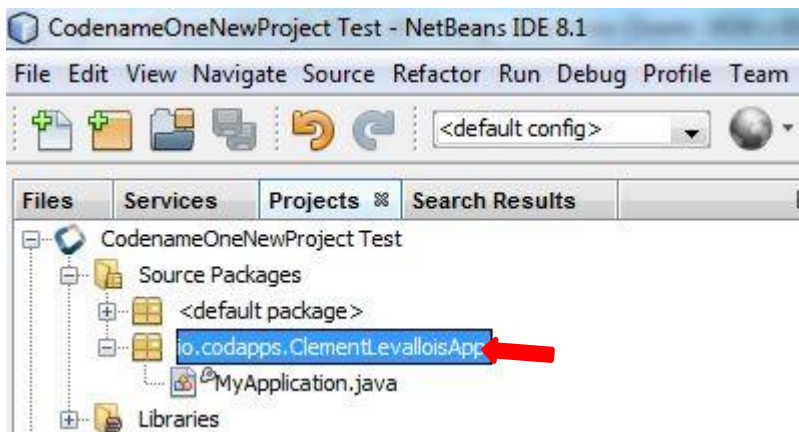


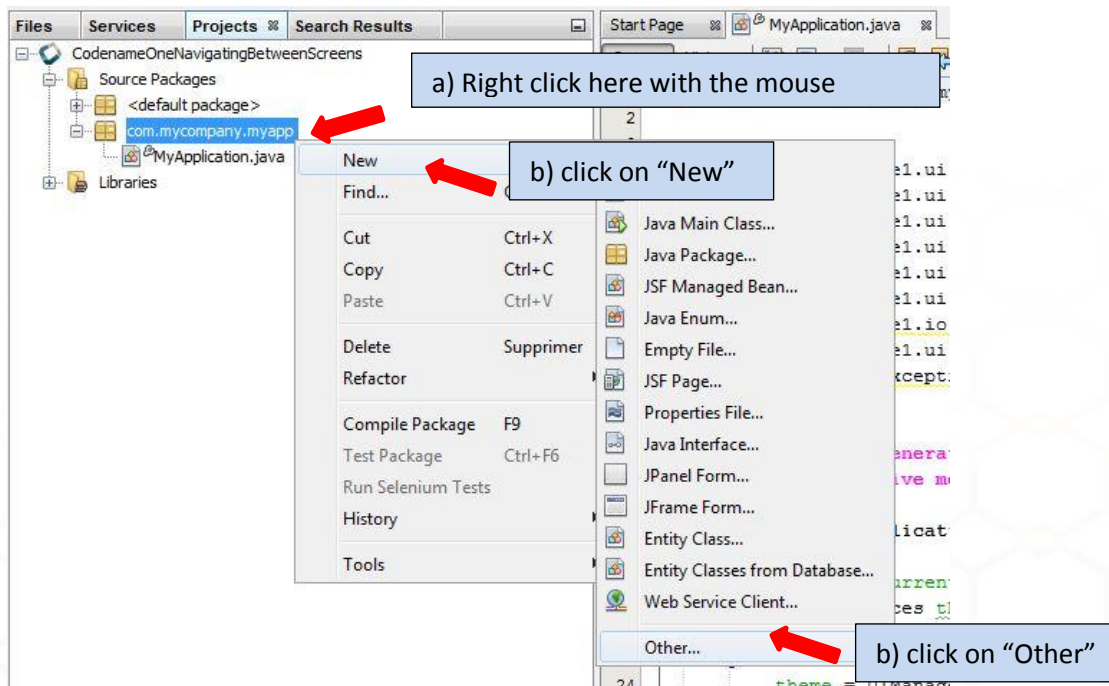
How to add the first screen to your app

To add the first screen to your mobile app, we need to create it. Screens are called “Form” in Codename One, the tool we use here. So let’s create a Form (a screen).

1. Double click on the name of your project to expand its content. Click on the folder that has the name of your package name. It should be highlighted in blue.

