Salesforce and iOS

Integration and some data handling tips

Create a Salesforce Project

- Use Terminal to build a starter project
  - Use **cd** to navigate to your project directory
  - **git clone https://github.com/forcedotcom/SalesforceMobileSDK-iOS.git**
  - **cd SalesforceMobileSDK-iOS**
  - **./install.sh**
    - This final step will pull all the dependencies that the SDK needs
    - Be sure to always open the project **SalesforceMobileSDK.xcworkspace**

Getting Started

- Visit https://trailhead.salesforce.com for free training on using Salesforce with Swift and iOS
- Some coding examples are available in Swift—check the webpage for links
- Begin your project by designing database schema using the Salesforce apps

Protocols for Easy Integration

- Make your structs conform to
  - Hashable
  - Identifiable
  - Codable
    - struct Customer: Hashable, Identifiable, Decodable {
      var id: String
      var name: String
      var brotherPrinters: [printers]
      }

Use Combine for UI Tasks

- ObservableObject
  - Automatically update your UI when updates are pushed from Salesforce
- SmartStore will manage your connection with Saleforce
- SyncManager will manage asynchronous data communication
- Mark your models with @Published to announce changes to listeners

Typical Fetch Request

```
class CustomersListModel: ObservableObject {

   @Published var customers: [Customer] = []

   private var customerCancellable: AnyCancellable?

   func fetchCustomers(){
      let request = RestClient.shared.request(forQuery: "SELECT Id, Name, Email FROM Customers ORDER BY Name ASC", apiVersion: nil)

      customerCancellable = RestClient.shared.publisher(for: request)
         .receive(on: RunLoop.main)
         .tryMap({ (response) -> Data in
            response.asData()
         })
         .decode(type: CustomerResponse.self, decoder: JSONDecoder())
         .map({ (record) -> [Customer] in
            record.records
         })
         .catch( { error in
            Just([])
         })
         .assign(to: \.customers, on:self)
   }
}
```