Introduction to Java

The Java Language

- Java was published in 1995 by James Gosling of Sun Microsystems
- Java is strongly typed (data types required) like C and C++ but utilizes a virtual machine (JVM) to run on multiple devices (portability). Also has ‘automatic garbage collection’
- C# was later announced by Microsoft in 2000, many say to compete with Java!

A History of Java

The Java programming language was designed and implemented by a small team of people headed by **James Gosling** at Sun Microsystems in Mountain View, California throughout the early 1990s.

Java/Sun was acquired by Oracle Corporation in 2010.

https://www.oracle.com/java/moved-by-java/timeline/

A History of Java

The original team worked on designing software for consumer electronics.

They quickly found that existing programming languages, e.g. C and C++ were not adequate.

James was deterred by the lack of memory deallocation (garbage collection), use of pointers (access to memory addresses) in C++, and lack of portability.

Programs written in C and C++ had to be compiled for a particular computer chip. When a new chip came out the software had to be re-compiled to make full use of new features in the chip.

https://www.oracle.com/java/moved-by-java/timeline/

A History of Java

In 1990 James Gosling started the design of a new programming language that was meant to be more appropriate for consumer electronics, without the problems of traditional languages such as C and C++.

This project, called the Green Project, resulted in the development of a computer language which Gosling called Oak after an oak tree outside his office window at Sun.

But it was later discovered that there was already a computer language called Oak (Oak Technologies). When a group of Sun people visited a local coffee shop, the name Java was suggested - it stuck and the rest is history.

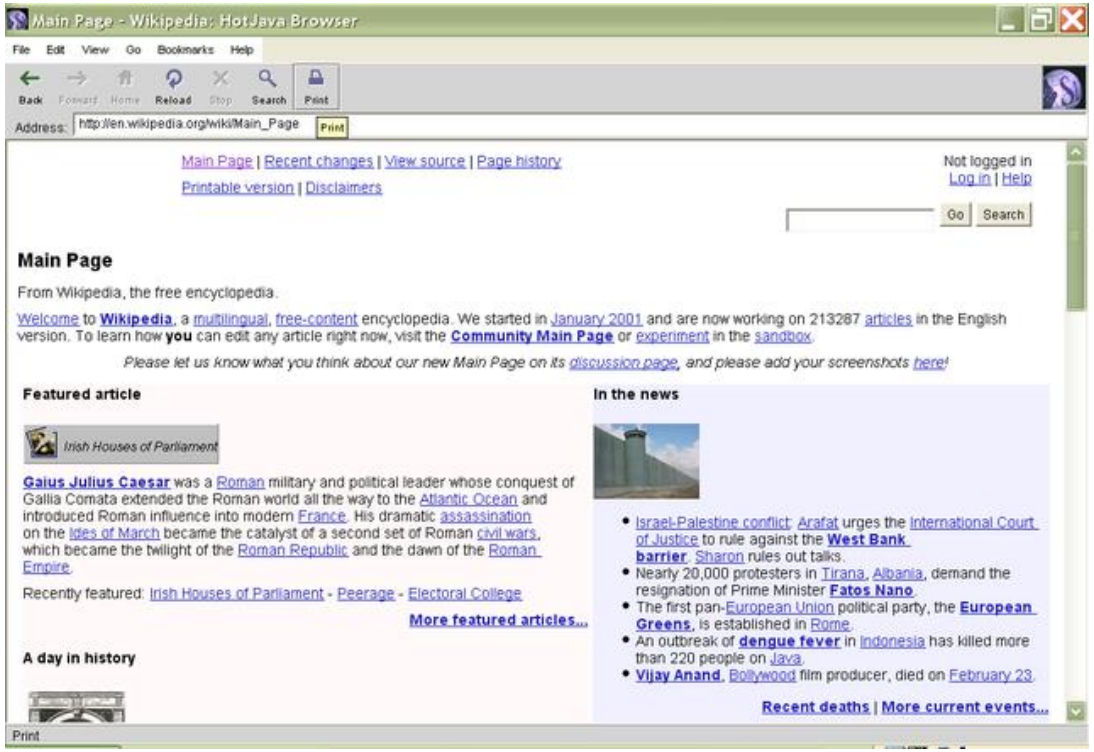
https://www.oracle.com/java/moved-by-java/timeline/

Java takes off

By sheer good fortune, the World Wide Web www exploded in popularity in 1993 and Sun saw the immediate potential of using Java to create Web pages (later Java Applets).

The team created ‘WebRunner’, named as homage to the movie Blade Runner. The browser was created using the Oak programming language and ran Oak applications. It later became ‘HotJava’.

Sun formally announced Java at a conference in May 1993, and Java (JDK) 1.0 was released in 1996.

