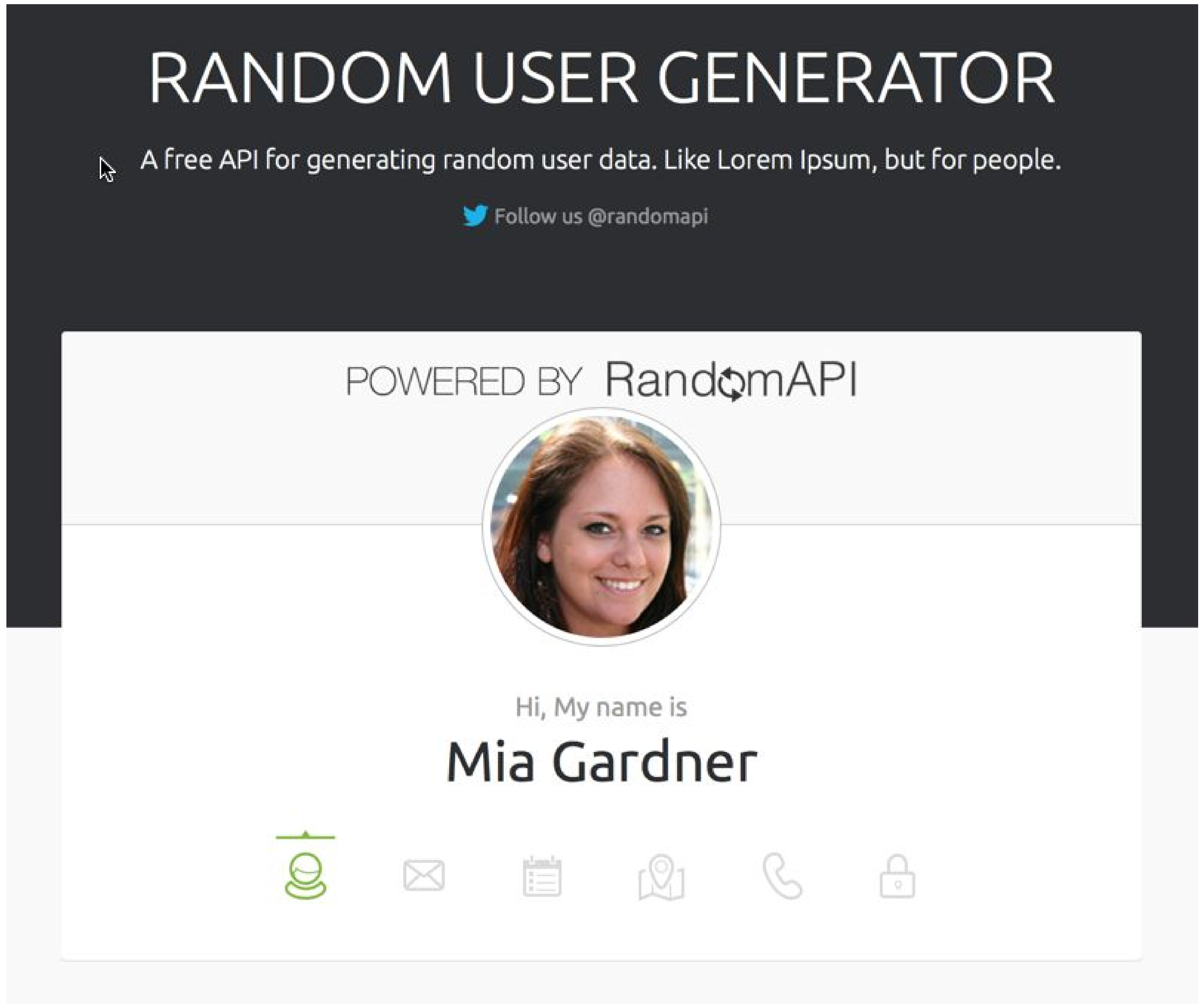
Alloy Model Binding

A Random User Sample App

Random User App

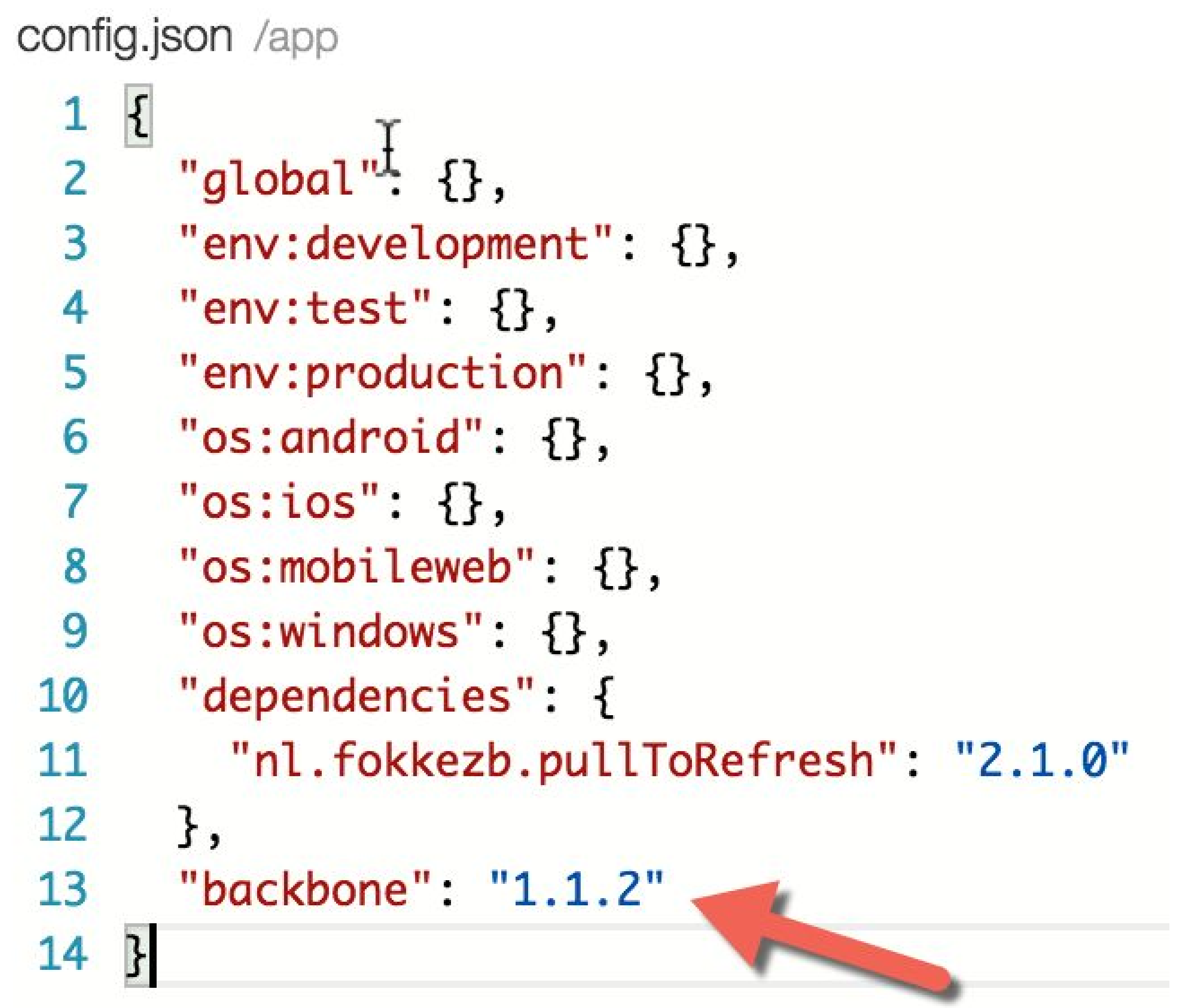


- App to demo Alloy Model Validation & Binding
- Using RandomUserMe API http://randomuser.me
- Login Form + Validation
- TableView Collection Binding
- ListView Collection Binding
- ScrollableView Collection Binding
- Filter collection to single user with details view

Why Alloy Model Data Binding?

- Consistent approach to data across projects
- Allows for Separation of Concerns
- Less code
- Light Controllers
- Light Views
- Keep styling in TSS stylesheets
- Easier for ‘front-end’ designers to collaborate with developers to build out UI

Alloy Models use Backbone


config.json /app

```
{
  "global": {},
  "env:development": {},
  "env:test": {},
  "env:production": {},
  "os:android": {},
  "os:ios": {},
  "os:mobileweb": {},
  "os:windows": {},
  "dependencies": {
    "nl.fokkezb.pullToRefresh": "2.1.0"
  },
  "backbone": "1.1.2"
}
```

- Alloy relies on the Backbone API to sync model data to persistent storage
- Alloy Models inherit from the Backbone.Model class

Backbone gives:

- Structure to apps by providing **Models** with key-value binding and custom events
- **Collections** with a rich API of enumerable functions
- Connects it all to your existing API over a **RESTful JSON** interface.

