// Application programming interface

//fixer.io

JSON => JavaScript Object Notation

Google search JSON Beautifier => codebeautify.org/jsonviewer

```
// ViewController.swift
 import UIKit


class ViewController: UIViewController{
    @IBOutlet weak var cadLabel: UILabel!
    ↓ similar ones


    override func viewDidLoad(){
        super.viewDidLoad()
    }                                                            // 1. Request & session
                                                                   2. response & Data
                                                                   3. Parsing & JSON serialization

    @IBAction func getRatesClicked(_ sender:Any){
        let url = URL(string:"http://data.fixer.io/api/latest?access_key= ..... ")
        let session = URLSession.shared
                                            →//with completionHandler
        let task = session.dataTask(with:url!){ (data, response, error) in
            if error ! =nil{
                let alert = UIAlertController(title"Error", message: error?.LocalizedDescription, preferedStyle: UIAlertController.style.alert)
                let okButton = UIAlertAction(title:"Ok", style: UIAlertAction.Style.default, handler: nil()
                alert.addAction(okButton)
                self.present(alert, animated: true, completion: nil)          //p-list → App Transport Security Setting
            } else{                                                             ↓Allow Arbitrary Loads => Yes
                if data != nil{                                                 ↓ then can request http request
                    do{
                        let jsonResponse = try JSONSerialization.jsonObject(with:Data!, options: JSONSerialization.ReadingOptions.mutableContainers)
                            as! Dictionary<String,Any>
                        DispatchQueue.main.async{                                       //SYNC => one thing at a time
                            if let rates = jsonResponse["rates"] as? [string: Any]{      ASYNC => two things at one time
                                if let cad = rates["CAD"] as? Double{                   main
                                    self.cadLabel.text = "CAD: \(cad)"                  Global
                                }                                                       // round the string
                                ↓ similar ones                                          = String(format:"CAD: %.2f", cad)

                            }
                        }
                    } catch{
                        print("error")
                    }
                }
            }
        }
        task.resume()        // start the task
}
```

```
//Face Recognition
 import UIKit
 import LocalAuthentication

class ViewController : UIViewController{
   @IBOutlet var myLabel: UILabel!

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

   @IBAction func signInClicked(sender: AnyObject){
      let authContext = LAContext()   //come with the LocalAuthentication library
      var error : NSError?
                                                                          //convert to error Pointer
       if authContext.canEvaluatePolicy(.deviceOwnerAuthenticationWithBiometrics, error:&error){
          authContext.evaluatePolicy(.deviceOwnerAuthenticationWithBiometrics, localizedReason:"Is it You?"){ (success ,error) in
             if success == true{
                DispatchQueue.main.async{
                   self.performSegue(withIdentifier:"toSecondVC",sender:nil)
                }
             }else{
                DispatchQueue.main.async{
                   self.myLabel.text = "Error!"
                }
             }                                                     //p-list => Face ID Usage Description
          }
       }
   }
```