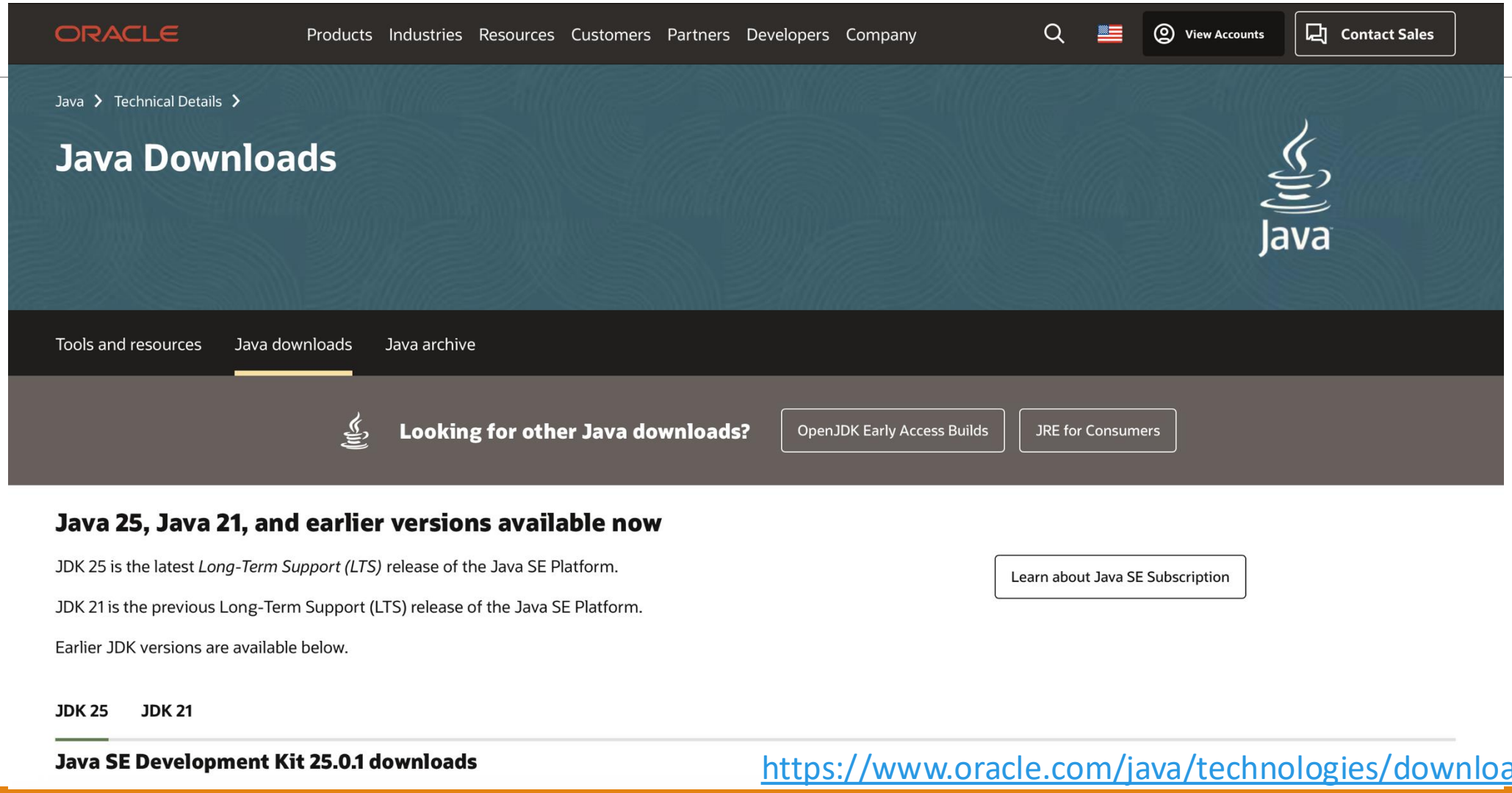


https://www.oracle.com/java/moved-by-java/timeline/

Why Java?

- Java is one of the most popular programming languages, alongside C++, C#, and Python, and web programming languages (HTML, CSS and JavaScript).
- According to Oracle, **Three billion** devices run Java.
- Furthermore, Android apps are also developed using Java.

Download Java from Oracle:



https://www.oracle.com/java/technologies/downloads/

Why Java?

There are five main design goals that informed the creation of Java (Oracle 1999):

1. It must be simple, object-oriented, and familiar
2. It must be robust and secure (no pointers)
3. It must be architecture-neutral and portable (JVM)
4. It must execute with high performance (automatic memory management)
5. It must be interpreted, threaded and dynamic

Java Virtual Machine (JVM)

The Java Virtual machine

A **Java virtual machine** (**JVM**) is an abstract computing machine.

There are three notions of the JVM:

- specification,
- implementation
- instance

An instance of the JVM can execute any executable computer program compiled into Java bytecode. It is the code execution component of the Java platform.

Java Bytecode

Java bytecode is the instruction set of the Java virtual machine.

Each bytecode is composed by one, or in some cases two, bytes that represent the instruction (opcode), along with zero or more bytes for passing parameters.

- Essentially it is created when the high level language is converted into binary.

The Java Virtual Machine

The process of converting your source code into machine code is a **two stage process** in Java.

First the program is compiled into what is called Java Byte Code (essentially binary)

Providing the machine that you are working on has a Java Virtual Machine (JVM) the program can then be interpreted/linked and run.

This is what makes the Java platform independent. A Java program can be run on any platform, Unix, Windows, Mac etc

Compilers and Interpreters

Compilers

- operate on the entire program;
- provide a permanent binary file which may be executed (or run).