Alloy Backbone Support:

- Alloy been stuck on 0.9.2 for a LONG time which been an issue for some advanced scenarios when needed full Backbone API
- Alloy 1.6.0 (Ti SDK 4.0.0) will introduce support for Backbone 1.1.2
- However, due to breaking changes 0.9.2 is still default, need to change config.json to use http://docs.appcelerator.com/platform/latest/#!/guide/Alloy_Backbone_Migration

Models need a Sync Adapter

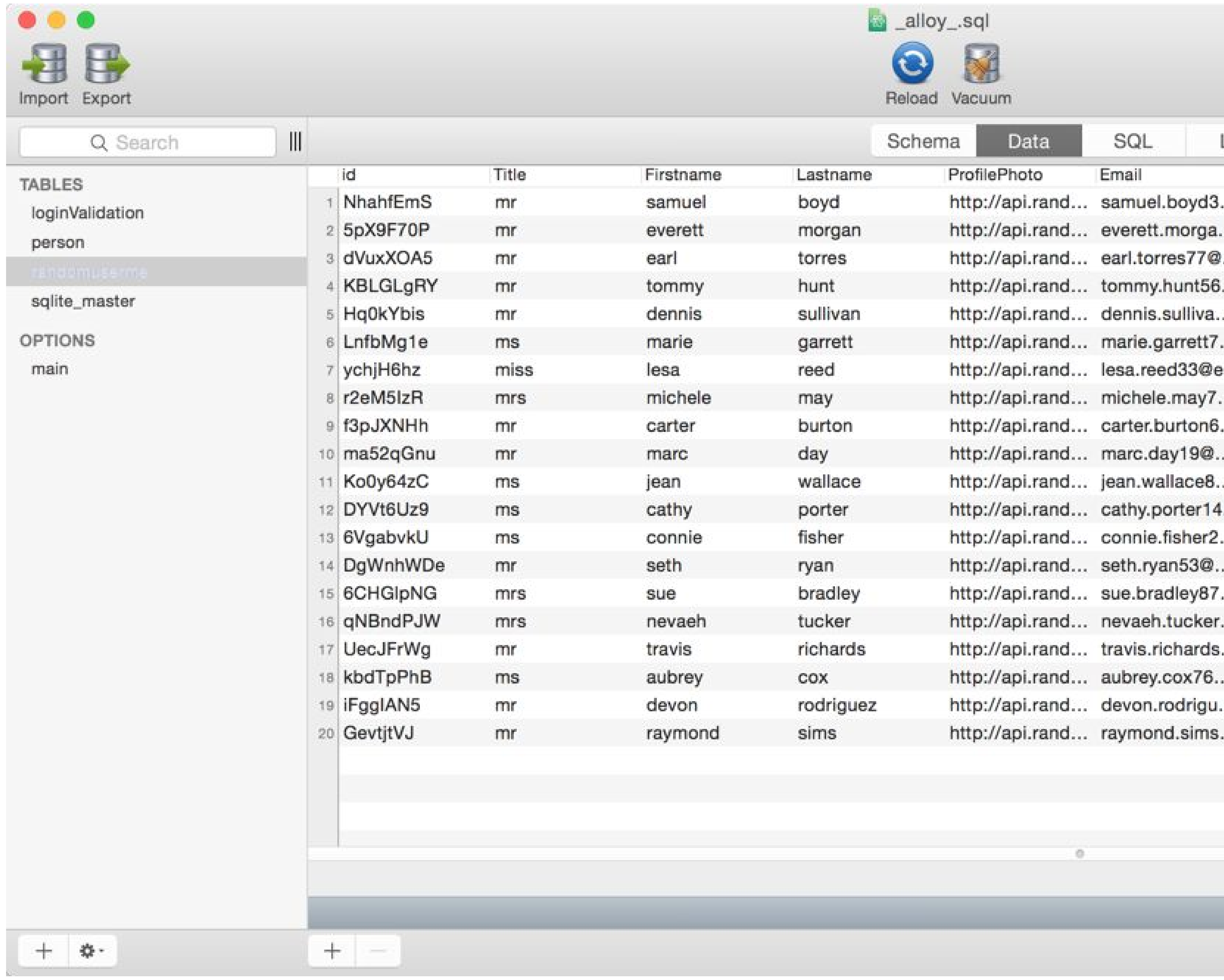
- A sync adapter allows you to store and load your models to a persistent storage device, such as an on-device database or remote server

Alloy has two in-build adapters:

- *sql* for a SQLite database on the Android and iOS platform
- *properties* for storing data locally in the Titanium SDK context

You can also write your own adapter:

- *Rest API Adapter by* Mads Møller - https://github.com/viezel/napp.alloy.adapter.restapi
- *Rest SQL Adapter by* Mads Møller - https://github.com/viezel/napp.alloy.adapter.restsql
  - Offline Cache version of API in SQLite Database
- *ACS Adapter by Aaron Saunders* - https://github.com/aaronksaunders/Appcelerator-Cloud-Services-Sync-Adapter
- Windmill Adapter?



