**Использование отладчика Android Studio по максимуму**

Это хитрость, о которой я узнал совсем недавно у Senior Android разработчика в моей компании, и теперь я чувствую себя несчастным, сожалея о времени, которое я провел в ожидании сборки Gradle, чтобы проверить свои изменения при написании Android приложений.

Вот типичный сценарий, который каждый Android разработчик мог бы встретить, по крайней мере, один раз в течение своего жизненного цикла разработки. У вас есть список элементов, которые вы хотите показать в **ListView** или **RecyclerView**.

Ниже приведен наш возлюбленный метод onBindViewHolder, который связывает вашу модель с вашими view-компонентами **RecyclerView**.

@Override

public void onBindViewHolder(ViewHolder holder, final int position) {

final String name = values.get(position);

holder.txtHeader.setText(name);

holder.txtFooter.setText("Footer: " + name);

}

Теперь, допустим, вы захотели изменить цвет текста для каждого третьего элемента в списке. Таким образом, код будет выглядеть примерно так:

@Override

public void onBindViewHolder(ViewHolder holder, final int position) {

final String name = values.get(position);

holder.txtHeader.setText(name);

if (position % 3 == 0) {

holder.txtHeader.setTextColor(Color.GREEN);

}

holder.txtFooter.setText("Footer: " + name);

}

Затем вы нажмёте **Run** и дождётесь завершения сборки и увидите ваши изменения, так?

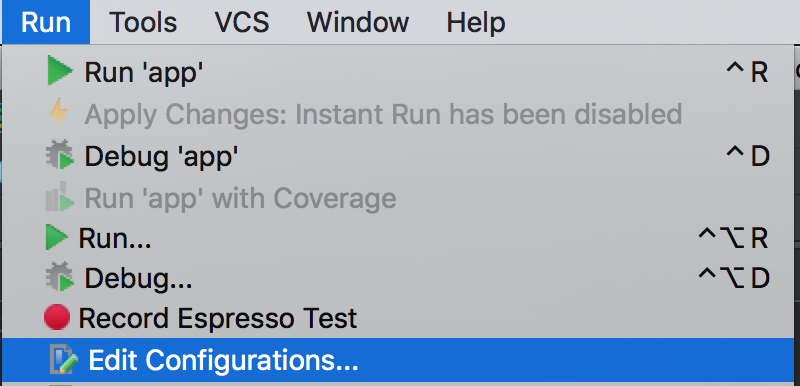
Теперь вы бы подумали, существует ли другой путь для достижения этой же цели?

Твой выход, Android Studio! Да, нам не нужен внешний плагин или инструмент для достижения вышеупомянутой задачи и более того, нам даже не придется заново собирать проект. Вы не ослышались, мы обойдёмся без Gradle :) Вот как!

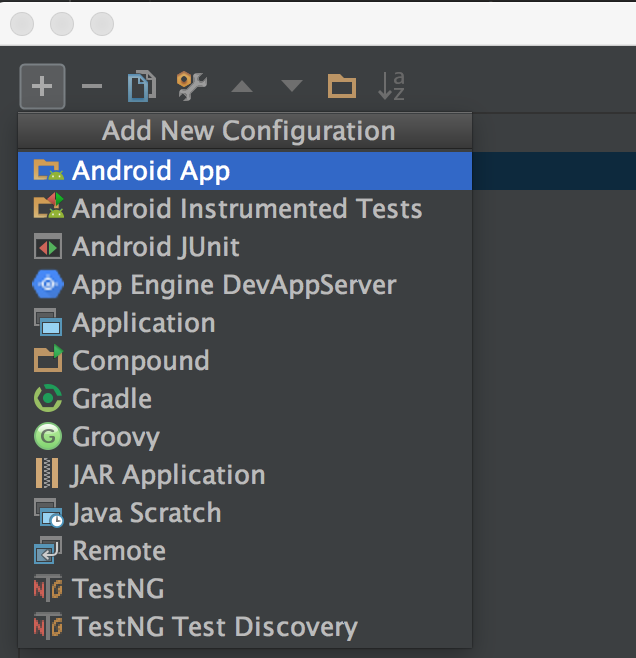
**Шаг 1 — Необходимо определить конфигурацию запуска**

Такая конфигурация запуска позволит нам запускать наше приложение и присоединять к нему отладчик из Android Studio, а также вы сможете присоединить его к уже запущенному процессу.

Нажмите **Run → Edit Configurations**.