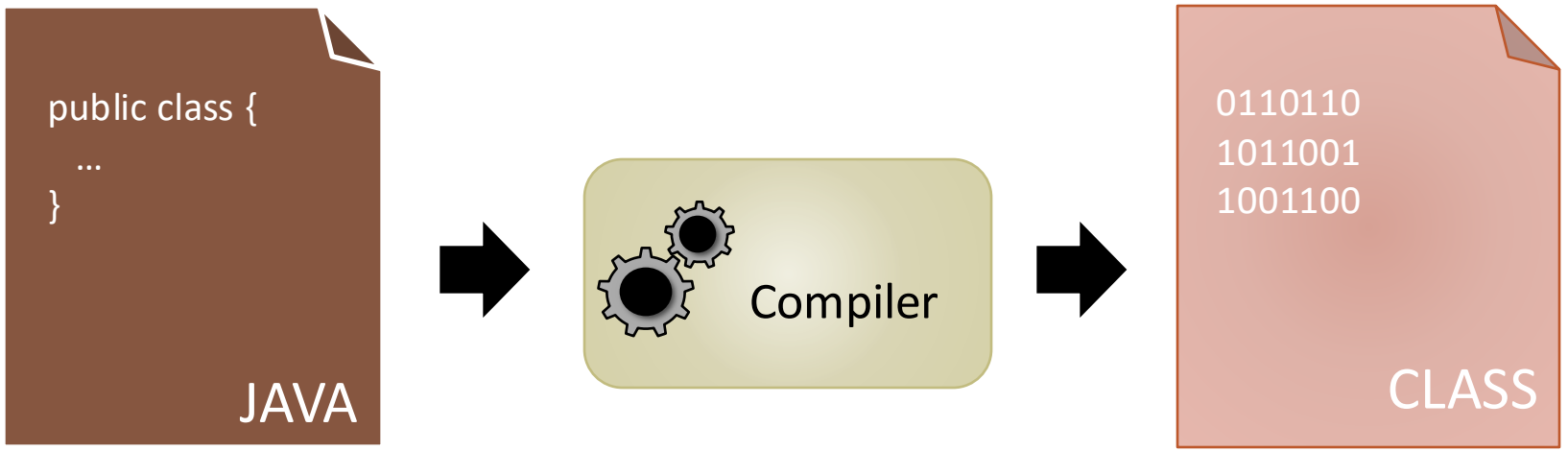
Interpreters:

- execute programs directly without producing an executable
- data types are inferred from values provided

With Java, the processes of **compilation and interpretation are combined**.

Compilation process (javac)

Compilation



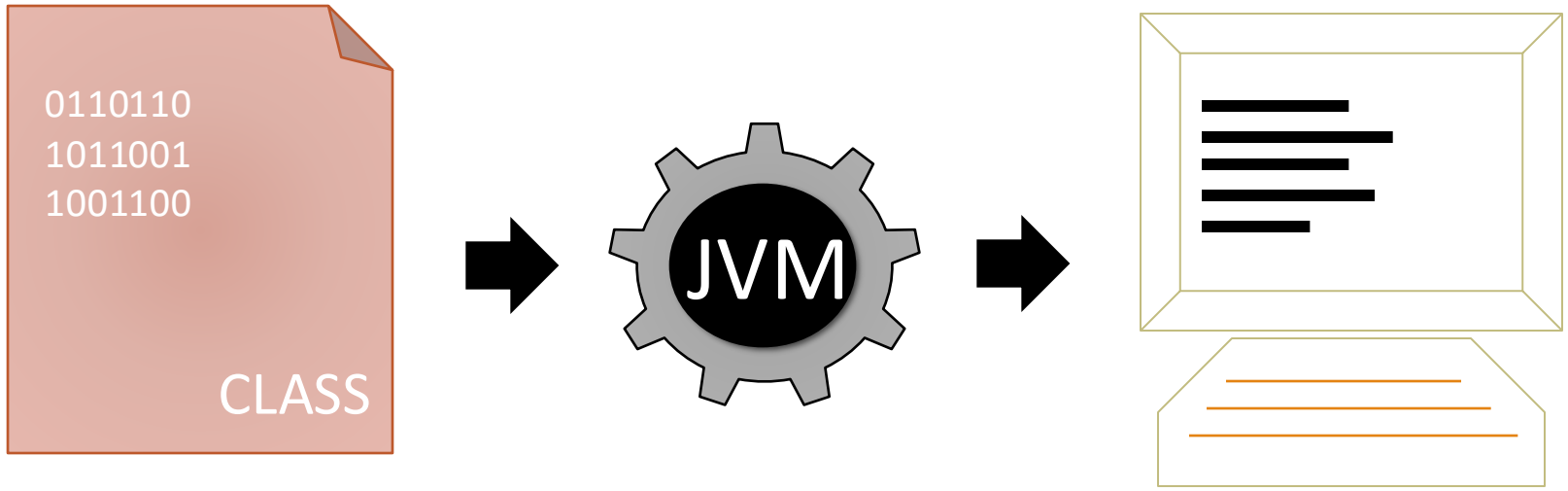
*.java
source code

javac

*.class
byte code

Interpretation process (JVM & JRE)

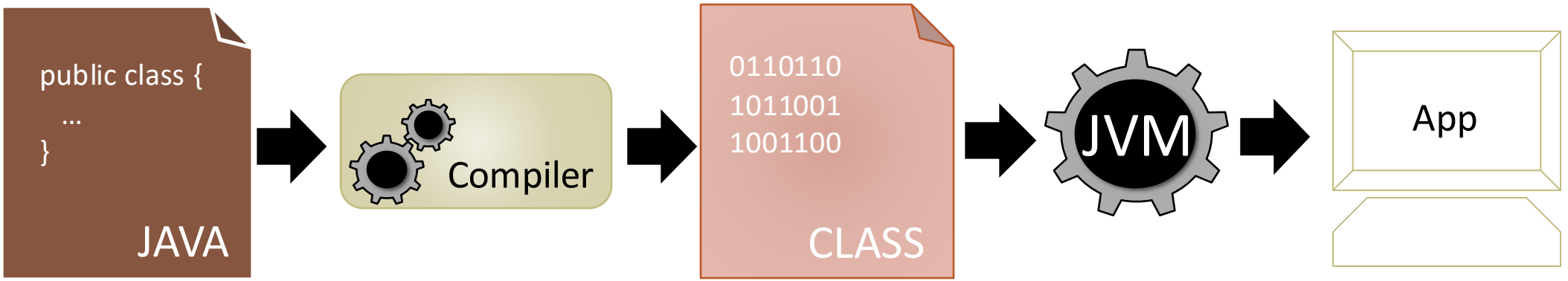
Interpretation



*.class
byte code

JRE installed on
Mac, Linux, Windows…

Compilation and interpretation



Compilation

Interpretation

Java Acronyms

- JDK: Java Development Toolkit (for compiling applications)
- JRE: Java Runtime Environment (for running applications)
- JVM: Java Virtual Machine (for interpreting java code)
- Java API: Application Programming Interface
- JIT: Just In Time compilation
- IDE: Integrated Development Environment

The Java Language

Exploration of the differences

In the following section we shall explore some of the differences between the Java and Python language.