	id	Title	Firstname	Lastname	ProfilePhoto	Email
1	NhahfEmS	mr	samuel	boyd	http://api.rand…	samuel.boyd3.
2	5pX9F70P	mr	everett	morgan	http://api.rand…	everett.morga.
3	dVuxXOA5	mr	earl	torres	http://api.rand…	earl.torres77@
4	KBLGLgRY	mr	tommy	hunt	http://api.rand…	tommy.hunt56.
5	Hq0kYbis	mr	dennis	sullivan	http://api.rand…	dennis.sulliva..
6	LnfbMg1e	ms	marie	garrett	http://api.rand…	marie.garrett7.
7	ychjH6hz	miss	lesa	reed	http://api.rand…	lesa.reed33@e
8	r2eM5IzR	mrs	michele	may	http://api.rand…	michele.may7.
9	f3pJXNHh	mr	carter	burton	http://api.rand…	carter.burton6.
10	ma52qGnu	mr	marc	day	http://api.rand…	marc.day19@..
11	Ko0y64zC	ms	jean	wallace	http://api.rand…	jean.wallace8..
12	DYVt6Uz9	ms	cathy	porter	http://api.rand…	cathy.porter14
13	6VgabvkU	ms	connie	fisher	http://api.rand…	connie.fisher2.
14	DgWnhWDe	mr	seth	ryan	http://api.rand…	seth.ryan53@..
15	6CHGlpNG	mrs	sue	bradley	http://api.rand…	sue.bradley87.
16	qNBndPJW	mrs	nevaeh	tucker	http://api.rand…	nevaeh.tucker.
17	UecJFrWg	mr	travis	richards	http://api.rand…	travis.richards.
18	kbdTpPhB	ms	aubrey	cox	http://api.rand…	aubrey.cox76..
19	iFggIAN5	mr	devon	rodriguez	http://api.rand…	devon.rodrigu.
20	GevtjtVJ	mr	raymond	sims	http://api.rand…	raymond.sims.

Model Definition File

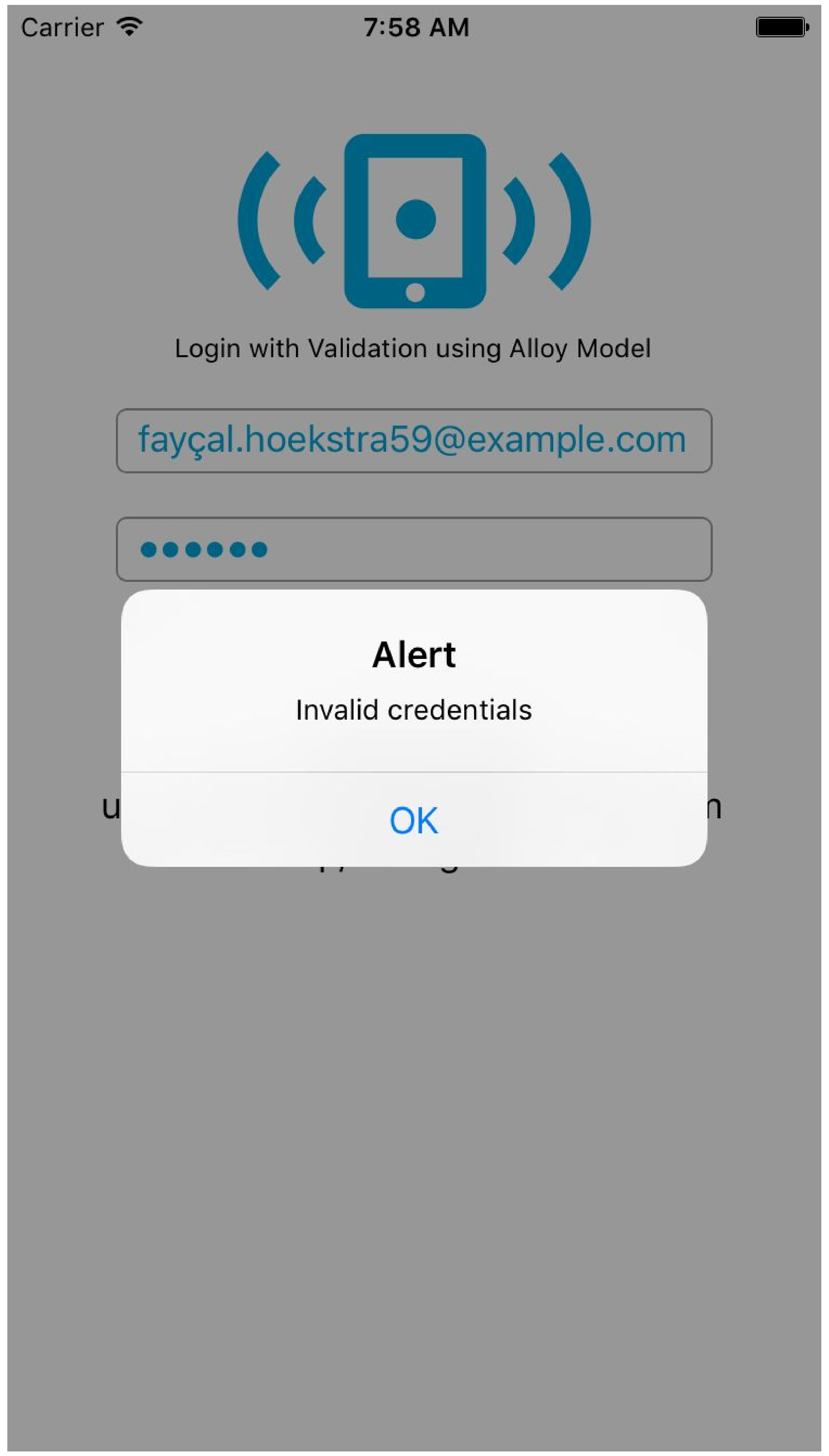
Model File

```
exports.definition = {
    config: {
        // table schema and adapter information
    },
    extendModel: function (Model) {
        _.extend(Model.prototype, {
            // Extended, override or implement Backbone.Model
        });
        return Model;
    },
    extendCollection: function (Collection) {
        _.extend(Collection.prototype, {
            // Extended, override or implement Backbone.Collection
        });
        return Collection;
    }
};
```

