View binding

ویو بایندینگ آخرین و بهترین روش برای دسترسی به ویوها هست.

ضعف find view by id برای دسترسی به view ها این بود که فشار زیادی به cpu, gpu وارد میکرد، چون برای دسترسی به view ها ما هر تعداد لایه که داشتیم میومد بررسی میکرد تا بتونه به view مورد نظر دسترسی پیدا کنه.

بعدش باترنایف اومد که توسط جیک وارتون معرفی شد.

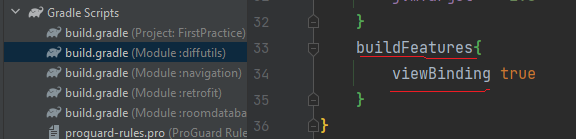
بعد kotlin extension معرفی شد، که بدون معرفی کردن view میشد ازش استفاده کرد که خیلی زیاد خطای null pointer exception میداد چون view هنوز ساخته نشده بود ولی لایه اش اجرا میشد و ارور null pointer exception میداد.

بعد data binding اومد که یکسری محدودیت ها داره که مثلاً بخوایم چندین حالت رو توی یک view قرار بدیم اذیت میکنه و حالت داینامیکیش کمتر.

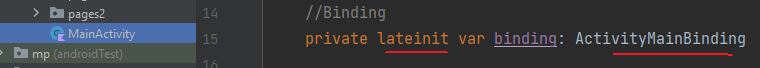
در حال حاضر view binding معرفی شده و خود گوگل هم توصیه میکنه که از view binding استفاده کنیم.

برای استفاده از view binding میایم یه کد به gradle اضافه میکنیم، بعد موقع استفاده view binding میاد برای دسترسی به view ها فقط توی همون صفحه ایی که هستیم سرچ میکنه و کل view هایی که داریم رو در تمام لایه هامون بررسی نمیکنه و با توجه به دسترسی که بهش به لایه مورد نظر میدیم فقط میاد ویوهای اون لایه xml رو بررسی میکنه.

از نظر پرفرمنس و سرعت خیلی عالیه.



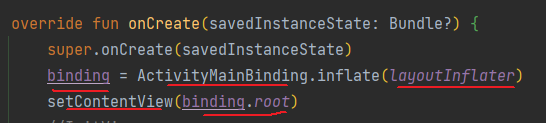
برای استفاده از view binding میایم داخل gradle module و داخل قسمت android و این کد رو اضافه میکنیم.



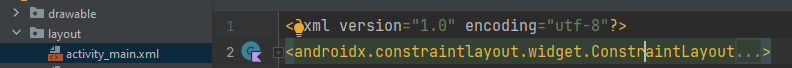
برای استفاده از view binding میایم بصورت lateinit تعریف میکنیم چون اول null هست و ما نمیدونیم قرار با چه مقداری پر بشه.

اسمش رو binding یا هرچیزی که دوست داشتیم میزاریم و نحوه ساختن به این صورت هست که میاد یه کلاس میسازه که ما توسط این کلاس به فایل و لایه xml مون دسترسی داریم و مثلاً کلاس Main Activity ما فایل xml ش میشه main\_activity که موقع تعریف view binding میشه کلاس Acitvity Main Binding .

بعد میریم متغییر binding رو داخل متد on create و initialize میکنیم:



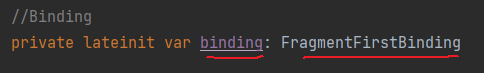
که از layout inflater که جز موارد app compat activity هست و به ما اجاز میده که بتونیم توی فایل های کاتلین و کلاس هامون از لایه ها استفاده کنیم.



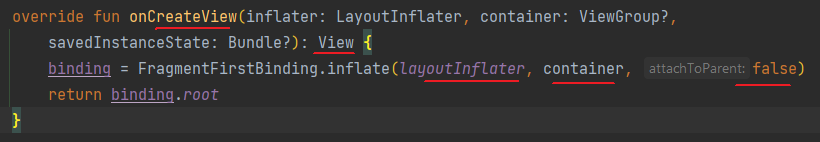
و binding.root به parent و اولین لایه ی ما داخل اون فایل xml .



برای دسترسی به تک تک view هایی که داخل لایه هامون قرار داره باید از binding استفاده کنیم.

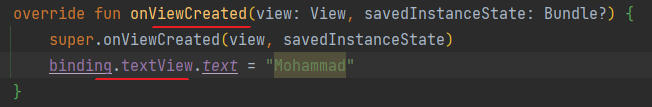


توی fragment هم به این صورت استفاده میکنیم.



داخل فرگمنت میایم توی on create view و binding رو initialize میکنیم.

بخاطر nullable بهتر که علامت سوال view رو برداریم.



موقع استفاده هم کدهارو داخل متد on view created مینویسیم که خیالمون راحت باشه که دیگه view ساخته شده و error null pointer exception نمیده.

منبع:

استاد عزیزم آقای مهندس نوری و سایت نوری آکادمی، دوره نخبگان معماری

[آموزش کامل برنامه نویسی اندروید با نوری آکادمی - نوری آکادمی (nouri.academy)](https://nouri.academy/)