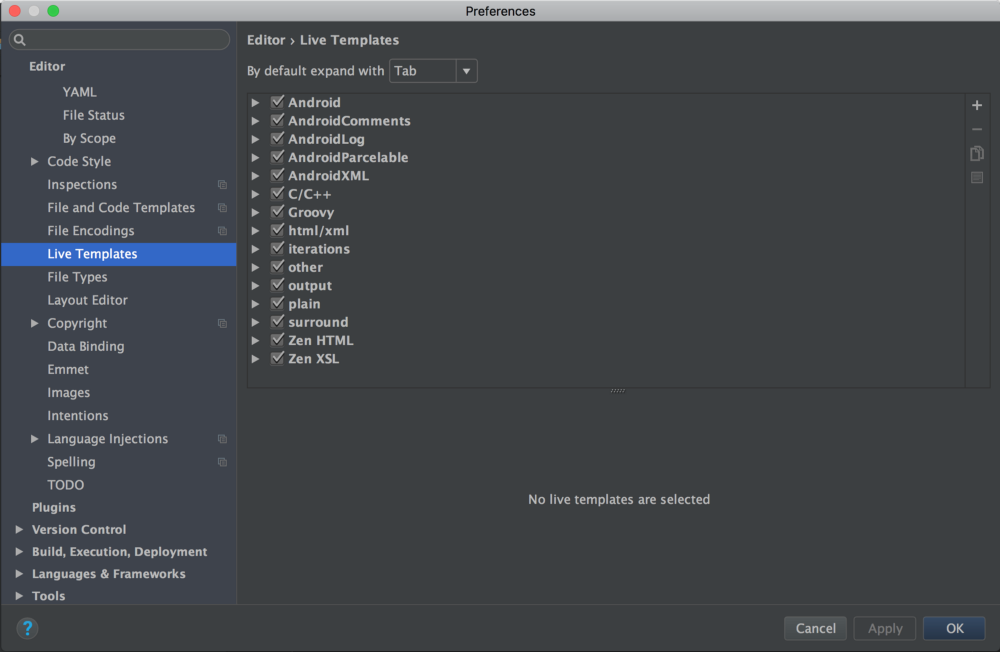
**Live Templates**

Code completion can improve your productivity by reducing how much you have to type,

Live Templates are code snippets that I can insert into my code by typing their abbreviation and pressing tab. They live in the Editor section of the Preferences.

Android Studio has already a defined set of *Live Templates*you can use*.*You can find the list of the available templates inside the Live Templates entry in the Android Studio preferences.



For example, type  psfi  and press enter in Android Studio. It will automatically generate the words public static final int. That saves you 19 keystrokes! There are already some predefined live templates that you can use:

* logm  – logs your method name being called with its arguments that were passed to it. (I find this one the most useful)
* psfs  – Replaces with public static final String
* .null  – When you want to surround your code with a null check, instead of typing if ( item == null){ } . Type item followed by .null , then press enter. This will surround the object with a null check!

This is awesome right? What is even better is the ability to create your own Live Templates.

A template contains hard-coded text and placeholder tokens (or variables). The parts that are marked by the $ character on either end are variables and normally would be the things that I’d be expected to type in

For example, one of the built-in templates looks like this:

for(int $INDEX$ = 0; $INDEX$ < $LIMIT$; $INDEX$++) {

$END$

}

Here, we have three variables $INDEX$, $LIMIT$ and $END$.