Model file has three objects:

- Config - defines table schema and adapter info
    - e.g. database columns, API endpoint, set headers
- Two functions can extend / override
    - extendModel
      e.g. for local validation
    - extendCollection
      e.g. for sorting

Using a Model for Local Validation


Carrier
7:58 AM
Login with Validation using Alloy Model
fayçal.hoekstra59@example.com
••••••
Alert
Invalid credentials
OK

loginValidation.js Model File (simplified)

```
exports.definition = {
    extendModel: function (Model) {
        _.extend(Model.prototype, {
            // extended functions and properties go here
            validate: function (attributes, options) {
                for (var key in attributes) {
                    var value = attributes[key];
                    if (key === "email") {
                        if (value.length > 0) {
                            if (!validateEmail(value)) {
                                return "Invalid email";
                            }
                        } else {
                            return "No email address entered";
                        }
                    }
                }
                return; // return nothing for success
            }
        });

        return Model;
    }
};
```

controller.js File

- Create model file for local validation

```
var model = Alloy.createModel("loginValidation");
```

- When login button pressed use Backbone Events

```
model.set({
     email : $.txtEmail.value,
     password : $.txtPassword.value
 });

model.on("error", function(model, error) {
     alert(error);
});

model.on("change", function(model){
     loginUser();
});
```

Using a Model Collection for Databinding

randomuserme.json

```
results:
[
        {
                user:
                {
                        gender: "female",
                        name:
                        {
                                title: "ms",
                                first: "andrea",
                                last: "moore"
                        },
                        location:
                        {
                                street: "2394 park lane",
                                city: "coventry",
                                state: "durham",
                                postcode: "YQ2 5WY"
                        },
                        email: "andrea.moore21@example.com",
                        username: "tinycat608",
                        password: "nimbus",
                        salt: "0Obk9i1e",
                        md5: "3bcfe5d97cf834b0e8fd99bdea5070a1",
                        sha1: "ab006d402cecc426015edc06547a51b6def14ba5",
                        sha256: "73bbcc45cc71be415811f5a5ca41e7f2eaac51459d0507d802b38929f8d6c5d5",
                        registered: "1276110521",
                        dob: "481999452",
                        phone: "013873 47234",
                        cell: "0701-904-609",
                        NINO: "SA 16 70 18 G",
                        picture:
                        {
                                large: "http://api.randomuser.me/portraits/women/44.jpg",
                                medium: "http://api.randomuser.me/portraits/med/women/44.jpg",
                                thumbnail: "http://api.randomuser.me/portraits/thumb/women/44.jpg"
                        },
                        version: "0.6",
                        nationality: "GB"
                },
                seed: "dogfish"
        }
]
```

