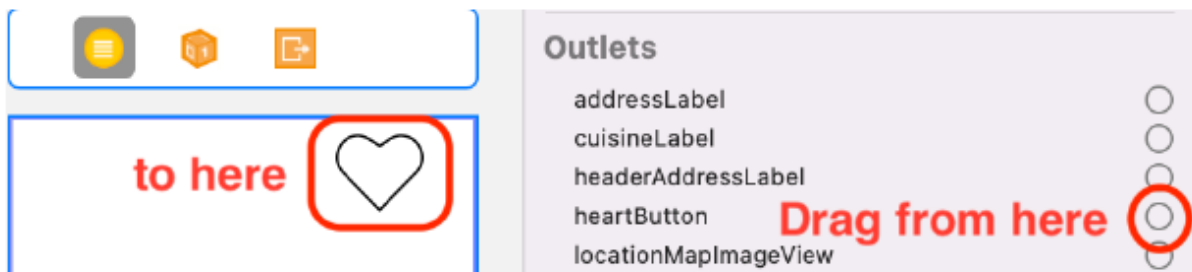
Zwróć uwagę, że nazwa kontrolera widoku zmienia się na Kontroler widoku szczegółów restauracji po ustawieniu klasy. W przeciwieństwie do widoku tabeli w scenie kontrolera widoku lokalizacji, widok tabeli w scenie kontrolera widoku szczegółów restauracji zawiera komórki statyczne, co oznacza, że liczba komórek nie jest generowana dynamicznie na podstawie danych z obiektu modelu. Jak widać w zarysie dokumentu, istnieje dziewięć komórek, a każda komórka została już skonfigurowana z

odpowiednimi obiektami widoku. Kliknięcie każdej komórki widoku tabeli w konspekcie dokumentu spowoduje wyświetlenie tej komórki w obszarze edytora.

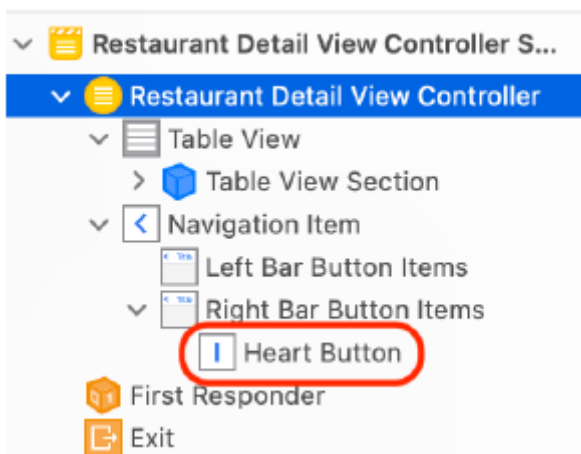
2. Kliknij przycisk Inspektora połączeń. Zobaczysz wszystkie gniazda, które dodałeś wcześniej w klasie RestaurantDetailViewController:



3. Kliknij i przeciągnij z wylotu heartButton do serca na pasku nawigacyjnym:



4. Gniazdo heartButton jest teraz podłączone. Zauważ, że opis widoku zmieni się na Przycisk Serca w zarysie dokumentu:



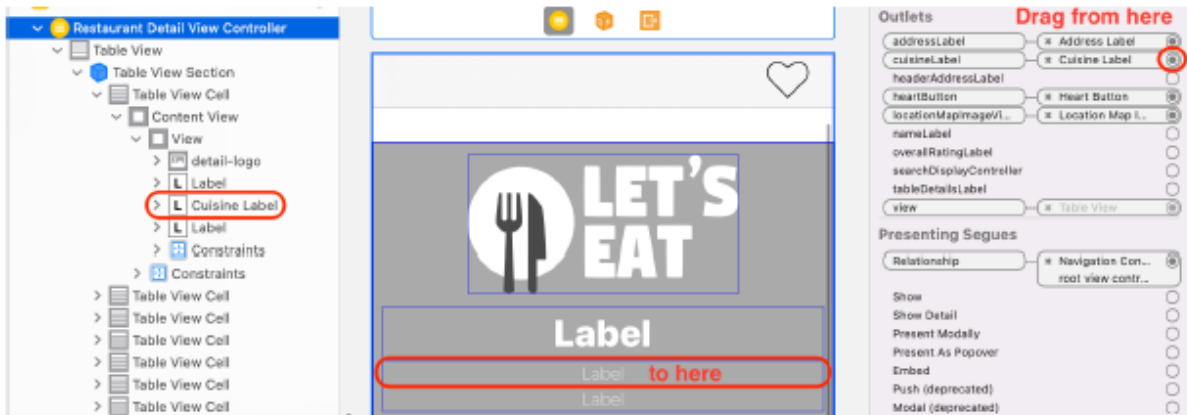
5. Kliknij ostatnią komórkę widoku tabeli w zarysie dokumentu, aby zobaczyć dół widoku tabeli, a następnie kliknij opcję Kontroler widoku szczegółów restauracji (może być konieczne kliknięcie kilka razy). Powinieneś zobaczyć UIImageView. Kliknij i przeciągnij z wylotu locationMapImageView do widoku obrazu w ostatniej komórce, aby je połączyć. Zwróć uwagę, że nazwa zmieni się z widoku obrazu na widok obrazu mapy lokalizacji w zarysie dokumentu:



6. Kliknij ósmą komórkę widoku tabeli w zarysie dokumentu i kliknij opcję Kontroler widoku szczegółów restauracji. Kliknij i przeciągnij z wyjścia addressLabel do Label w ósmej komórce, aby je połączyć. Zauważ, że nazwa zmieni się z Label na Address Label w konspekcie dokumentu:



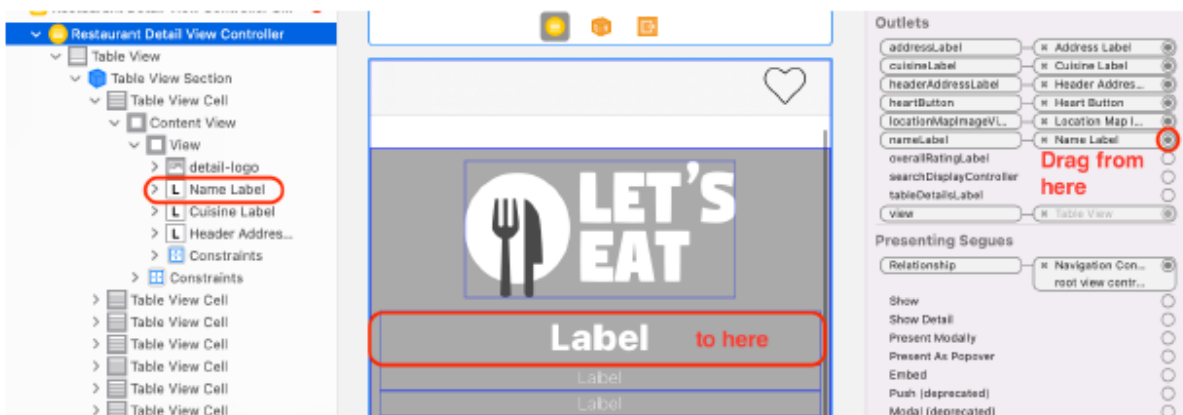
7. Kliknij pierwszą komórkę widoku tabeli w zarysie dokumentu i kliknij opcję Kontroler widoku szczegółów restauracji. Kliknij i przeciągnij z wylotu cuisineLabel do drugiej etykiety w pierwszej komórce, aby je połączyć. Zwróć uwagę, że nazwa zmieni się z etykiety na etykietę kuchni w konspekcie dokumentu:



8. Kliknij i przeciągnij z wyjścia headerAddressLabel do trzeciej etykiety w pierwszej komórce, aby je połączyć. Zauważ, że nazwa zmieni się z Label na Header Address Label w konspekcie dokumentu:



9. Kliknij i przeciągnij od wylotu nameLabel do pierwszej etykiety w pierwszej komórce, aby je połączyć. Zauważ, że nazwa zmieni się z Label na Name Label w konspekcie dokumentu:



10. Kliknij trzecią komórkę widoku tabeli w zarysie dokumentu i kliknij Kontroler widoku szczegółów restauracji. Kliknij i przeciągnij z gniazdka ogólnegoRatingLabel do etykiety z dużą czarną 0,0 w środku, aby je połączyć. Zwróć uwagę, że nazwa zmieni się z etykiety na etykietę oceny ogólnej w zarysie dokumentu:



11. Kliknij drugą komórkę widoku tabeli w zarysie dokumentu i kliknij opcję Kontroler widoku szczegółów restauracji. Kliknij i przeciągnij z wylotu tableDetailsLabel do etykiety tuż nad trzema czerwonymi przyciskami w drugiej komórce, aby je połączyć. Zwróć uwagę, że nazwa zmieni się z etykiety na etykietę szczegółów tabeli w konspekcie dokumentu:



Wszystkie punkty sprzedaży dla klasy RestaurantDetailViewController zostały już skonfigurowane. W następnej sekcji zmodyfikujesz klasę RestaurantDetailViewController, aby odbierać dane restauracji z występów RestaurantListViewController i MapViewController i wyświetlać je na ekranie szczegółów restauracji.

Wyświetlanie danych w statycznym widoku tabeli

Wszystkie punkty sprzedaży w klasie RestaurantDetailViewController zostały pomyślnie połączone z elementami interfejsu użytkownika na ekranie szczegółów restauracji. Ponieważ jest to statyczny widok tabeli, nie będziesz stosować protokołu UITableViewDataSource do wypełniania gniazd. Zamiast tego napiszesz niestandardowe metody, aby to zrobić. Wykonaj następujące kroki:

1. Kliknij plik RestaurantDetailViewController w nawigatorze projektów.
2. Dodaj kod, aby zaimportować framework MapKit po istniejącej instrukcji importu:

```
import MapKit
```

Jest to wymagane, ponieważ będziesz używać właściwości i metod platformy MapKit do generowania obrazu mapy dla widoku obrazu w ostatniej komórce.

3. Dodaj prywatne rozszerzenie zawierające kod, aby ustawić etykiety na ekranie szczegółów restauracji po ostatnim nawiasie klamrowym:

```
private extension RestaurantDetailViewController {
    func setupLabels() {
        guard let restaurant =
            selectedRestaurant else {
```



```

return
}

title = restaurant.name

titleLabel.text = restaurant.name

cuisineLabel.text = restaurant.subtitle

headerAddressLabel.text = restaurant.address

tableDetailsLabel.text = "Table for 7, tonight
at 10:00 PM"

addressLabel.text = restaurant.address
}
}

```

Metoda `setupLabels()` jest dość prosta; pobiera wartości z instancji `RestaurantItem` i umieszcza je w punktach sprzedaży w instancji `RestaurantDetailViewController`, z wyjątkiem `tableDetailsLabel`, do której właśnie przypisano ciąg.

4. W ostatniej komórce wyświetli się obraz mapy. Aby to zrobić, wygenerujesz obraz z regionu mapy i ustawisz wylot `locationMapImageLabel`, aby wyświetlić ten obraz. Ten obraz będzie również zawierał ten sam niestandardowy obraz adnotacji, który został użyty na ekranie mapy. Dodaj następującą metodę po `setupLabels()` i przed ostatnim nawiasem klamrowym:

```

func createMap() {
guard let annotation = selectedRestaurant, let long
= annotation.long, let lat = annotation.lat else {
return
}

let location = CLLocationCoordinate2D(latitude:
lat, longitude: long)

takeSnapshot(with: location)
}

```

Ta metoda tworzy wystąpienie `CLLocationCoordinate2D` przy użyciu właściwości `lat` i `long` właściwości `selectedRestaurant` i przypisuje je do lokalizacji. Następnie wywołuje metodę `takeSnapshot(with:)`, przekazując lokalizację jako parametr.

5. Zobaczysz błąd, ponieważ `takeSnapshot(with:)` nie został jeszcze zaimplementowany, więc dodaj następujący kod po funkcji `createMap()`, aby go zaimplementować:

```

func takeSnapshot(with location:
CLLocationCoordinate2D) {

