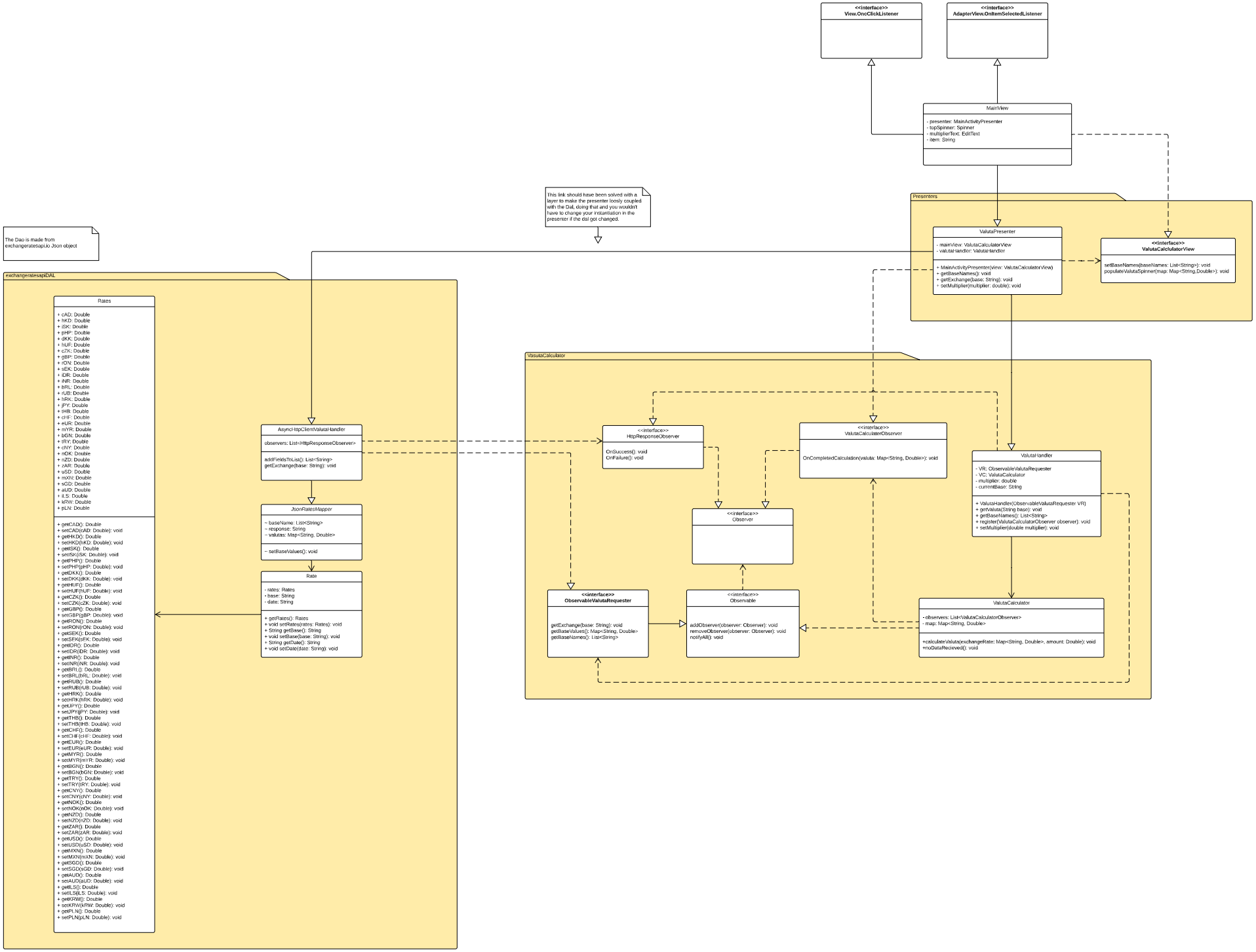
<<interface>>
View.OnclickListener
<<interface>>
AdapterView.OnItemSelectedListener
MainView
- presenter: MainActivityPresenter
- topSpinner: Spinner
- multiplierText: EditText
- item: String
This link should have been solved with a layer to make the presenter loosly coupled with the Dal, doing that and you wouldn't have to change your instantiation in the presenter if the dal got changed.