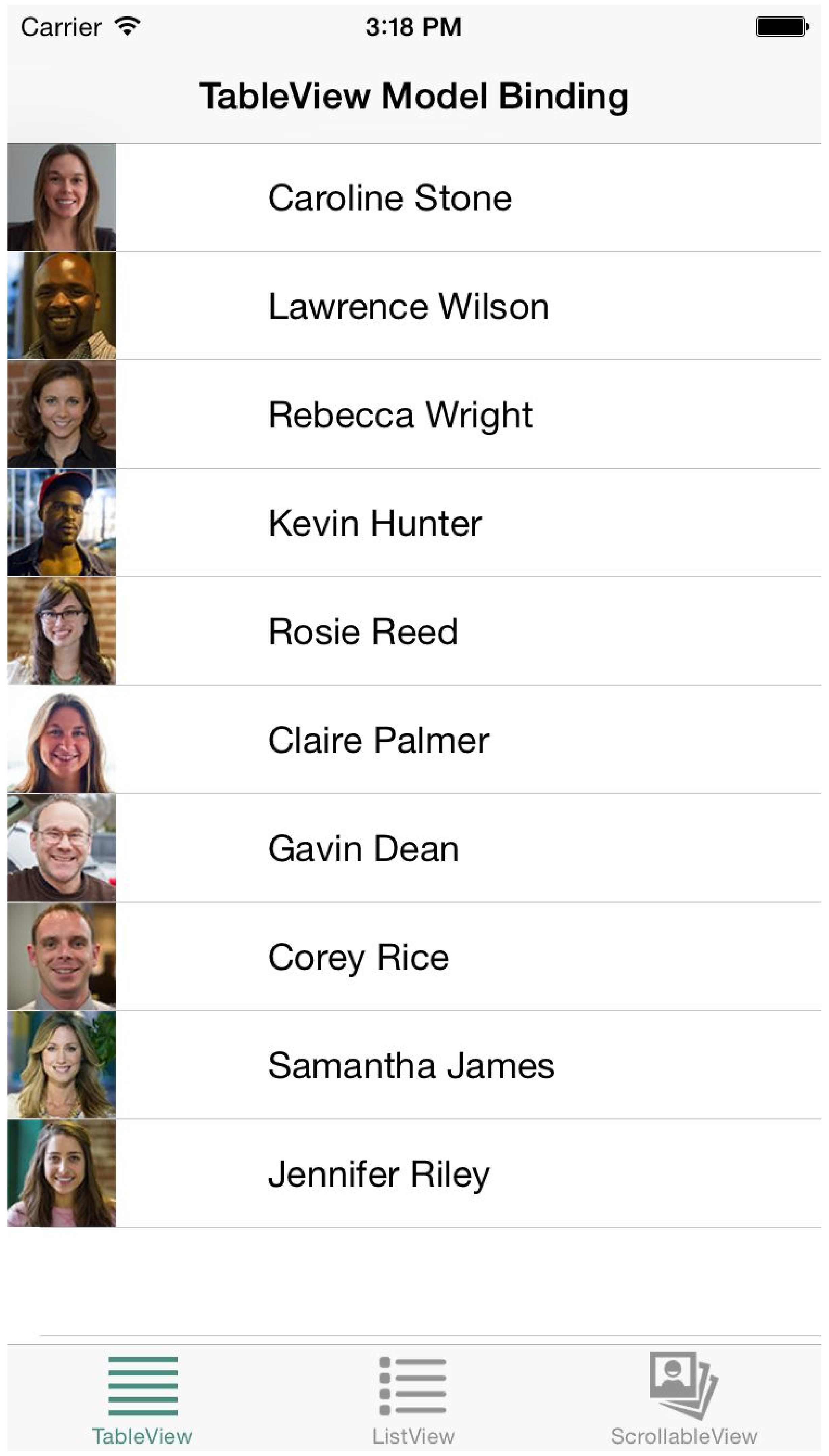
randomuserme.js Model File (sqlrest adapter)

```
config : {
        "columns": {
            "id":"TEXT PRIMARY KEY",
            "Title": "text",
            "Firstname":"text",
            "Lastname": "text",
            "ProfilePhoto": "text",
            "Email": "text",
            "Password": "text"
        },
        "URL": "http://api.randomuser.me/?results=20",
        "adapter" : {
            "type" : "sqlrest",
            "collection_name" : "randomuserme",
            "idAttribute" : "id"
        }
}
```

randomuserme.js Model File (sqlrest adapter)

```
parentNode: function(data) {
   var persons = [];
   _.each(data.results, function(_entry) {
     var entry = {};

     entry.id = _entry.user.registered;
     entry.Title = _entry.user.name.title;
     entry.Firstname = _entry.user.name.first;
     entry.Lastname = _entry.user.name.last;

     entry.fullName = entry.Firstname +” “ +entry.Lastname;

     entry.ProfilePhoto = _entry.user.picture.large;
     entry.Email = _entry.user.email;
     entry.Password = _entry.user.password;

     persons.push(entry);
   });
   return persons;
 }
```