[](https://camo.githubusercontent.com/ad87f4cbc14f172dbb8df841ffa11cf15e4757d0/68747470733a2f2f646576636f6c696272692e636f6d2f63702f77702d636f6e74656e742f75706c6f6164732f323031372f31312f312d64614c6445486d54506370547461654667765a5a6b672e706e67)

В верхнем левом углу диалогового окна щелкните значок «+» и выберите **Android App**.

[](https://camo.githubusercontent.com/f39f6914c335af5005128628a8e34ddd76049b34/68747470733a2f2f646576636f6c696272692e636f6d2f63702f77702d636f6e74656e742f75706c6f6164732f323031372f31312f312d4b6658554d774c4e586c654c424361304a79496d4c412e706e67)

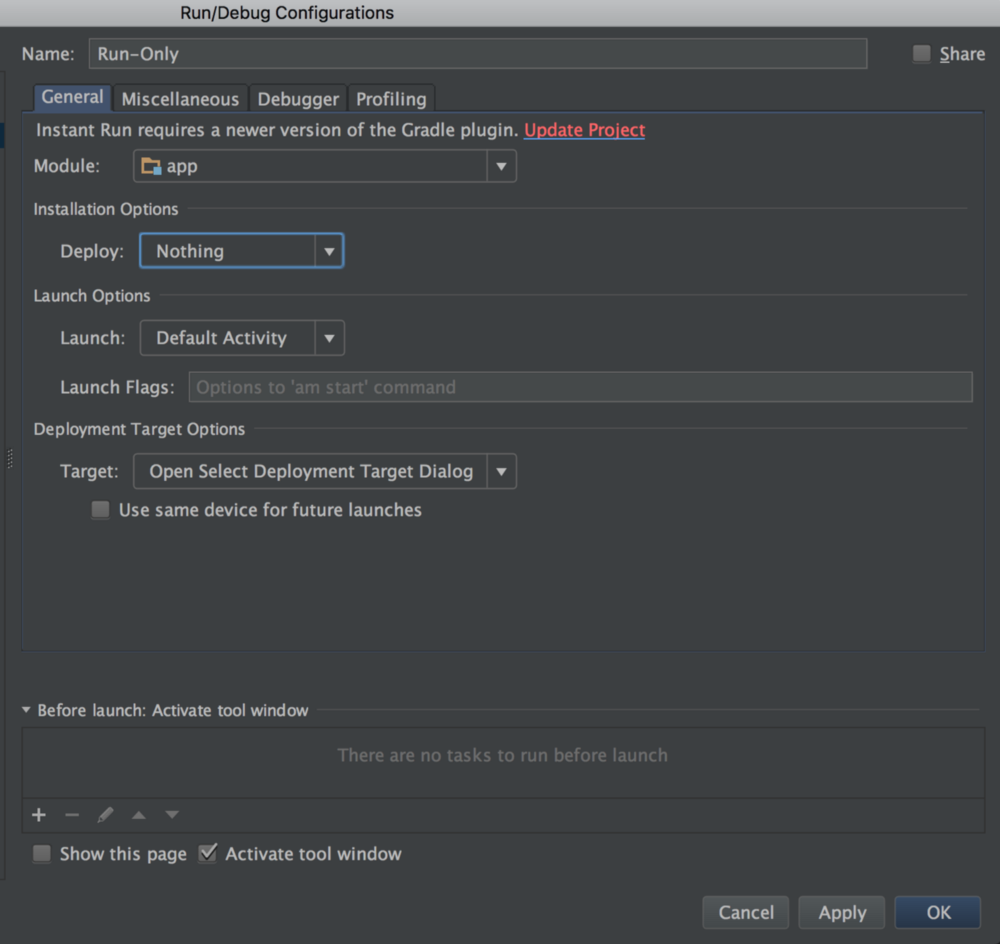
Теперь дайте ему имя, мне нравится называть его **Run-Only**, но вы можете называть его как угодно.

Выберите модуль, который содержит код вашего приложения, в приведенном ниже скриншоте он называется *app*.

**Важный шаг:**

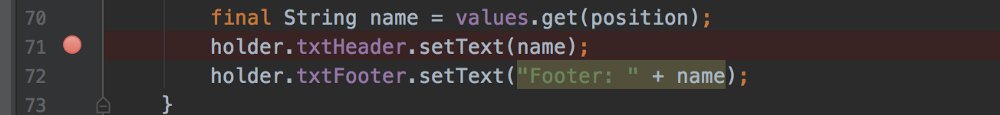
В разделе Installation Options выберите **Nothing**;  
В Launch Options выберите **Default Activity**;  
В разделе Before Launch удалите **Gradle-aware Make**.

