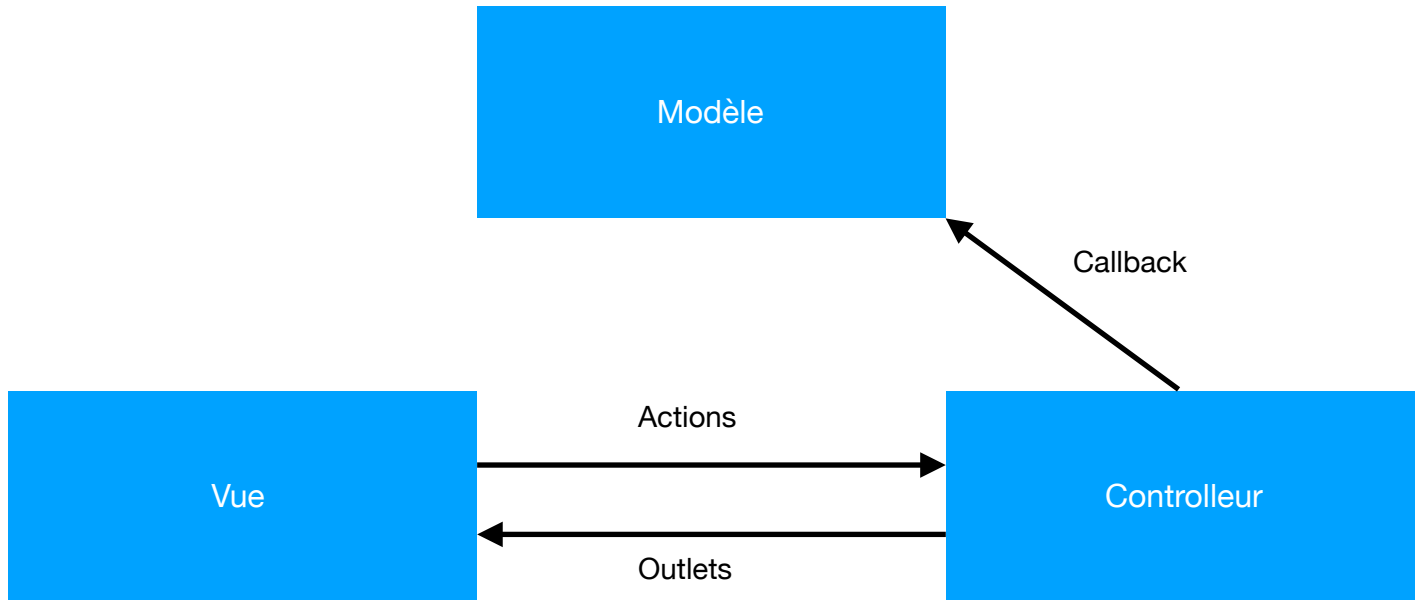


# PDF Baluchon

## Modèle MVC



## Bonus Baluchon

Pour le bonus j'ai décidé d'intégrer une fonction qui me permet de recueillir toutes les monnaies pour pouvoir les convertir en euros.

```
func getSymbols(callback: @escaping (Bool, [String]?) -> Void) {
    let fixerUrl = URL(string:
        "http://data.fixer.io/api/symbols?access_key=d595ac5936090b6eea37cd18e76aa86b")
    dataTask?.cancel()
    guard let url = fixerUrl else { return }
    dataTask = rateSession.dataTask(with: url) { (data, response, error) in
        guard let data = data, error == nil else {
            callback(false, nil)
            return
        }
        guard let response = response as? HTTPURLResponse, response.statusCode == 200
        else {
            callback(false, nil)
            return
        }
        guard let responseJSON = try? JSONDecoder().decode(Symbols.self, from: data)
        else {
            callback(false, nil)
            return
        }
        var currencies = [String]()
        responseJSON.symbols.forEach { (key, value) in
            currencies.append(key)
        }
        callback(true, currencies.sorted(by: <))
    }
    dataTask?.resume()
}
```

Ensuite j'ai créé grâce à Postman une structure decodable un tableau [String: String] des données des monnaies :

```
struct Symbols: Decodable {  
    let symbols: [String: String]  
}
```

Pour finir j'ai intégré dans mon contrôleur un picker view pour me permettre d'y intégrer toutes les monnaies et pouvoir choisir celle qu'on souhaite pour la convertir en euros :

```
private let exchangeRateService = ExchangeRateService()  
private var allRates = [String]()  
  
/// This is the currency conversion  
private var currency = "AED"  
  
// MARK: - View Life Cycle  
  
override func viewDidLoad() {  
    super.viewDidLoad()  
    ratesPickerView.dataSource = self  
    ratesPickerView.delegate = self  
    exchangeRateService.getSymbols { (success, symbols) in  
        if success, let symbols = symbols {  
            DispatchQueue.main.async {  
                self.allRates = symbols  
                self.ratesPickerView.reloadAllComponents()  
            }  
        }  
    }  
}
```

Mise en place du picker view :

```
// MARK: - UIPickerViewDelegate, UIPickerViewDataSource  
  
extension ExchangeRateViewController : UIPickerViewDelegate, UIPickerViewDataSource {  
  
    func numberOfComponents(in pickerView: UIPickerView) -> Int {  
        return 1  
    }  
  
    func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {  
        return allRates.count  
    }  
  
    func pickerView(_ pickerView: UIPickerView, titleForRow row: Int, forComponent component: Int) -> String? {  
        return allRates[row]  
    }  
  
    func pickerView(_ pickerView: UIPickerView, didSelectRow row: Int, inComponent component: Int) {  
        currency = allRates[row]  
    }  
}
```