It is not intended to be an exhaustive coverage of all Java language features, but highlight some of the key differences in approach.

The intention is to also reinforce the core programming concepts. Same concepts, different syntax/expression across language.

Output

Output (Hello World) in Java vs Python


```java
public class Program{
    public static void main(String[] args){
        System.out.println("Hello World");
    }
}
```

```python
def main():
    print("Hello, World!")

if __name__ == "__main__" :
    main()
```

Input

String Input in Java vs Python


```java
import java.util.*;  //to get access to Scanner
public class Input{
    public static void main(String[] args){
        System.out.println("Enter a string: ");
        Scanner reader = new Scanner(System.in);
        String input = reader.nextLine();
        reader.close(); //release resources
        System.out.println("You entered "+ input);
    }
}
```


```python
input = input("Enter a string: ")
print("You entered:", input)
```

Numerical Input in Java vs Python


```java
import java.util.*;  //to get access to Scanner
public class Input{
    public static void main(String[] args){
        System.out.println("Enter an integer: ");
        Scanner reader = new Scanner(System.in);
        int number = reader.nextInt();
        reader.close(); //release resources
        System.out.println("You entered "+ number);
    }
}
```


```python
number = int(input("Enter an integer: "))
print("You entered:", number)
```

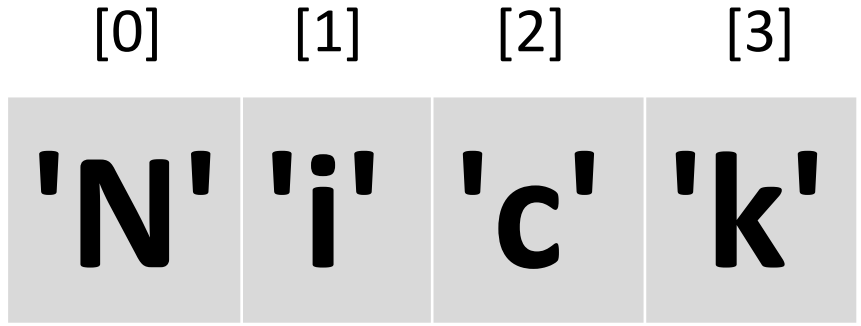
Strings

A String object is an array

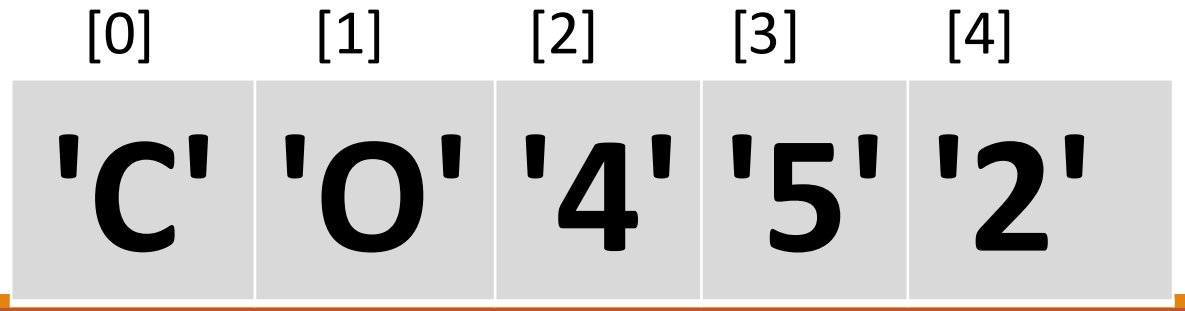
A String object is an **immutable array of characters**.

Each character has a numbered position in the array (index):

String name = "Nick";



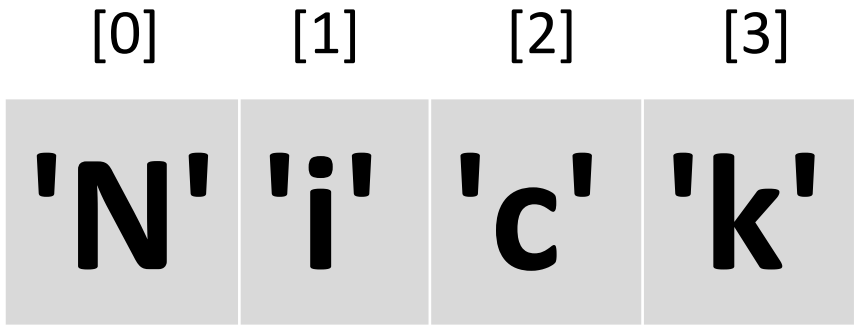
String code = "CO452";



Referring to characters in a String

You can refer to letters of a String through the index value.
In Java you can pass the index value as a parameter to the method **charAt()**

String name = "Nick";



System.out.println(name.**charAt**(0)); // displays 'N'

Java’s equals() method

Whilst the equality operator (==) can be applied to primitive data (int, char, boolean), Strings are classes, so **the equality operator would compare memory addresses of String objects** rather than the values stored in each object

Use the method **equals** to compare the values stored at String variables rather than comparing memory addresses

```
if(name.equals("Nick"))
```

Comments

JavaDoc comments

```
/**
 * This is a Javadoc comment for class Hello
 */
public class Hello
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
```

Ja

Constructor Summary

Constructors

Constructor and Description
`ArrayList()` Constructs an empty list with an initial capacity of ten.
`ArrayList(Collection<? extends E> c)` Constructs a list containing the elements of the specified collection, in the order they are returned by the collection's iterator.
`ArrayList(int initialCapacity)` Constructs an empty list with the specified initial capacity.