Таким образом, конфигурация должна выглядеть следующим образом:

[](https://camo.githubusercontent.com/dee99fb634c27bd62dc0e6518f52086dae41a34c/68747470733a2f2f646576636f6c696272692e636f6d2f63702f77702d636f6e74656e742f75706c6f6164732f323031372f31312f312d78686e4878786266336b4643766a30583174516442672e706e67)

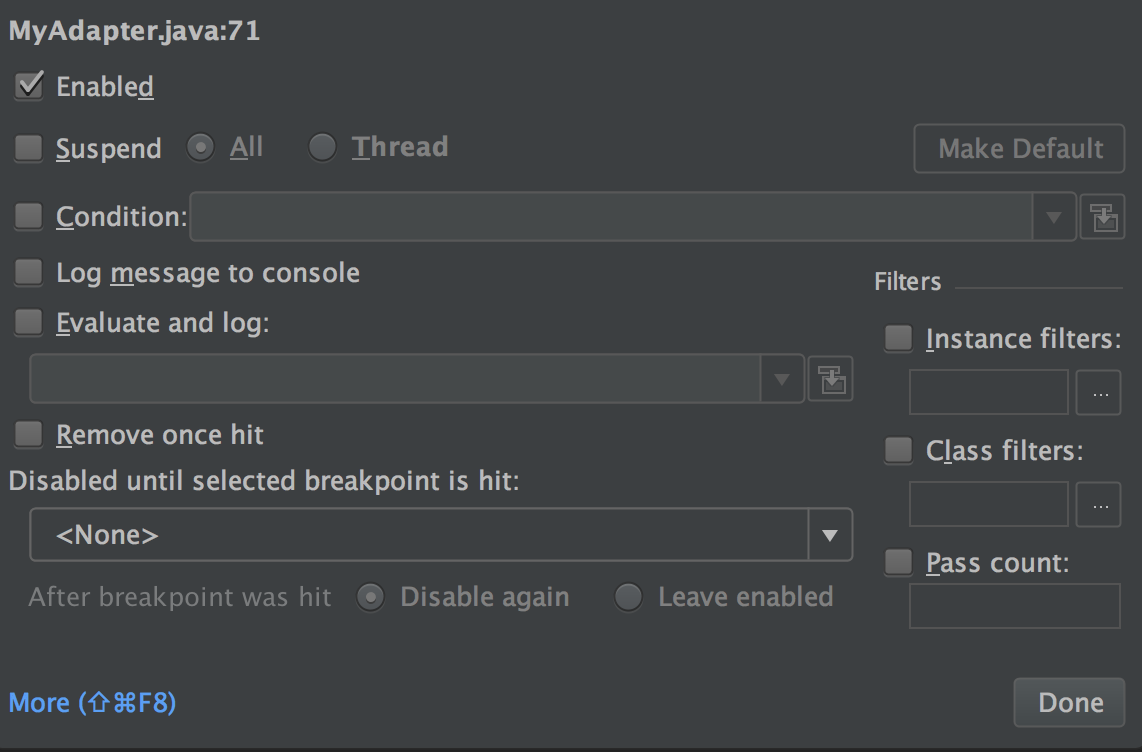
Теперь вы можете применить и сохранить эту конфигурацию. Сейчас она должна быть выбрана автоматически. Если нет, то просто выберите её из списка.

Теперь поставьте *breakpoint* поближе к строке, которую вы хотите проверить. В нашем случае мы разместим ее там, где мы устанавливаем текст.

