



## مقدمه

در این تمرین، قصدمان بر آن است که اپلیکیشنی پیاده سازی کنیم که قابلیت های زیر را داشته باشد:

۱. نمایش لیستی از ارز های دیجیتال

- نمایش نام کامل + نام مخفف + قیمت فعلی بر حسب دلار
- نمایش تصویر سکه
- نمایش تغییرات ۱ ساعت اخیر
- نمایش تغییرات ۲۴ ساعت اخیر
- نمایش تغییرات ۷ روزه اخیر

۲. نمایش نمودار OHLC های روزانه ارز مدنظر در اثر کلیک شدن روی آن.

در ادامه API هایی معرفی خواهد شد که خدمات لازم جهت به دست آوردن موارد بالا را ارائه می دهند. برای ارسال درخواست به این API ها از OkHttp۳ و برای برقراری ارتباط با UI Thread باید از handler استفاده کنید. در ادامه نحوه ی استفاده از آن ها و پیاده سازی اپلیکیشن توضیح خواهیم داد.

یک نمونه از نحوه ارسال درخواست به کمک OkHttp۳ ضمیمه شده است. (نمونه درخواست دریافت candle ها)








## شرح اپلیکیشن

این اپلیکیشن از دو صفحه تشکیل شده است. صفحه اول شامل لیستی از ارز های دیجیتال است. توجه کنید که تعداد ارز های دیجیتال زیاد است و لذا نباید اطلاعات آن ها را همگی در لحظه اول بگیرید (چون در این صورت دچار requests too many می شوید) برای حل این مشکل ابتدا تعدادی مثلا ۱۰ سکه را درخواست می دهید و سپس در انتهای این لیست، یک عدد button بگذارید که اگر روی آن کلیک شود ۱۰ سکه دیگری لود گردد. همچنین توجه کنید که در محل مناسبی در صفحه یک دکمه برای reload کردن صفحه بذارید که اگر کاربر روی آن کلیک کند دیتا مجدداً لود شود. قبل از لود شدن سکه ها یک progressBar در صفحه قرار دهید تا کاربر حس نکند صفحه فریز شده است.

برای طراحی ابتدا نیاز است تا یک سلول از آن را پیاده سازی کنید و پس از دریافت داده ها، لیستی از آن ها را ایجاد کرده و در layout صفحه قرار دهید.

**توجه:** در این تمرین قصد پرداختن به UI به صورت حرفه ای نداریم و شما موظف هستید به کمک المان های زیر به طراحی بپردازید:

- LinearLayout
- Button
- TextView
- ImageView

	BTC   Bitcoin	112368
	1h: 0% 1D: -2% 7D: 2%	
	ETH   Ethereum	3888
	1h: 0% 1D: -2% 7D: 22%	
	XRP   XRP	08
	1h: 0% 1D: -5% 7D: 32%	
	USDT   Tether	1\$
	1h: 0% 1D: 0% 7D: 0%	
	BCH   Bitcoin Cash	288\$
	1h: 0% 1D: -4% 7D: 0%	
	BSV   Bitcoin SV	224\$
	1h: -1% 1D: -4% 7D: 5%	
	LTC   Litecoin	58\$
	1h: 0% 1D: -3% 7D: 3%	

شکل ۱ : نمونه صفحه اول

حال که سکه ها لود شده است، شخص در اثر کلیک کردن روی هر کدام از سکه ها به صفحه دوم می رود که در آن لیستی که در هر سطر از لیست، اطلاعات یک OHCL candle در آن نوشته شده است، منظور از اطلاعات هر شمع چهار عدد Open، Close، High و Low است.

کاربر می تواند انتخاب کند که candle های یک هفته اخیر یا candle های یک ماه اخیر یعنی ۳۰ عدد کندل را مشاهده کند. یک button برای مورد اول و یک button دیگر برای مورد دوم در صفحه به طور مناسب قرار دهید.

**امتیازی:** می توانید به جای نمایش عددی candle ها، آن ها را در نموداری مناسب رسم کنید. می توانید از کتابخانه [MPAndroidChart](#) استفاده کنید. مثال



شکل ۲ : نمونه ای از نمودار ساده Candle stick

## مدیریت حافظه و شبکه

در هر بار استفاده از اپ اگر شخص به اینترنت متصل بود که داده های به روز از اینترنت دریافت و نشان داده می شود و اگر به اینترنت متصل نبود باید آخرین دیتا هایی که در دفعه قبلی اتصال به اینترنت، دریافت شده بود نشان داده شود. طبیعتاً در دفعه اول که اگر کاربر اینترنت نداشته باشد نباید چیزی نشان داده شود و خوب است اگر پیام مناسبی به شخص داده شود. ذخیره سازی اطلاعات و نیز فراخوانی آن ها نباید در ترد اصلی انجام شود.

## دریافت اطلاعات به کمک API

در این تمرین از API های زیر برای تحقق نیازمندی های تمرین استفاده کنید:

- [CoinMarketCap](#)

دریافت اطلاعات coin ها (نام، تصویر و تغییرات قیمت)  
اطلاعات لازم برای صفحه اول را از این API به دست می آورید.

- [Coin.io](#)

اطلاعات لازم برای رسم نمودار صفحه دوم و داده های OHLC را از این API بگیرید.

نکات مهم :

- برای راحتی کار یک تابع جهت زدن درخواست و دریافت Candle stick ها، ضمیمه شده است.
- تمامی درخواست های سمت شبکه باید توسط ترد های worker انجام شود و نتایج آن ها به کمک Handler به UI اضافه می شود.
- برای لود کردن تصویر از کتاب خانه [Glide](#) استفاده کنید.
- در مواقعی که لود کردن دارید حتما progressBar بگذارید.
- رخ داد های حساس به ایجاد race condition را به دقت مدیریت کنید.
- از ایجاد درخواست های تکراری سمت شبکه پرهیز کنید و درخواست ها و داده های دریافتی را مدیریت کنید. (مثلا مدام کلیک کردن روی دکمه های صفحه دوم نباید درخواست های مکرر سمت شبکه ایجاد کند) نتایج را پس از اولین دریافت کش کنید).
- [Weak Reference](#) را مطالعه کنید و در Handler از آن استفاده کنید.
- در هر جا که نیاز به Thread دارید همانجا new Thread ننزید و از [Thread pool Executor](#) برای هندل کردن تسک های پارالل استفاده کنید (این کار overhead درخواست سخت افزاری ایجاد Thread به ازای هر تسک پارالل را دیگر ندارد)

## نحوه ی تحویل تمرین

- برای تمرین یک ریپازیتوری در گیت هاب ایجاد کنید تا اعضای گروه کدها را طبق میزان مشارکت خود پوش کنند.
- پس از پایان تمرین یک تگ با نام hw۱ ریلیز کنید
- تنها کامیت های قبل از تگ بالا در نظر گرفته میشود.
- تحویل و نمره دهی نهایی به تمرین به صورت تحویل حضوری (مجازی) انجام می شود.

سلامت باشید