Method Summary

Methods

Modifier and Type	Method and Description
`boolean`	`add(E e)` Appends the specified element to the end of this list.
`void`	`add(int index, E element)` Inserts the specified element at the specified position in this list.
`boolean`	`addAll(Collection<? extends E> c)` Appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's Iterator.
`boolean`	`addAll(int index, Collection<? extends E> c)` Inserts all of the elements in the specified collection into this list, starting at the specified position.
`void`	`clear()` Removes all of the elements from this list.
`Object`	`clone()` Returns a shallow copy of this `ArrayList` instance.
`boolean`	`contains(Object o)`

Variables and Types

Data types

Java type	Description	Range of values
byte	Very small integers	-128 to 127
short	Small integers	-32,768 to 32,767
int	Big integers	-2,147,483,648 to 2,147,483,647
long	Very big integers	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	Real numbers	$+/- 1.4 * 10^{-45}$ to $3.4 * 10^{38}$
double	Very big real numbers	$+/- 4.9 * 10^{-324}$ to $1.8 * 10^{308}$
char	Characters	Unicode character set
boolean	Either true or false	true, false

Naming Variables in Java

You can choose any name for variables as long as

- the name is not already a word in the Java language (such as **class**, **void**);
- the name has no spaces in it;
- the name does not include operators such as + and -;
- the name starts either with a letter, an underscore (_), or a dollar sign ($).

The convention in Java programs is to begin the name of a variable with a *lowercase* letter.

Variables in Java vs Python


```java
public class Variables{
    public static void main(String[] args){
        char letter = 'A';
        String name = "Nick";
        int id = 9876;
        double pi = 3.14;
    }
}
```


```python
letter = 'A'
name = "Nick"
id = 9876
pi = 3.14
```

Constants

Constants in Java vs Python


```java
public class Constants
{
    public static void main(String[] args)
    {
        final int MAX_MARK = 100;
        final double PI = 3.14;
    }
}
```


```python
MAX_MARK = 100
PI = 3.14
```

Selection

If statements in Java vs Python


```java
...
    int number = 5;
    if (number > 0){
        System.out.println("The number is positive.");
    }
    else if(number < 0){
        System.out.println("The number is negative.");
    }
    else{
        System.out.println("The number is 0.");
    }
...
```


```python
number = 5
if number > 0:
    print("The number is positive.")
elif number < 0:
    print("The number is negative.")
else:
    print("The number is 0.")
```

switch statement

The switch statement selects between cases

```
char grade = 'C';

switch(grade) {
    case 'A' : System.out.println("You achieved an A grade"); break;
    case 'B' : System.out.println("You achieved a B grade"); break;
    case 'C' : System.out.println("You achieved a C grade"); break;
    case 'D' : System.out.println("You achieved a D grade"); break;
    case 'F' : System.out.println("You achieved a F grade"); break;
    default : System.out.println("Invalid grade");
}
```

switch statements in Java vs Python


```java
...
   char grade = 'C';
   switch(grade) {
      case 'A' : System.out.println("You achieved an A grade"); break;
      case 'B' : System.out.println("You achieved a B grade"); break;
      case 'C' : System.out.println("You achieved a C grade"); break;
      default : System.out.println("Invalid grade");
    }
...
```

```python
grade = 'C'
match grade:
   case 'A' : print("You achieved an A grade")
   case 'B' : print("You achieved an B grade")
   case 'C' : print("You achieved an C grade")
   case _ : print("Invalid grade")
```

The '?' Operator: an example

```
int a = 18;
int b = 7;
int highest;
                          true     false
highest = (a > b)   ?    a   :     b;     //assign highest

System.out.println("The highest number is " + highest);
```

If the comparison evaluates to **true**, the **?** operator returns the value (**a**) to the **left of the : (colon)**

If the comparison evaluates to **false**, the **?** operator returns the value (**b**) to the **right of the : (colon)**

Iteration

For each in Java vs Python


```
...
   public class Program{
       public static void main(String[] args){
           int[] numbers = {1,2,3,4,5};
           for (Integer i : numbers){
               System.out.println(i);
           }
       }
    }
```

```
for i in range(1,6):
    print(i)
```

Java do while loop

The do while loop repeats whilst true

```
char response;
do
 {
     // program instructions go here
     System.out.print("another go (y/n)?");
     response = reader.nextChar();  // Java input
 }
 while (response == 'y');
```

Functions

Functions in Java vs Python


```
...
   public class Program{
       public static void main(String[] args){
           sayHello();
       }
       public static void sayHello(){
           System.out.println("Hello World");
       }
   }
```


```
def say_hello():
    print("Hello, World!")

say_hello()
```

Arrays

For loop in Java vs Python


```
…
   public class Program{
       public static void main(String[] args){
           int[] numbers = {1,2,3,4,5};
           for (int i = 0; i < 5; i++){
               System.out.println(numbers[i]);
           }
       }
    }
```

```
numbers = [1,2,3,4,5]

print(numbers)
```

Java’s Collections Framework

Java Collections

Java created classes around key data structures:

- List
- LinkedList
- ArrayList
- Set
- HashSet
- Map

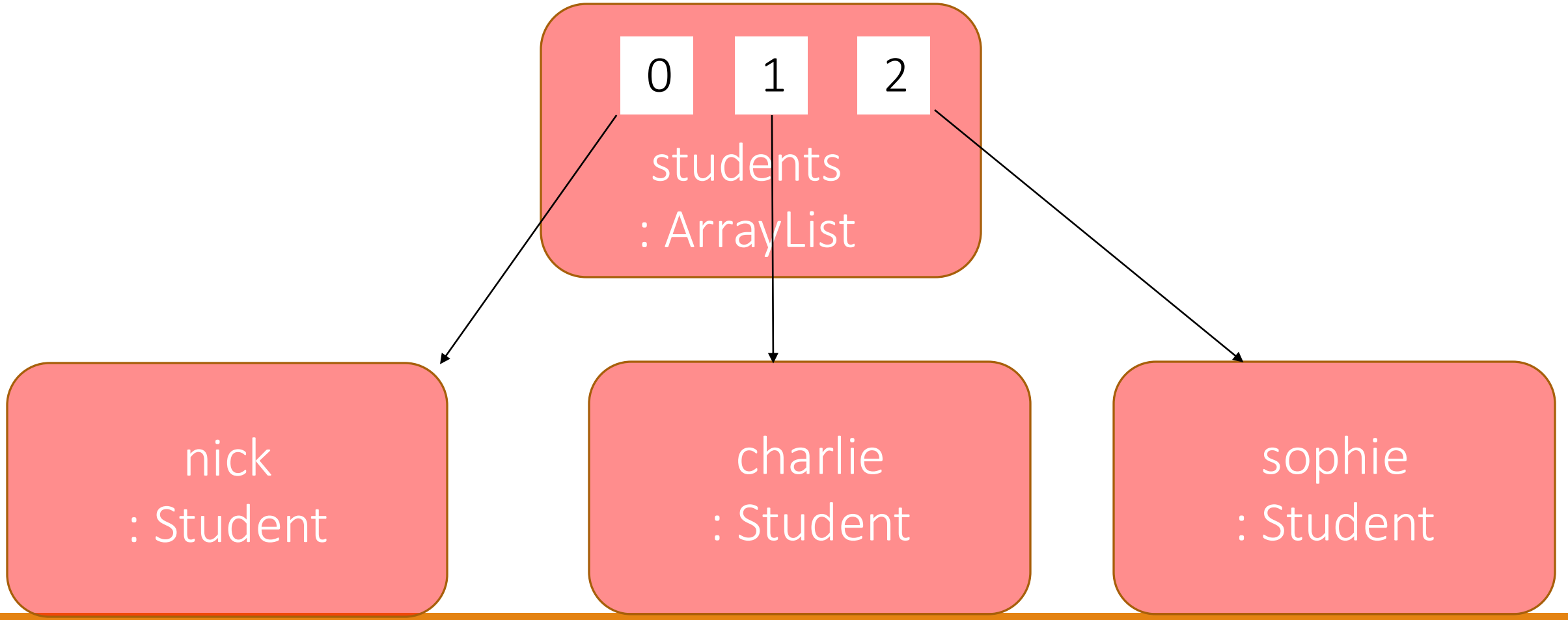
ArrayList

An instantiation of an ArrayList allows objects to be added (appended) to the end of the list. An ArrayList has no fixed-size so more memory can be allocated if more objects are added to the list.

Objects added do have an index position.

Can find objects by iterating through the list, or if the index is known, can be accessed directly through the index position.

Visualisation of an ArrayList



Syntax for instantiating an ArrayList

The syntax for creating objects of Collection classes is similar to creating objects of classes. **However, the difference is that Collections are Generic and require the type of objects to be stated in the angle brackets <>**

```
private ArrayList<Student> students = new ArrayList<>();
```

Scope Collection < type of objects > identifier = new constructor<>();

Reminder of syntax for creating objects

Similar to creating arrays (last week), we have to declare objects (variables) of a class type (data type). Then the **`new`** keyword instructs the compiler to allocate the appropriate amount of memory for an object (variable) of this type. Then call the constructor (method with the same name as the class).

Student nick = new Student();

classname objectname = new classname();

(aka. 'constructor')

Syntax for instantiating an ArrayList

Remember to import the ArrayList Class from the java.util package above the class definition.

```
import java.util.ArrayList;
/**
* Class comment…
*/
public class StudentTester
{
    private ArrayList<Student> students = new ArrayList<>();
}
```

Some methods of the ArrayList

add()

remove()

clear()

get()

size()

Adding objects through the method

Conveniently, we can invoke the ‘add’ method through an ArrayList object to append the items to the list:

students.**add**(nick);

students.**add**(charlie);

students.**add**(sophie);

For each loop with collection

The for each loop can be used to iterate through collections of objects.

Requires an object to be declared of the type of item that is in the collection:

```
for(Student student : students)
{
   student.print();
}
```

Finding an item in an ArrayList

… and can check that sought after value matches an item of the ArrayList:

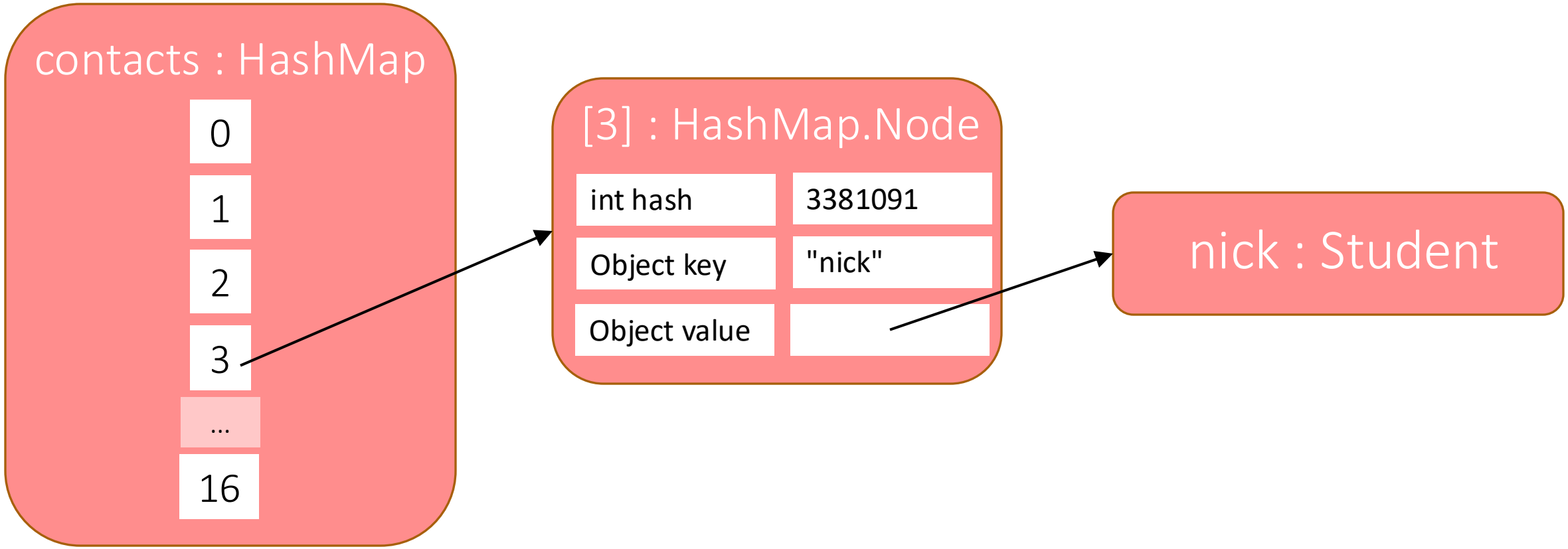
```
public Student findByID(int id)
{
    for(Student student : students)
    {
       if(student.getID() == id)
          return student;
    }
    return null;
}
```

Hash Map

Items are added to an ArrayList in the order that the ‘add’ method is called (added to the end of the list). LinkedLists also append or prepend items.

HashMaps, however, will ‘put’ items (values) in a position that corresponds with their ‘key’. Items (values) can subsequently be retrieved by their key (e.g a String literal). A String can be converted to an integer position by the process of Hashing. Each character has an integer ASCII value.

Visualisation of a HashMap



Classes

Classes in Java vs Python


```java
class Student{
    private int id;
    private String name;

    public Student(int id, String name){
        this.id = id;
        this.name = name;
    }
}
```


```python
class Student:
    def __init__(self, id, name):
        self.id = id
        self.name = name
```

Objects

Objects in Java vs Python


```
…
 public class Program{
     public static void main(String[] args){
         Student nick = new Student(1234, "Nick");
         nick.print();
     }
 }
```

```
from student import Student


nick = Student(1234, "Nick")
nick.print()
```

Inheritance

Inheritance in C# vs Python


```java
class Child extends Parent {
    private string name;

    public Child(int id, String name){
        super(id); //call parent constructor
        this.name = name;
    }
}
```

```python
class Child(Parent):
    def __init__(self, id, name):
        super().__init__(id)
        self.name = name
```

Summary

Summary

Java went public in 1995 (JDK 1.0 released January 1996)

Java Virtual Machine (JVM) was a transition from the ‘compilation’ to hardware specs (C/C++) and towards interpretation by VM (Python/Java/C#) now as there is more need for cross platform technology.

Java known for its portability (JVM), automatic ‘garbage collection’, Collections (generics), and for underpinning apps made in Android Studio.