```

```
let mapSnapshotOptions = MKMapSnapshotter.Options()

var loc = location

let polyline = MKPolyline(coordinates: &loc, count:
1 )

let region = MKCoordinateRegion(polyline.
boundingMapRect)

mapSnapshotOptions.region = region

mapSnapshotOptions.scale = UIScreen.main.scale

mapSnapshotOptions.size = CGSize(width: 340,
height: 208)

mapSnapshotOptions.showsBuildings = true

mapSnapshotOptions.pointOfInterestFilter =
.includingAll

let snapShotter = MKMapSnapshotter(options:
mapSnapshotOptions)

snapShotter.start() { snapshot, error in
guard let snapshot = snapshot else {
return
}

 UIGraphicsBeginImageContextWithOptions(
mapSnapshotOptions.size, true, 0)

snapshot.image.draw(at: .zero)

let identifier = "custompin"

let annotation = MKPointAnnotation()

annotation.coordinate = location

let pinView = MKPinAnnotationView(annotation:
annotation, reuseIdentifier: identifier)

pinView.image = UIImage(named: "customannotation")!

let pinImage = pinView.image

var point = snapshot.point(for: location)

let rect = self.locationMapView.bounds
```

```

if rect.contains(point) {
let pinCenterOffset = pinView.centerOffset
point.x -= pinView.bounds.size.width / 2
point.y -= pinView.bounds.size.height / 2
point.x += pinCenterOffset.x
point.y += pinCenterOffset.y
pinImage?.draw(at: point)
}
if let image =
    UIGraphicsGetImageFromCurrentImageContext() {
    UIGraphicsEndImageContext()
    DispatchQueue.main.async {
    self.locationMapView.image = image
    }
    }
    }
    }
}

```

Pełny opis tej metody wykracza poza zakres tej książki, ale oto proste wyjaśnienie jej działania. Biorąc pod uwagę lokalizację, wykonuje migawkę mapy w tej lokalizacji, dodaje niestandardową adnotację używaną wcześniej na ekranie mapy, konwertuje ją na obraz i przypisuje ją do gniazda `locationMapView` w instancji `RestaurantDetailViewController`.

6. Napisałeś wszystkie metody, które są wymagane dla klasy `RestaurantDetailViewController`, aby wyświetlić żądane szczegóły instancji `RestaurantItem` na ekranie Szczegóły restauracji. W rozszerzeniu `private` przed definicją metody `setupLabels()` dodaj metodę `initialize()`, która wywołuje metody `setupLabels()` i `createMap()`:

```

func initialize() {
    setupLabels()
    createMap()
}

```

7. Zmodyfikuj metodę `viewDidLoad()`, aby wywoływała metodę `initialize()`, gdy instancja `RestaurantDetailViewController` ładuje swój widok:

```

override func viewDidLoad() {
    super.viewDidLoad()
    initialize()
}

```

}

Przypomnij sobie, że w rozdziale 16, Pierwsze kroki z MapKit, skonfigurowałeś już klasę MapViewController do przekazywania instancji RestaurantItem do instancji RestaurantDetailViewController. Stwórz i uruchom swoją aplikację i przejdź do ekranu Mapa. Kliknij jedną z restauracji, aby wyświetlić dymek z objaśnieniem. Kliknij przycisk w dymku objaśnienia, a na ekranie szczegółów restauracji powinny pojawić się szczegóły restauracji:



Jeśli przewiniesz w dół, zobaczysz obraz mapy w ostatniej komórce:



Zakończyłeś modyfikowanie klasy `RestaurantDetailViewController`, ale nadal musisz przekazać wybraną instancję `RestaurantItem` z instancji `RestaurantListViewController` do instancji `RestaurantDetailViewController`. Zrobisz to w następnej sekcji.

Przekazywanie danych do instancji `RestaurantDetailViewController`

Dodałeś i podłączyłeś punkty sprzedaży dla ekranu szczegółów restauracji w klasie `RestaurantDetailViewController`. Dodano również kod do tej klasy, aby uzyskać dane restauracji z wystąpienia `RestaurantItem` i użyć go do wypełnienia jego punktów sprzedaży. Ostatnią rzeczą, którą musisz zrobić, to przekazać wybraną instancję `RestaurantItem` z instancji `RestaurantListViewController` do instancji `RestaurantDetailViewController`. Wykonaj następujące kroki:

1. Kliknij plik `RestaurantListViewController` w nawigatorze projektów.
2. Dodaj następujący kod po `viewDidLoad()`: aby wywołać metodę `showRestaurantDetail(segue:)`, jeśli identyfikator `segue` jest `showDetail`:

```
override func prepare(for segue: UIStoryboardSegue,
sender: Any?) {
    if let identifier = segue.identifier {
        switch identifier {
        case Segue.showDetail.rawValue:
            showRestaurantDetail(segue: segue)
        default:
            print("Segue not added")
        }
    }
}
```

Przypomnij sobie, że dodano przejście między sceną kontrolera widoku listy restauracji a sceną kontrolera widoku szczegółów restauracji w scenorysie. Przed przejściem wystąpienia `RestaurantListViewController` do innego kontrolera widoku sprawdzany jest identyfikator `segue`. Jeżeli identyfikatorem przejścia jest `showDetail`, to wykonywana jest metoda `showRestaurantDetail`. Tylko przejście między sceną kontrolera widoku listy restauracji a szczegółami restauracji

Scena kontrolera widoku ma identyfikator `showDetail`, więc docelowy kontroler widoku musi być instancją `RestaurantDetailViewController`.