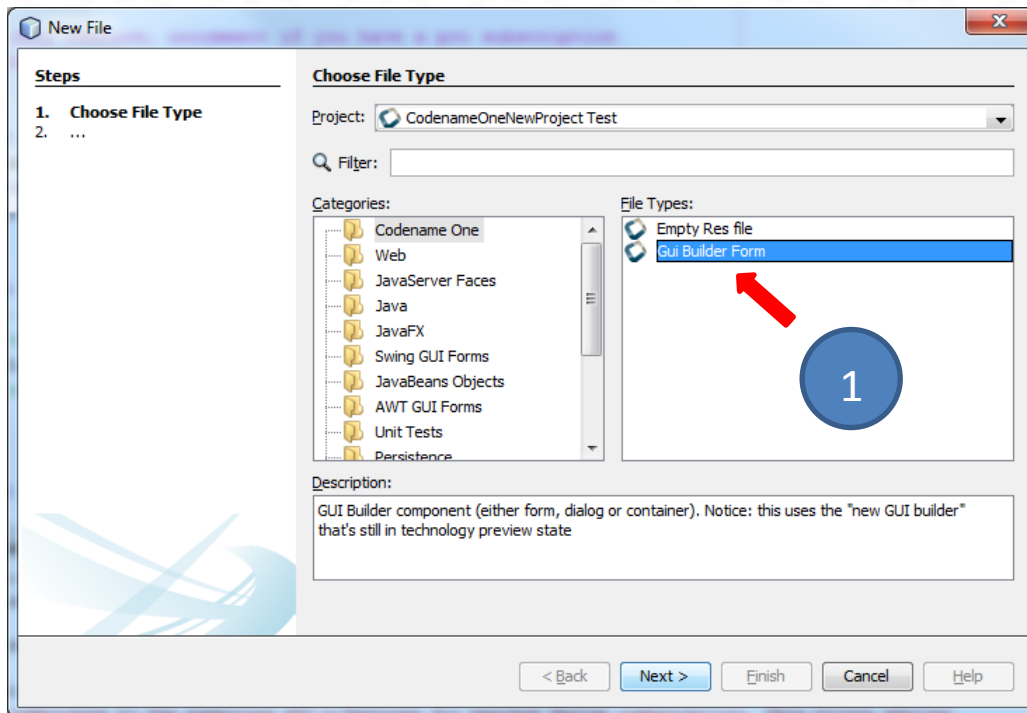


2. Right click on this folder, then click “New” then “Other”:

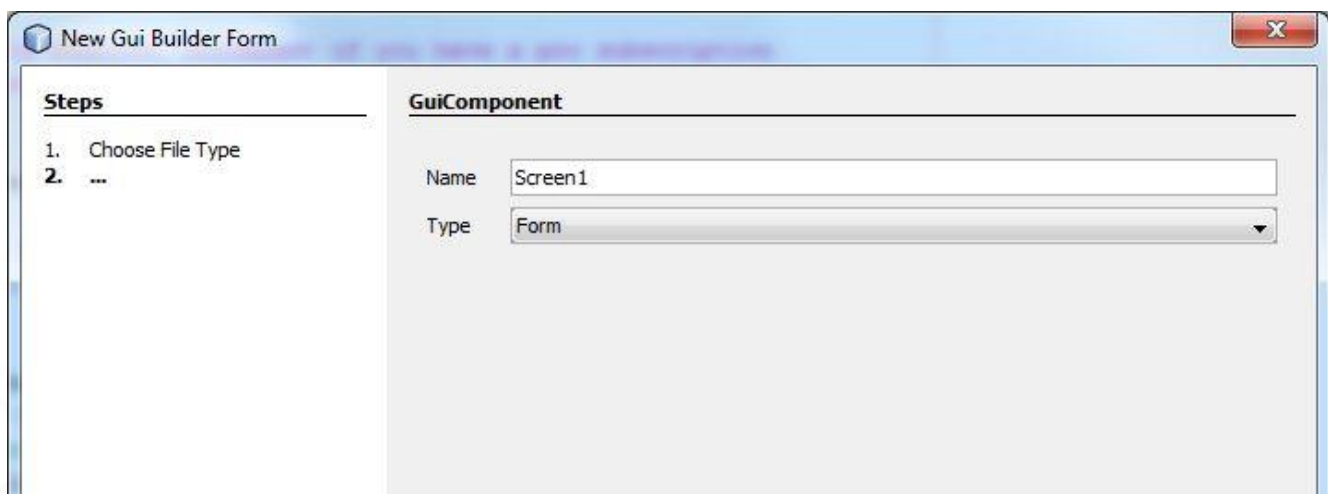


How to add the first screen to your app

3. Select “GUI Builder Form” then click on “Next”:



4. Pick a name for your screen. **As always, don't use spaces, special characters or accents like é ö or à in the name!**
I suggest you call it simply Screen1 (don't forget the capital S!). Then click “Finish”



Module 2: How to add text, pics, links etc. to your app

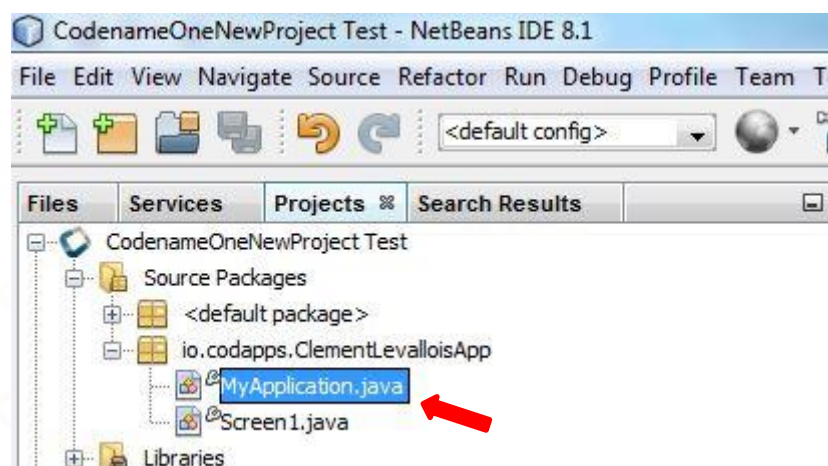
Level of difficulty: ● ○ ○ ○
Estimated time: 5 m

How to add the first screen to your app

5. You can see now that a screen called “Screen1” has been added to the files of your mobile app:



6. Now, a bit of coding. We will study how to code in module 5, but we need here to write just three lines of code. It does not matter if you don't understand the logic. **Basically, these lines of code will tell the app: show “Screen1” when the user of the phone opens the app.**
- a) Double-click on the file “MyApplication.java” to see its content
You will see the content on this file appear on the right (see next page).



