

Glide 的 `preload()` 方法默认是异步的，并且不会直接提供回调。如果你需要在 Glide 预加载完成后获得回调，可以使用 Glide 的 `RequestListener` 来实现。以下是具体方法：

使用 `RequestListener` 实现预加载并回调

示例代码：

```
import com.bumptech.glide.Glide
import com.bumptech.glide.request.RequestListener
import com.bumptech.glide.request.target.Target

fun preloadImageWithCallback(context: Context, imageUrl:
String, onComplete: (Boolean) -> Unit) {
    Glide.with(context)
        .load(imageUrl)
        .listener(object : RequestListener<Any> {
            override fun onLoadFailed(
                e: GlideException?,
                model: Any?,
                target: Target<Any>?,
                isFirstResource: Boolean
            ): Boolean {
                // 加载失败回调
                onComplete(false)
                return false // 返回 false 表示让 Glide 继
续处理错误状态
            }

            override fun onResourceReady(
                resource: Any?,
                model: Any?,
                target: Target<Any>?,
                dataSource:
com.bumptech.glide.load.DataSource?,
                isFirstResource: Boolean
            ): Boolean {
                // 加载成功回调
                onComplete(true)
            }
        })
}
```

```

        return false // 返回 false 表示让 Glide 继续处理加载的资源
    }
})
    .preload() // 预加载图片
}

```

调用示例：

```

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

preloadImageWithCallback(context, imageUrl) { success ->
    if (success) {
        println("Image preloaded successfully!")
    } else {
        println("Image preload failed!")
    }
}

```

批量预加载并回调

如果需要批量预加载一组图片并在全部完成后回调，可以使用计数器逻辑：

示例代码：

```

fun preloadImagesWithCallback(
    context: Context,
    imageUrls: List<String>,
    onComplete: (successCount: Int, failureCount: Int) ->
Unit
) {
    var successCount = 0
    var failureCount = 0
    val totalImages = imageUrls.size

    for (imageUrl in imageUrls) {
        Glide.with(context)
            .load(imageUrl)
    }
}

```

```

        .listener(object : RequestListener<Any> {
            override fun onLoadFailed(
                e: GlideException?,
                model: Any?,
                target: Target<Any>?,
                isFirstResource: Boolean
            ): Boolean {
                failureCount++
                if (successCount + failureCount ==
totalImages) {
                    onComplete(successCount,
failureCount)
                }
                return false
            }

            override fun onResourceReady(
                resource: Any?,
                model: Any?,
                target: Target<Any>?,
                dataSource:
com.bumptechn.glide.load.DataSource?,
                isFirstResource: Boolean
            ): Boolean {
                successCount++
                if (successCount + failureCount ==
totalImages) {
                    onComplete(successCount,
failureCount)
                }
                return false
            }
        })
        .preload()
    }
}

```

调用示例:

```

val imageUrls = listOf(
    "https://example.com/image1.jpg",
    "https://example.com/image2.jpg",

```

```

        "https://example.com/image3.jpg"
    )

    preloadImagesWithCallback(context, imageUrls)
    { successCount, failureCount ->
        println("Preloading completed. Success:
    $successCount, Failures: $failureCount")
    }

```

使用带有尺寸的预加载并回调

如果你需要为特定尺寸的图片预加载，也可以结合 `override()` 方法：

示例代码：

```

fun preloadImageWithSizeAndCallback(
    context: Context,
    imageUrl: String,
    width: Int,
    height: Int,
    onComplete: (Boolean) -> Unit
) {
    Glide.with(context)
        .load(imageUrl)
        .override(width, height)
        .listener(object : RequestListener<Any> {
            override fun onLoadFailed(
                e: GlideException?,
                model: Any?,
                target: Target<Any>?,
                isFirstResource: Boolean
            ): Boolean {
                onComplete(false)
                return false
            }

            override fun onResourceReady(
                resource: Any?,
                model: Any?,

```

```
        target: Target<Any>?,
        dataSource:
com.bumptech.glide.load.DataSource?,
        isFirstResource: Boolean
    ): Boolean {
        onComplete(true)
        return false
    }
    })
    .preload()
}
```

调用示例：

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

preloadImageWithSizeAndCallback(context, imageUrl, 200,
200) { success ->
    if (success) {
        println("Image with specified size preloaded
successfully!")
    } else {
        println("Failed to preload image with specified
size!")
    }
}
```

注意事项

1. **缓存机制：** Glide 的预加载会将图片存储在内存缓存或磁盘缓存中，因此即使没有立即显示，后续加载速度会更快。
2. **性能优化：**
 - 批量预加载时，避免一次性加载过多图片，建议分批处理。
 - 结合 RecyclerView 滑动监听动态预加载图片。

3. **线程安全**：RequestListener 回调会在主线程中触发，确保更新 UI 安全。