Collection + Binding


Carrier 3:18 PM

TableView Model Binding

Caroline Stone

Lawrence Wilson

Rebecca Wright

Kevin Hunter

Rosie Reed

Claire Palmer

Gavin Dean

Corey Rice

Samantha James

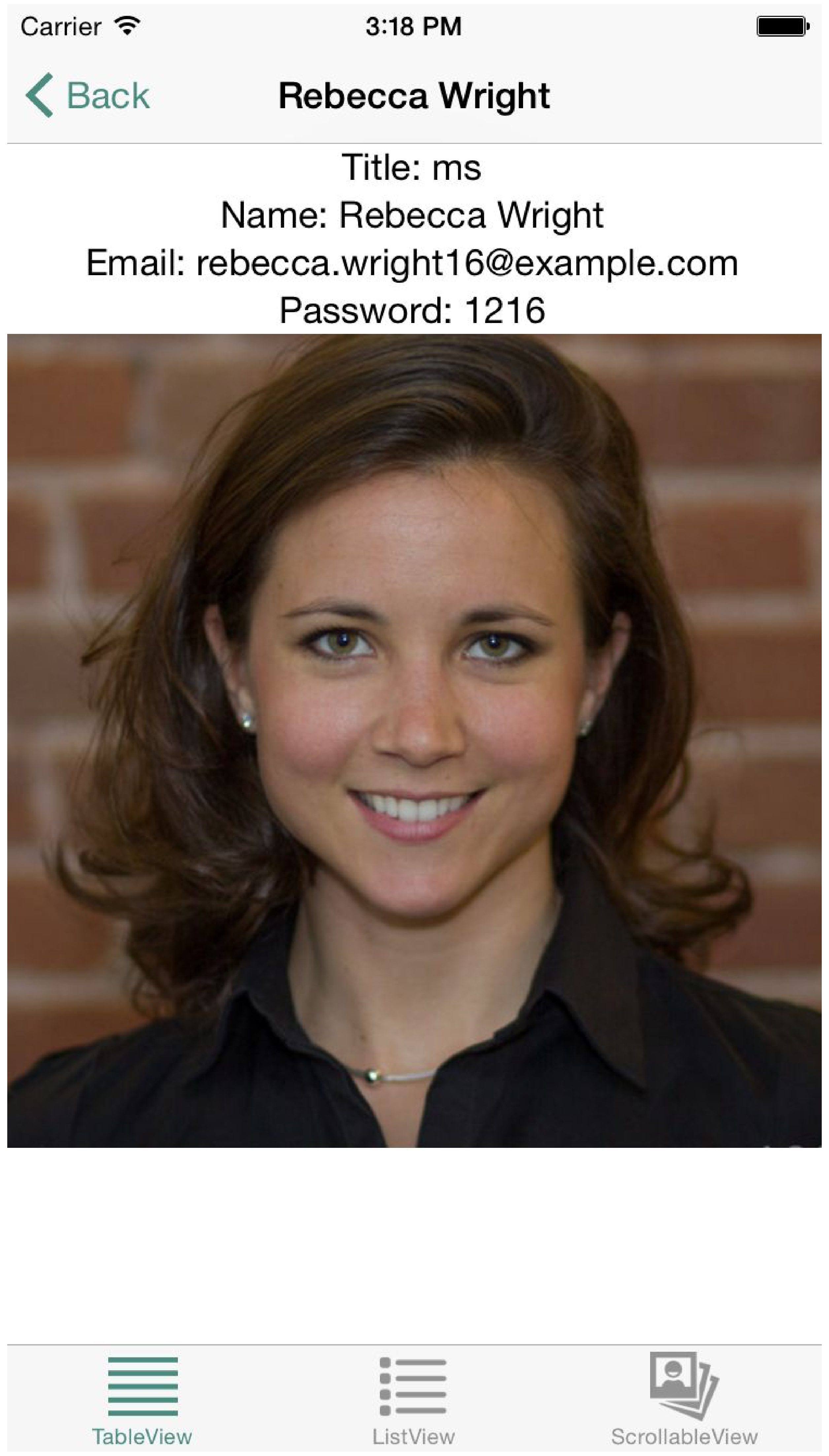
Jennifer Riley

View

```
<Collection src="randomuserme" />

<TableView dataCollection="randomuserme">

<TableViewRow onClick="rowClicked" rowId="{id}">
     <View layout="horizontal">
          <ImageView image="{ProfilePhoto}"/>
          <Label text="{Firstname} {Lastname}"/>
     </View>
</TableViewRow>
```

Controller

```
// Backbone fetch on Model
  Alloy.Collections.randomuserme.fetch();

// Clean up when window closes to avoid memory issues
function cleanup() {
    $.destroy();
}
```



Title: ms
Name: Rebecca Wright
Email: rebecca.wright16@example.com
Password: 1216

Filter a Collection

TableView View XML

Add Id to row via binding

```
<TableViewRow onClick="rowClicked" rowId="{id}">
```

