

在 Android 的 Kotlin 项目中使用 Glide 添加自定义请求头，可以通过实现一个自定义的 LazyHeaders 或 OkHttp 集成来实现。以下是两种方法：

方法 1：使用 LazyHeaders

Glide 提供了 LazyHeaders.Builder，可用于为每个请求添加自定义头。

```
val imageUrl = "https://example.com/image.jpg"

// 构建 Glide 的请求
Glide.with(context)
    .load(GlideUrl(imageUrl, LazyHeaders.Builder()
        .addHeader("Authorization", "Bearer your_token")
        .addHeader("Custom-Header", "HeaderValue")
        .build()))
    .into(imageView)
```

注意：

- LazyHeaders.Builder 可以动态添加多组请求头。
- 如果使用静态的 URL，可以直接通过 GlideUrl 和 LazyHeaders 配合。

方法 2：使用 OkHttpClient 自定义拦截器

如果需要更复杂的网络请求处理（例如动态请求头），可以将 Glide 与 OkHttp 集成。

1. 添加 Glide 的 OkHttp 集成库

在 build.gradle 中添加依赖：

```
implementation "com.github.bumptech.glide:okhttp3-  
integration:4.15.1" // 确保 Glide 版本一致
```

2. 创建 OkHttpClient 添加请求头

```
import com.bumptech.glide.Glide  
import  
com.bumptech.glide.integration.okhttp3.OkHttpUrlLoader  
import com.bumptech.glide.load.model.GlideUrl  
import okhttp3.Interceptor  
import okhttp3.OkHttpClient  
import okhttp3.Response  
import java.io.InputStream  
  
// 创建自定义拦截器  
class HeaderInterceptor : Interceptor {  
    override fun intercept(chain: Interceptor.Chain):  
Response {  
        val originalRequest = chain.request()  
        val newRequest = originalRequest.newBuilder()  
            .addHeader("Authorization", "Bearer  
your_token")  
            .addHeader("Custom-Header", "HeaderValue")  
            .build()  
        return chain.proceed(newRequest)  
    }  
}  
  
// 创建 OkHttpClient  
val okHttpClient = OkHttpClient.Builder()  
    .addInterceptor(HeaderInterceptor())  
    .build()  
  
// 注册 OkHttpClient 与 Glide  
Glide.get(context).registry.replace(  
    GlideUrl::class.java, InputStream::class.java,  
    OkHttpUrlLoader.Factory(okHttpClient)  
)  
  
// 使用 Glide 加载图片
```

```
val imageUrl = "https://example.com/image.jpg"
Glide.with(context)
    .load(imageUrl)
    .into(imageView)
```

哪种方法更好？

- **方法 1** 简单轻量，适用于固定请求头的场景。
- **方法 2** 灵活强大，适合动态请求头或需要自定义网络行为的场景（如超时、缓存）。

根据项目需求选择合适的方法即可。