Module 2: How to add text, pics, links etc. to your app

Level of difficulty: ● ○ ○ ○
Estimated time: 5 mn

How to add the first screen to your app

b) You should now see the content of the file “MyApplication.java” on the right:

```
1 package io.codagps.ClementLevalloisApp;
2
3
4 import com.codename1.ui.Display;
5 import com.codename1.ui.Form;
6 import com.codename1.ui.Dialog;
7 import com.codename1.ui.Label;
8 import com.codename1.ui.plaf.UIManager;
9 import com.codename1.ui.util.Resources;
10 import com.codename1.io.Log;
11 import com.codename1.ui.ToolBar;
12 import java.io.IOException;
13
14 /**
15  * This file was generated by <a href="https://www.codenameone.com/">Codename One</a> for the purpose
16  * of building native mobile applications using Java.
17  */
18 public class MyApplication {
19     private Form current;
20     private Resources theme;
21
22     public void init(Object context) {
23         theme = UIManager.initFirstTheme("/theme");
24
25         // Enable Toolbar on all Forms by default
26         Toolbar.setGlobalToolbar(true);
27
28         // Pro only feature, uncomment if you have a pro subscription
29         // Log.bindCrashProtection(true);
30
31     }
32
33     public void start() {
34         if(current != null){
35             current.show();
36             return;
37         }
38     }
39 }
```

Lines of code from the file
« MyApplication.java »

c) Now, scroll down in the file and find this bit of code. Note the opening and closing curly braces:

```
32
33 public void start() {
34     if(current != null){
35         current.show();
36         return;
37     }
38     Form hi = new Form("Hi World");
39     hi.addComponent(new Label("Hi World"));
40     hi.show();
41 }
42
```

d) Delete everything between the curly braces:

```
32
33 public void start() {
34
35 }
36
```


Module 2: How to add text, pics, links etc. to your app

Level of difficulty: ● ○ ○ ○
Estimated time: 5 mn

How to add the first screen to your app

- e) Write your first 3 lines of code inside the curly braces!
- careful: don't forget to capitalize letters where needed, don't forget dots and commas... Coding is an art of precision, where every spelling mistake is a bug!
 - also, we use Screen1 here because that is the name of the screen we created. You should use the name of the screen you created, visible in the files on the left!

```
32  
33 public void start() {  
34  
35     Screen1 box;  
36     box = new Screen1();  
37     box.show();  
38  
39 }  
40
```

That's it, well done! You have created the first screen of your app, and you have written the instructions telling the app to show the screen when the mobile app starts!

BONUS 1: What is the meaning of the lines of code?

You will see that in detail in module 5, but here is a preview:

Screen1 box;



In programming, we use a lot of boxes to describe things and actions. **Here, we create a box, specialized in containing objects of the type « Screen1 ».** Because boxes can contain only one type of objects, and we have to tell which type when we create the box. The semi-colon is just to say that we end the sentence.

box = new Screen1();



Now, we create a Screen1, and we put it in the box. So, the equal sign means: "put what is on the right inside the box which is on the left".

Module 2: How to add text, pics, links etc. to your app

Level of difficulty: ● ○ ○ ○
Estimated time: 5 mn

How to add the first screen to your app

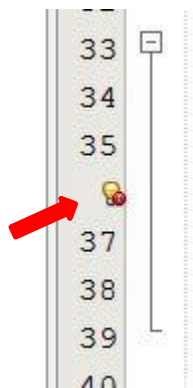
`box.show();`



We have a box with an object inside (a screen). We can now apply actions to the screen. To do that, we write the name of the box containing the object, then we write a dot, and after that the name of the action. And a semi-colon to finish. This instructs the mobile app to show on the phone the screen inside the box.

You might have noticed that when we type the dot after “box”, and if we wait a second, NetBeans shows a “menu” of actions. It is convenient: it means we don’t have to know by heart all the actions that we can apply to objects. We can lookup in the menu and find the one we need.

BONUS 2: Why the red and yellow sign which appears sometimes in the left margin?



```
public void start() {  
  
    Screen1 box;  
    box = new screen1();  
    box.show();  
  
}
```

The sign represents a light bulb with a red alert. It means there is an error in your code, and NetBeans has detected it for you.

You have two options:

- correct the error yourself: here, can you guess? The mistake is that it should our screen is called Screen1(), not screen1().
- Or, click on the light bulb in the left margin, read the suggestion by NetBeans and click on it to apply it. Here, NetBeans suggests to create a class “screen1” (a class is an object). Very bad idea in this case, we need the “Screen1” object we created, not a newly created, empty, irrelevant “screen1” object.