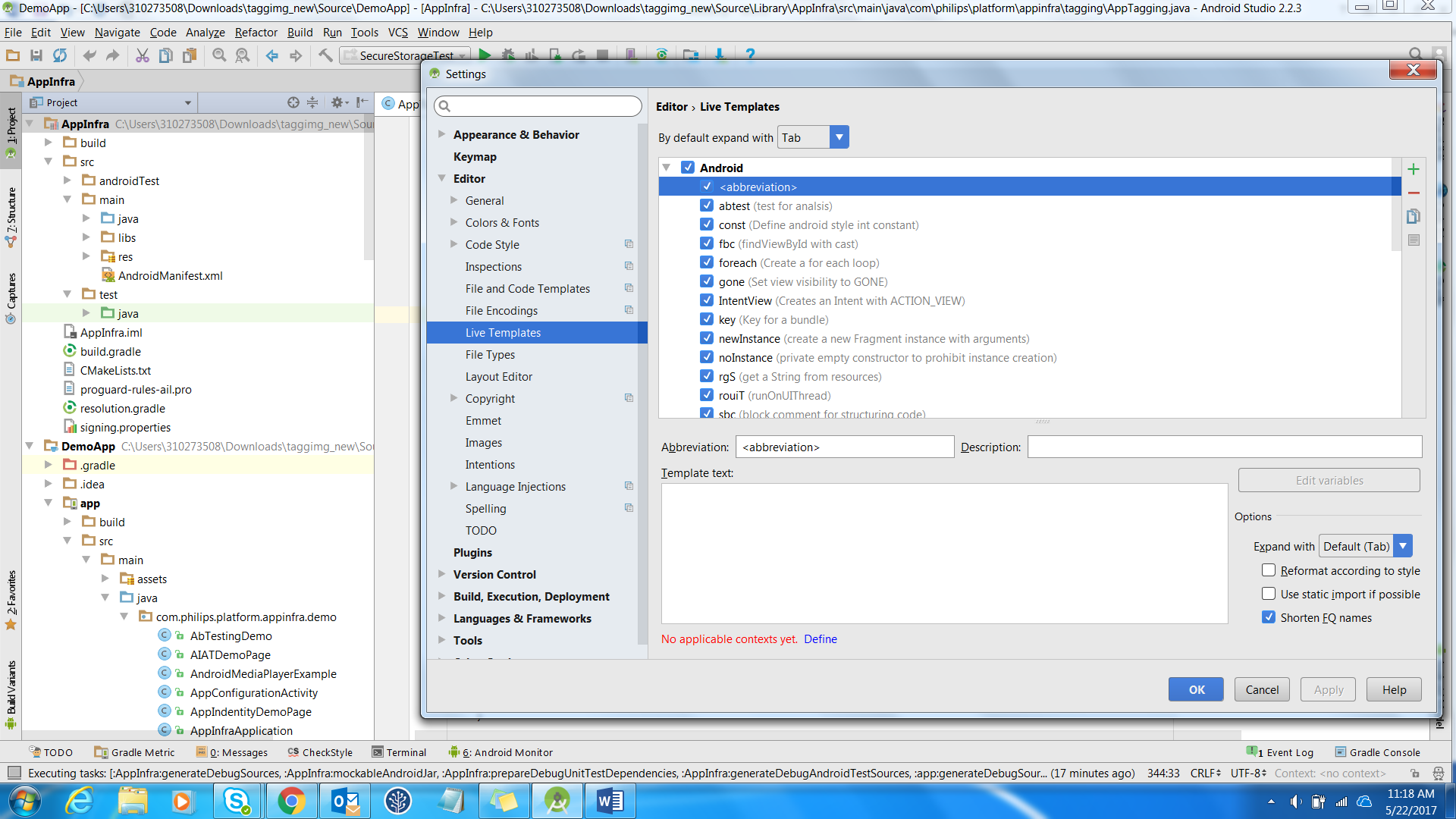
* $END$ is a special predefined variable that controls where your cursor will be placed once you’re done filling out the template. We will have to fill out the values of the other two.
* To use this template, I will type fori in the Java file and press tab. Android Studio will expand the template and put the cursor on the first variable that needs to be replaced. In this case, it will put the cursor where the template has the $INDEX$ token.
* As I type something, the other two places where $INDEX$ appears will copy what I’m typing, live!
* When I’m done naming the index variable, I will press tab or return and the cursor will move to the next variable that needs to be defined, which is $LIMIT$.
* Finally, after I finish typing in the $LIMIT$ variable, pressing tab or return will place the cursor where the $END$ variable is and will stop the template fill-out session.

**Write your own template**

You might have bits of code that are very specific to your project or, simply, they are not already on the list mentioned before. So you decide to create your own one, but how do I do that?

Let’s say we don’t have the *starter* template already and we would like to create it.

1. Go to the *Live Templates*tab from the Android Studio preferences or Go to File->Settings-> Editor->Live Templates.
2. Expand Android (With the help of Live Templates we can create various templates like Android, AndroidComments, AndroidLog, AndroidParcelable , AndroidXML , Bash , C/C++, CMake , Groovy , html/xml etc.). Once expand we can see all pre-defined templates.
3. Create your own template clicking on the + icon on the top right.



1. Select live templates and click
2. Write the abbreviation you would like to use and optionally add a description.

For example: AppInfraInitialization.

1. Add Some Description for reference. For example : AppInfra initialization
2. Define the context where this template should work, e.g.*Java, Groovy, Json.*
3. Write the code!

For example:

AppInfraInterface gAppInfra = new AppInfra.Builder().build(getApplicationContext());

AppInfra mAppInfra== (AppInfra)gAppInfra;

1. Use the Option for better code Reformate according to style.
2. Click Apply and Ok

**How to Use Live Template:**

1. Just type abbreviation (key) it will auto suggest and enter.
2. What value of code you tagged to particular abbreviation (key) it will appear automatically

For example: just type AppInfraInitialization. It will auto suggest AppInfraInitialization, and once Enter that below code will appear.

AppInfraInterface gAppInfra = new AppInfra.Builder().build(getApplicationContext());

AppInfra mAppInfra== (AppInfra)gAppInfra;

**How to distribute Live Template:**

1. Goto File->Export Settings.

2. Select Live Templates

3. Click OK, then ask for overwrite click Ok.

4. Click Show in explorer

5. settings.jar jar file will updated .

6. Share this .jar who have to use your template.

**How to use someone distributed settings.jar:**

1.Goto File->Import Settings

2.It will browse for settings .jar path, select proper settings.jar file (you taken from someone else shared settings.jar).

3. You can type abbreviation and use live templates shared by others.

**For more Information How to create live templates?**

<https://www.youtube.com/watch?v=LqIAg7EOkh0>

<https://android.jlelse.eu/android-studio-tips-for-productivity-4-59ca36b93c5a>