TableView Controller

Pass Id to new Window

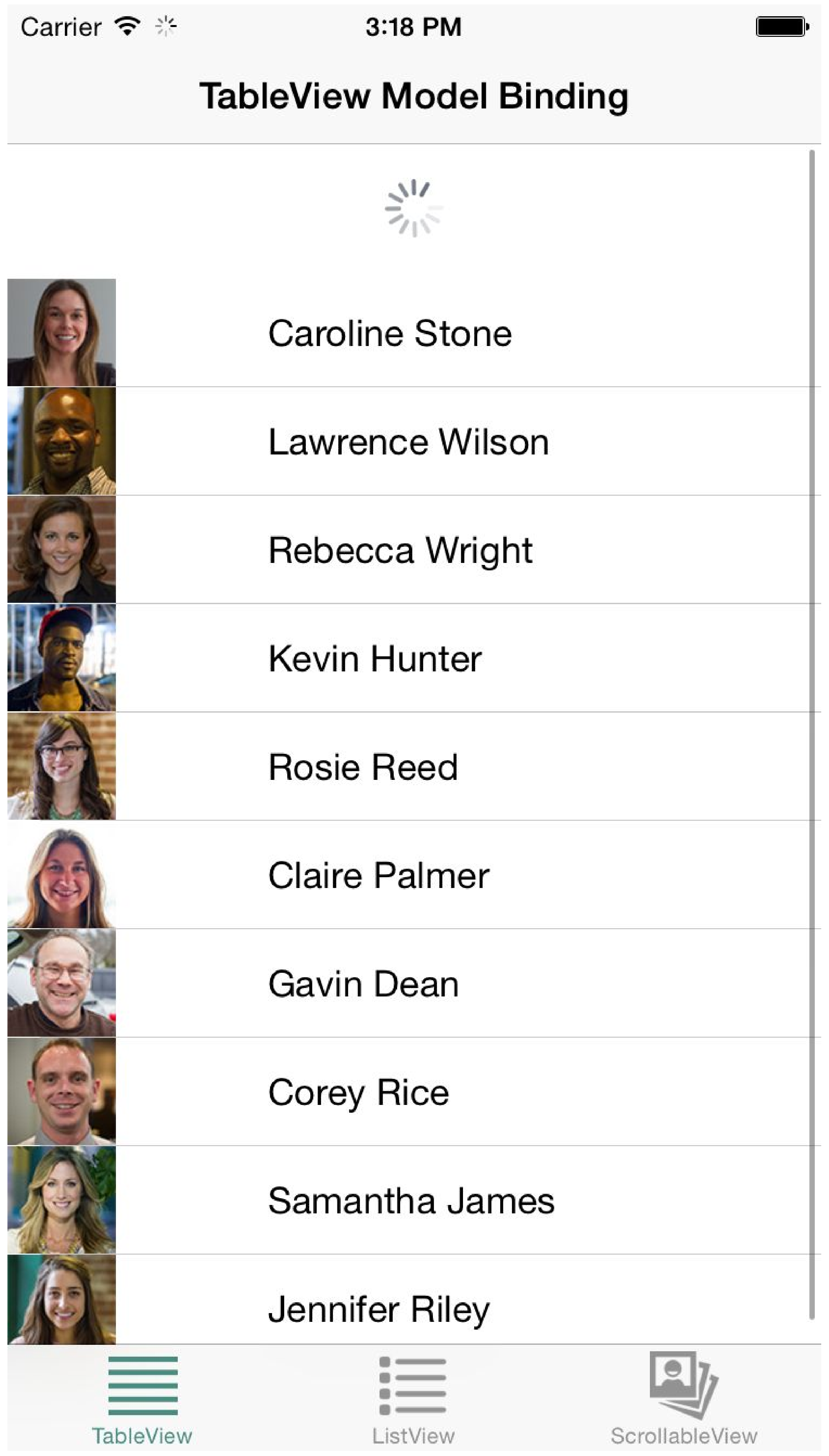
```
// get Id of row clicked, open a new window passing in Id
function rowClicked(e) {
    var id = e.rowData.rowId;
    var winPerson = Alloy.createController("person", {modelId: id}).getView();
    Alloy.Globals.tabTableView.open(winPerson);
}
```

Person Controller

Filter collection by Id

```
// Filter collection by Id passed over using Backbone
var person = Alloy.Collections.randomuserme.where({
    id: args.modelId
});
```

Bonus Feature - Pull to Refresh



Caroline Stone
Lawrence Wilson
Rebecca Wright
Kevin Hunter
Rosie Reed
Claire Palmer
Gavin Dean
Corey Rice
Samantha James
Jennifer Riley

TableView View XML

Add Refresh Widget

```
<Widget id="ptr" src="nl.fokkezb.pullToRefresh" onRelease="myRefresher">
```

TableView Controller

Make onRelease function do Backbone Fetch

```
// onRefresh make API call using Alloy Model via Backbone fetch
function myRefresher(e) {
    Alloy.Collections.randomuserme.fetch({
        success: e.hide,
        error: e.hide
    });
}
```

TableView Controller

onOpen of Window refresh Widget to set intial set of data

```
// onOpen of Window get API data by calling refresh
function init() {
    $.ptr.refresh();
}
```

Extra Stuff we didn't cover

Backbone Objects without Alloy

- You can use plain Backbone Collection and Model objects in place of Alloy versions

Migrations

- Migration is a description of incremental changes to a database, which takes your database from version 1 to version x.

- e.g. offline cache, do need to take this into account

Other UI Binding

- ButtonBar
- CoverFlow
- Map Annotations
- Picker
- TabbedBar
- Toolbar
- View

DataTransform

- Filter outside of model file

DataFunction

- Manually update View

Prefix Naming for Multiple Models in a View

Resources

- https://bitbucket.org/applification/alloymodelbinding
- http://docs.appcelerator.com/platform/latest/#!/guide/Alloy_Models
- https://github.com/appcelerator/alloy/tree/master/test/apps/models