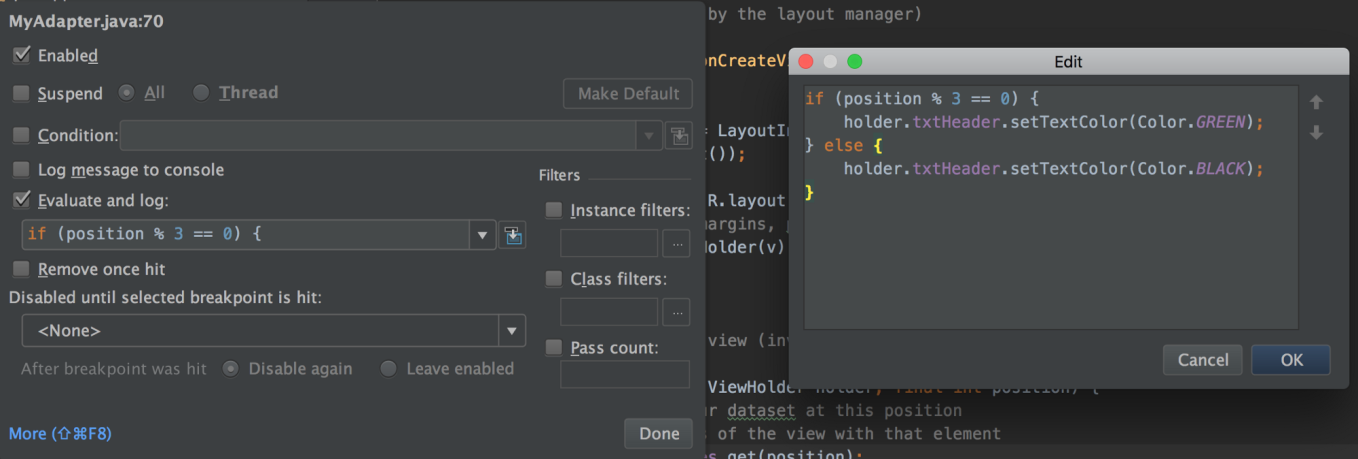
[](https://camo.githubusercontent.com/96e96ce4dcddd9bf90c37a4003cd817fc5a29816/68747470733a2f2f646576636f6c696272692e636f6d2f63702f77702d636f6e74656e742f75706c6f6164732f323031372f31312f312d4a6841733762542d3951764b347267547943637931672e706e67)

Щелкните правой кнопкой мыши на *breakpoint* и снимите флажок **Suspend** (рус. *приостановить*).

Как только вы снимете флажок, диалог расширится и покажет больше опций.

[](https://camo.githubusercontent.com/b098c0dd09b9409f9798a8d853621dab0fba470d/68747470733a2f2f646576636f6c696272692e636f6d2f63702f77702d636f6e74656e742f75706c6f6164732f323031372f31312f312d34614148786c45426534594947594d436862426263512e706e67)

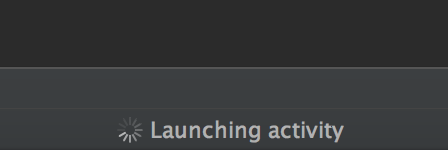
Нам интересен раздел **Evaluate and log**. Мы напишем там выражение, чтобы проверить изменения в нашем элементе RecyclerView. Нажмите на маленький значок голубого цвета справа от окна ввода *Evaluate and log*, чтобы развернуть его до более крупного редактора, и добавьте выражение для тестирования, и нажмите **Ok**, а затем **Done**.

[](https://camo.githubusercontent.com/793501f5cfb0cc3c6b2e4b9a502590770febd45b/68747470733a2f2f646576636f6c696272692e636f6d2f63702f77702d636f6e74656e742f75706c6f6164732f323031372f31312f312d5f656648776b7268446378596a395272706e515339672e706e67)

Теперь нажмите на иконку **Debug** с выбранной конфигурацией *Run-Only* и посмотрите на эту магию.

[Debug](https://camo.githubusercontent.com/534446b33857b36da5554ed54b2ce747437bd2f1/68747470733a2f2f646576636f6c696272692e636f6d2f63702f77702d636f6e74656e742f75706c6f6164732f323031372f31312f312d494246776358646e56414862513257436638494941412e706e67)

Приложение должно запуститься с вашей Activity по умолчанию, и вы должны увидеть внесенные там изменения. Также, если вы уделяете пристальное внимание IDE, в самом низу вы увидите только одну запускаемую задачу: *Launching Activity*.

[](https://camo.githubusercontent.com/02288640a6f2186ff8a53f5630e61ff481858f1f/68747470733a2f2f646576636f6c696272692e636f6d2f63702f77702d636f6e74656e742f75706c6f6164732f323031372f31312f312d62764b4a655f525f304645704b4e4a2d34765f5476512e706e67)

Хотелось бы услышать ваши впечатления, когда вы опробуете эту хитрость!

*Перевод статьи*[*«Making the most out of Android Studio Debugger»*](https://proandroiddev.com/making-the-most-out-of-the-android-studio-debugger-61713131d065)