Presenters
ValutaPresenter
- mainView: ValutaCalculatorView
- valutaHandler: ValutaHandler
+ MainActivityPresenter(view: ValutaCalculatorView)
+ getBaseNames(): void
+ getExchange(base: String): void
+ setMultiplier(multiplier: double): void
<<interface>>
ValutaCalculatorView
setBaseNames(baseNames: List<String>): void
populateValutaSpinner(map: Map<String,Double>): void
The Dao is made from exchangeratesapi.io Json object
exchangeratesapiDAL
Rates
+ cAD: Double
+ hKD: Double
+ iSK: Double
+ pHP: Double
+ dKK: Double
+ hUF: Double
+ cZK: Double
+ gBP: Double
+ rON: Double
+ sEK: Double
+ iDR: Double
+ iNR: Double
+ bRL: Double
+ rUB: Double
+ hRK: Double
+ jPY: Double
+ tHB: Double
+ cHF: Double
+ eUR: Double
+ mYR: Double
+ bGN: Double
+ tRY: Double
+ cNY: Double
+ nOK: Double
+ nZD: Double
+ zAR: Double
+ uSD: Double
+ mXN: Double
+ sGD: Double
+ aUD: Double
+ iLS: Double
+ kRW: Double
+ pLN: Double
+ getCAD(): Double
+ setCAD(cAD: Double): void
+ getHKD(): Double
+ setHKD(hKD: Double): void
+ getISK(): Double
+ setISK(iSK: Double): void
+ getPHP(): Double
+ setPHP(pHP: Double): void
+ getDKK(): Double
+ setDKK(dKK: Double): void
+ getHUF(): Double
+ setHUF(hUF: Double): void
+ getCZK(): Double
+ setCZK(cZK: Double): void
+ getGBP(): Double
+ setGBP(gBP: Double): void
+ getRON(): Double
+ setRON(rON: Double): void
+ getSEK(): Double
+ setSEK(sEK: Double): void
+ getIDR(): Double
+ setIDR(iDR: Double): void
+ getINR(): Double
+ setINR(iNR: Double): void
+ getBRL(): Double
+ setBRL(bRL: Double): void
+ getRUB(): Double
+ setRUB(rUB: Double): void
+ getHRK(): Double
+ setHRK(hRK: Double): void
+ getJPY(): Double
+ setJPY(jPY: Double): void
+ getTHB(): Double
+ setTHB(tHB: Double): void
+ getCHF(): Double
+ setCHF(cHF: Double): void
+ getEUR(): Double
+ setEUR(eUR: Double): void
+ getMYR(): Double
+ setMYR(mYR: Double): void
+ getBGN(): Double
+ setBGN(bGN: Double): void
+ getTRY(): Double
+ setTRY(tRY: Double): void
+ getCNY(): Double
+ setCNY(cNY: Double): void
+ getNOK(): Double
+ setNOK(nOK: Double): void
+ getNZD(): Double
+ setNZD(nZD: Double): void
+ getZAR(): Double
+ setZAR(zAR: Double): void
+ getUSD(): Double
+ setUSD(uSD: Double): void
+ getMXN(): Double
+ setMXN(mXN: Double): void
+ getSGD(): Double
+ setSGD(sGD: Double): void
+ getAUD(): Double
+ setAUD(aUD: Double): void
+ getILS(): Double
+ setILS(iLS: Double): void
+ getKRW(): Double
+ setKRW(kRW: Double): void
+ getPLN(): Double
+ setPLN(pLN: Double): void
AsyncHttpClientValutaHandler
observers: List<HttpResponseObserver>
addFieldsToList(): List<String>
getExchange(base: String): void
JsonRatesMapper
~ baseName: List<String>
~ response: String
~ valutas: Map<String, Double>
~ setBaseValues(): void
Rate
- rates: Rates
- base: String
- date: String
+ getRates(): Rates
+ void setRates(rates: Rates): void
+ String getBase(): String
+ void setBase(base: String): void
+ String getDate(): String
+ void setDate(date: String): void
ValutaCalculator
<<interface>>
HttpResponseObserver
OnSuccess(): void
OnFailure(): void
<<interface>>
ValutaCalculaterObserver
OnCompletedCalculation(valuta: Map<String, Double>): void
<<interface>>
Observer
<<interface>>
ObservableValutaRequester
getExchange(base: String): void
getBaseValues(): Map<String, Double>
getBaseNames(): List<String>
<<interface>>
Observable
addObserver(observer: Observer): void
removeObserver(observer: Observer): void
notifyAll(): void
ValutaHandler
- VR: ObservableValutaRequester
- VC: ValutaCalculator
- multiplier: double
- currentBase: String
+ ValutaHandler(ObservableValutaRequester VR)
+ getValuta(String base): void
+ getBaseNames(): List<String>
+ register(ValutaCalculatorObserver observer): void
+ setMultiplier(double multiplier): void
ValutaCalculator
- observers: List<ValutaCalculatorObserver>
- map: Map<String, Double>
+calculateValuta(exchangeRate: Map<String, Double>, amount: Double): void
+noDataRecieved(): void