3. Zobaczysz błąd, ponieważ metoda `showRestaurantDetail(segue:)` nie została zaimplementowana. Ta metoda przekaże wystąpienie `RestaurantItem` z wystąpienia `RestaurantListViewController` do wystąpienia `RestaurantDetailViewController`. Dodaj go po otwierającym nawiasie klamrowym prywatnego rozszerzenia w klasie `RestaurantListViewController`:

```
func showRestaurantDetail(segue: UIStoryboardSegue) {
    if let viewController = segue.destination as?
```

```

RestaurantDetailViewController, let indexPath =
collectionView.indexPathsForSelectedItems?.first {
selectedRestaurant = manager.restaurantItem(at:
indexPath.row)
viewController.selectedRestaurant =
selectedRestaurant
}
}

```

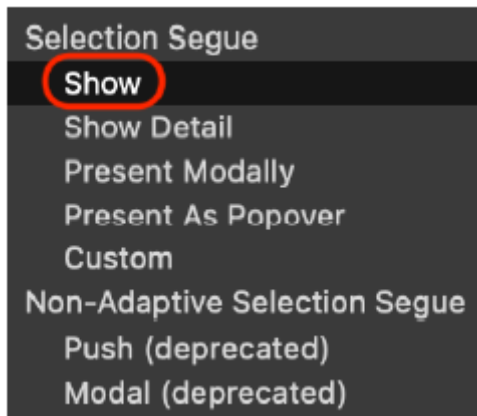
Ta metoda najpierw sprawdza, czy miejsce docelowe segue jest instancją RestaurantDetailViewController i pobiera indeks kolekcji, wyświetli komórkę, która została podsłuchiwana. Następnie menedżer zwraca instancję RestaurantItem zapisaną w tym indeksie, która jest przypisana do selectedRestaurant. Właściwość selectedRestaurant wystąpienia RestaurantDetailViewController zostanie następnie ustawiona na to wystąpienie.

Przyjrzyjmy się teraz scenie kontrolera widoku listy restauracji w głównym pliku scenorysu. Jest obecnie połączony z zastępczą sceną kontrolera widoku. Zaktualizujesz plik głównego scenorysu, aby usunąć symbol zastępczy i połączyć scenę kontrolera widoku listy restauracji ze sceną kontrolera widoku szczegółów restauracji w pliku scenorysu Szczegóły restauracji. Wykonaj następujące kroki:

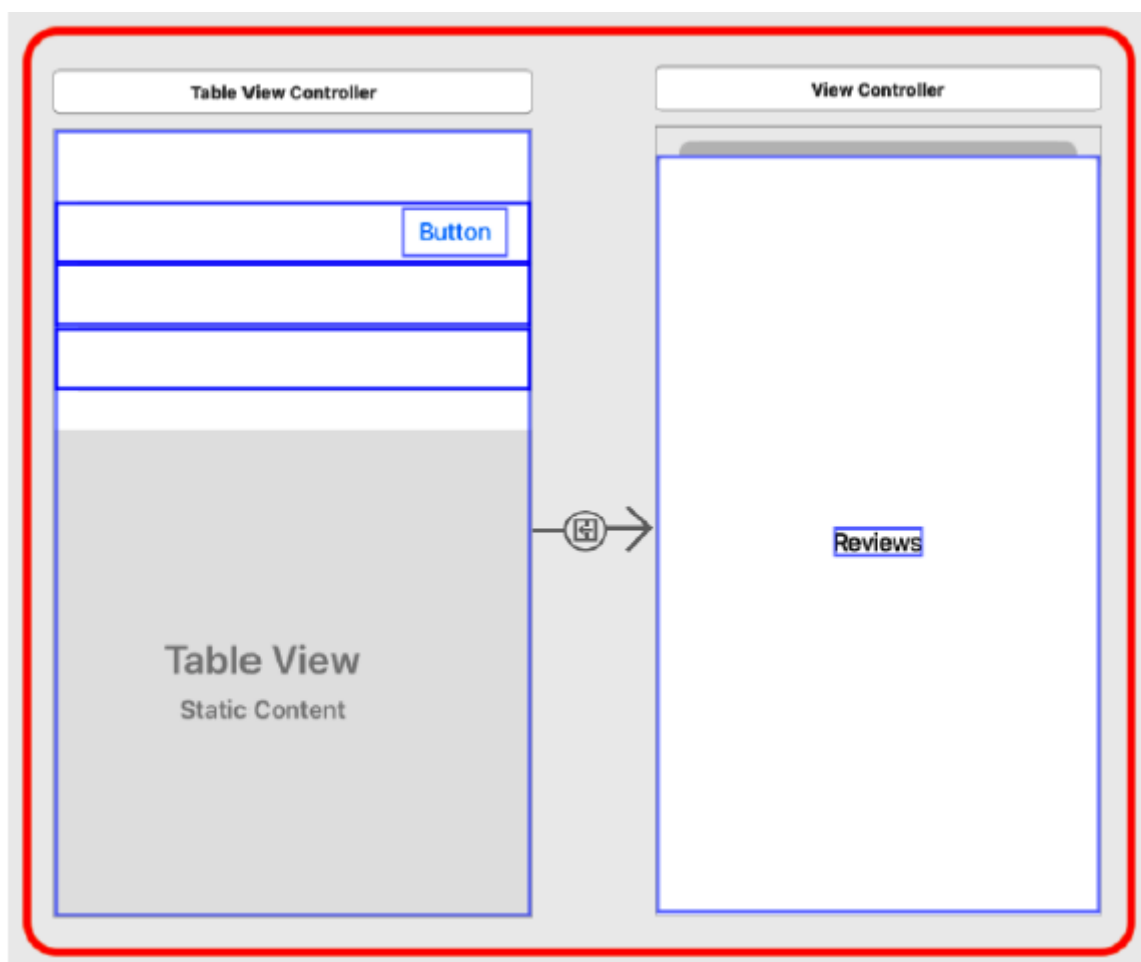
1. Kliknij plik głównego scenorysu i zlokalizuj scenę kontrolera widoku listy restauracji. Kliknij restauracjaKomórka w konspekcie dokumentu. Następnie naciśnij Ctrl + przeciągnij z restaurantCell do odwołania do scenorysu RestaurantDetail (dodano to odniesienie do scenorysu w rozdziale 16, Pierwsze kroki z MapKit), jak pokazano:



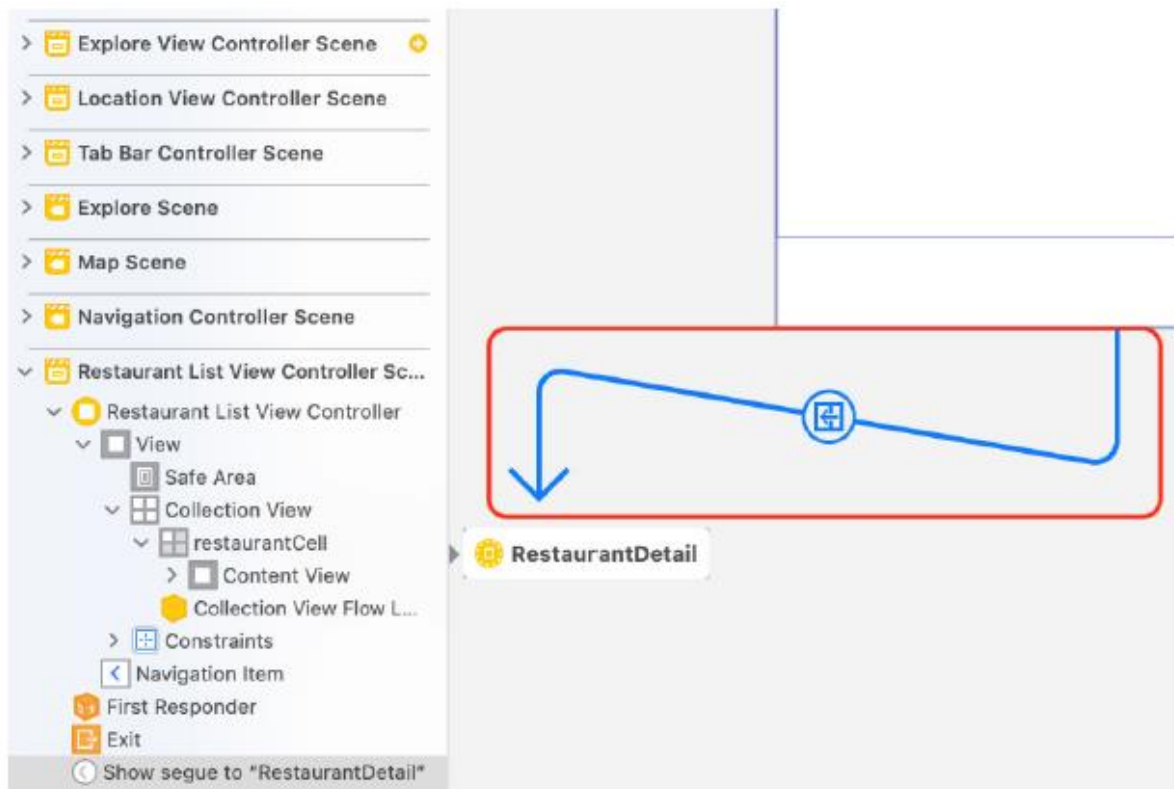
2. Wybierz Pokaż z wyskakującego menu, które się pojawi:



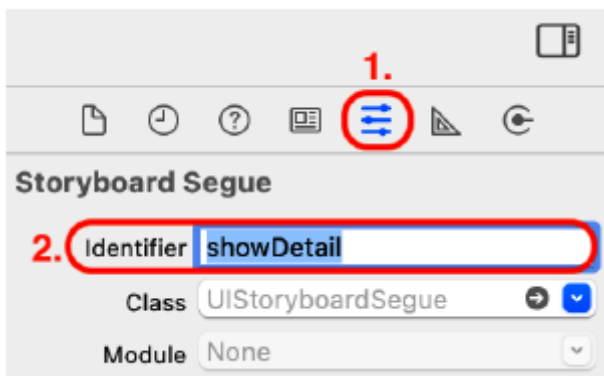
3. Usuń sceny zastępcze ze scenorysu, wybierając je i naciskając klawisz Delete na klawiaturze, ponieważ nie są już potrzebne:



4. Ustawisz identyfikator segue na showDetail. Jak wspomniano wcześniej, spowoduje to ustawienie właściwości `selectedRestaurant` wystąpienia `RestaurantDetailViewController`. Wybierz dodane przed chwilą przejście:



5. Kliknij przycisk Inspektora atrybutów. W sekcji Storyboard Segue ustaw Identyfikator na showDetail:



Zbuduj i uruchom swój projekt. Wybierz miasto i rodzaj kuchni. Kliknij jedną z restauracji na ekranie Lista restauracji. Szczegóły tej restauracji pojawią się na ekranie szczegółów restauracji:



Implementacja ekranu szczegółów restauracji została zakończona. Po wybraniu restauracji na ekranach Mapa lub Lista restauracji, szczegóły tej restauracji zostaną wyświetlone na ekranie Szczegóły restauracji. Wspaniały!

Podsumowanie

W tej Części połączyłeś punkty sprzedaży w klasie `RestaurantDetailViewController` z ekranem Szczegóły restauracji. Następnie dodano metody do `viewDidLoad()`, aby wypełnić widok tabeli, gdy wyświetlany jest ekran szczegółów restauracji. Na koniec przekazałeś odpowiednią instancję `RestaurantItem` z wystąpieniem `RestaurantListViewController` i `MapViewController` do wystąpienia `RestaurantDetailViewController`, umożliwiające wyświetlanie danych z tego wystąpienia `RestaurantItem` na ekranie szczegółów restauracji. Dzięki temu nauczyłeś się, jak tworzyć widoki tabel ze statycznymi komórkami wyświetlającymi dane, a także tworzyć niestandardowe obrazy mapy, które możesz teraz zaimplementować we własnych aplikacjach. Gratulacje! Wszystkie ekrany w Twojej aplikacji wyświetlają teraz dane. Jeśli jednak spojrzysz na ekran Szczegóły restauracji, nie ma żadnych ocen, recenzji ani zdjęć restauracji ani możliwości ich dodania.