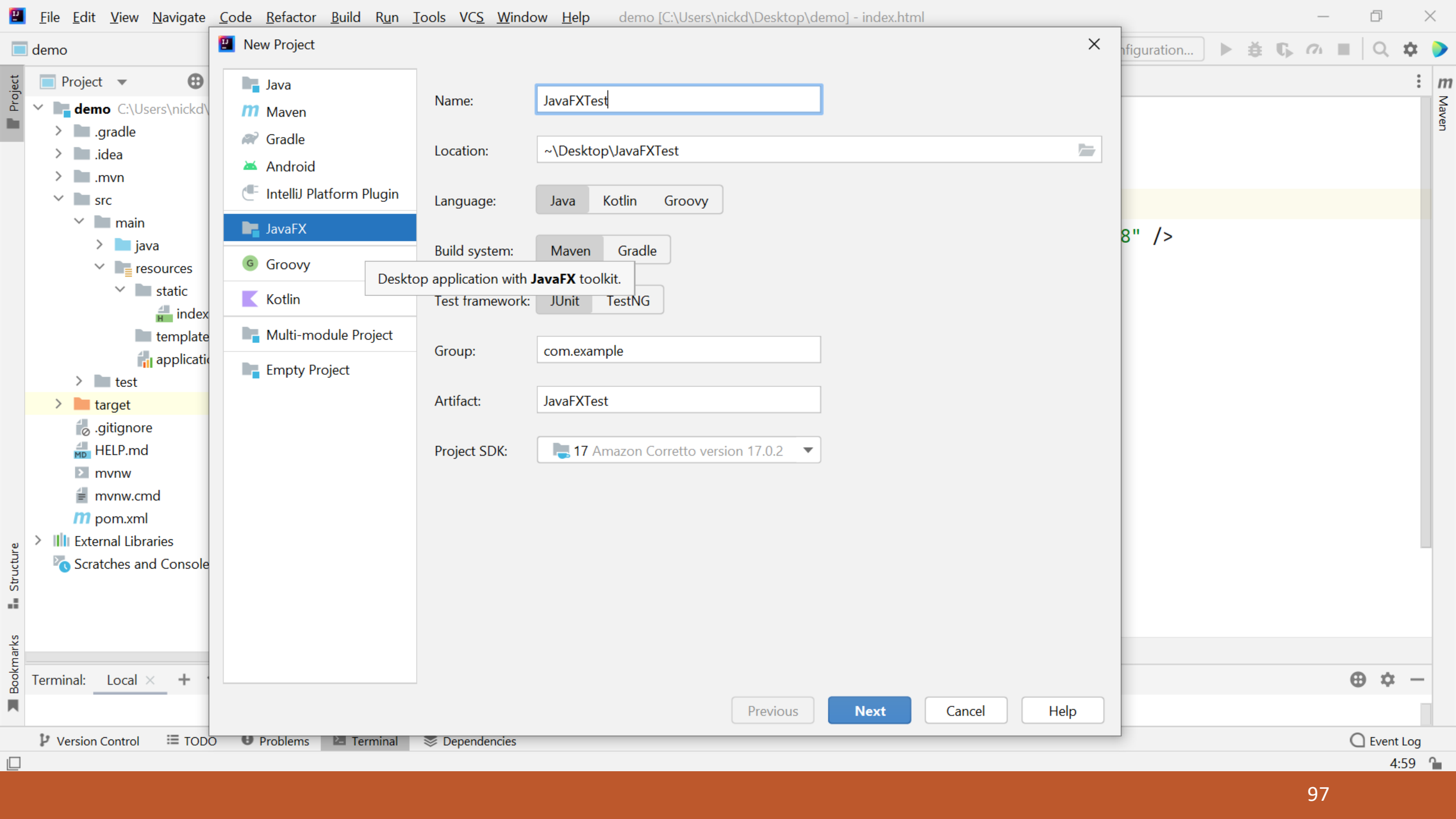
Java Graphics: JavaFX

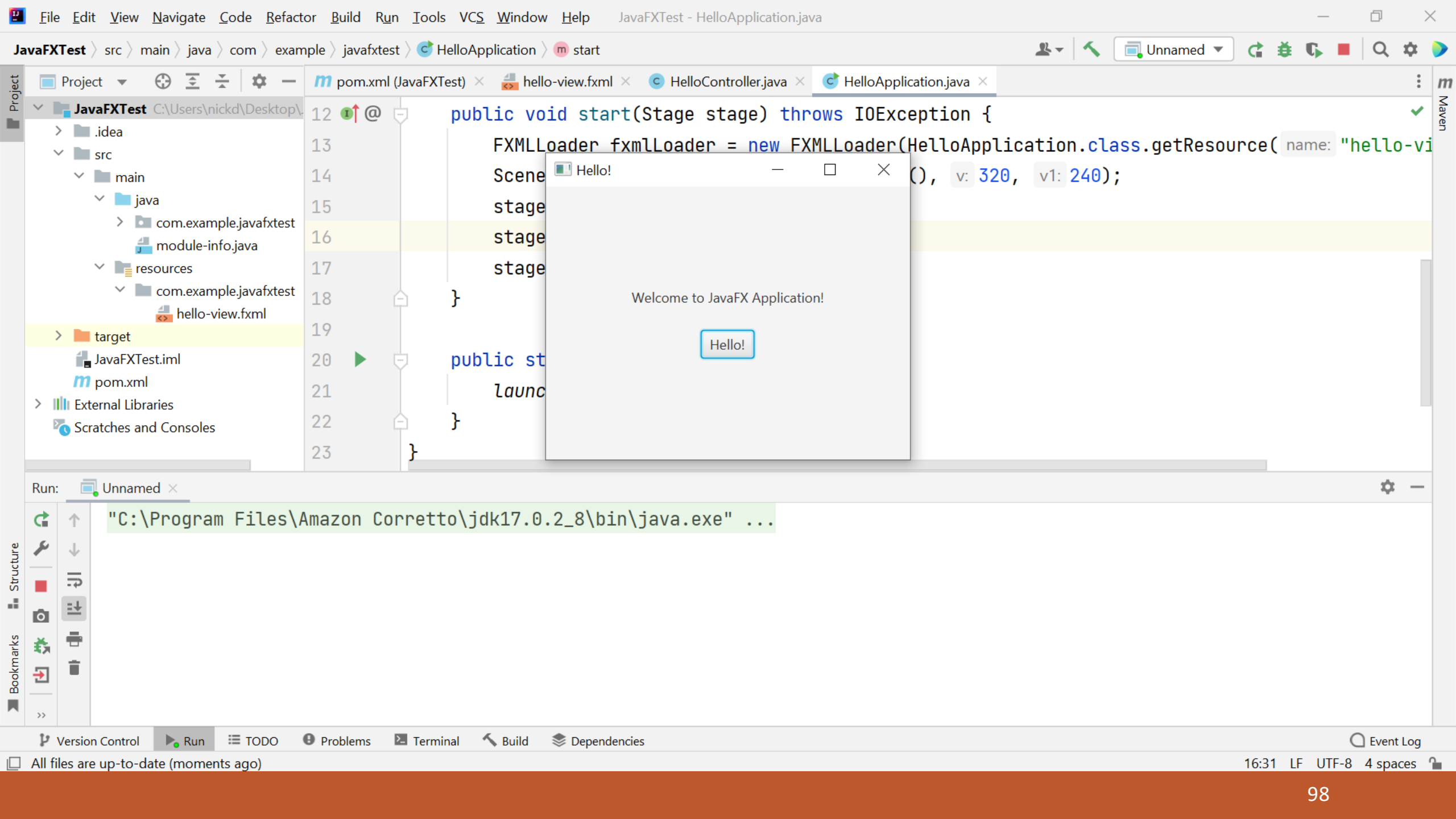
Java Graphics – AWT to Swing to JavaFX

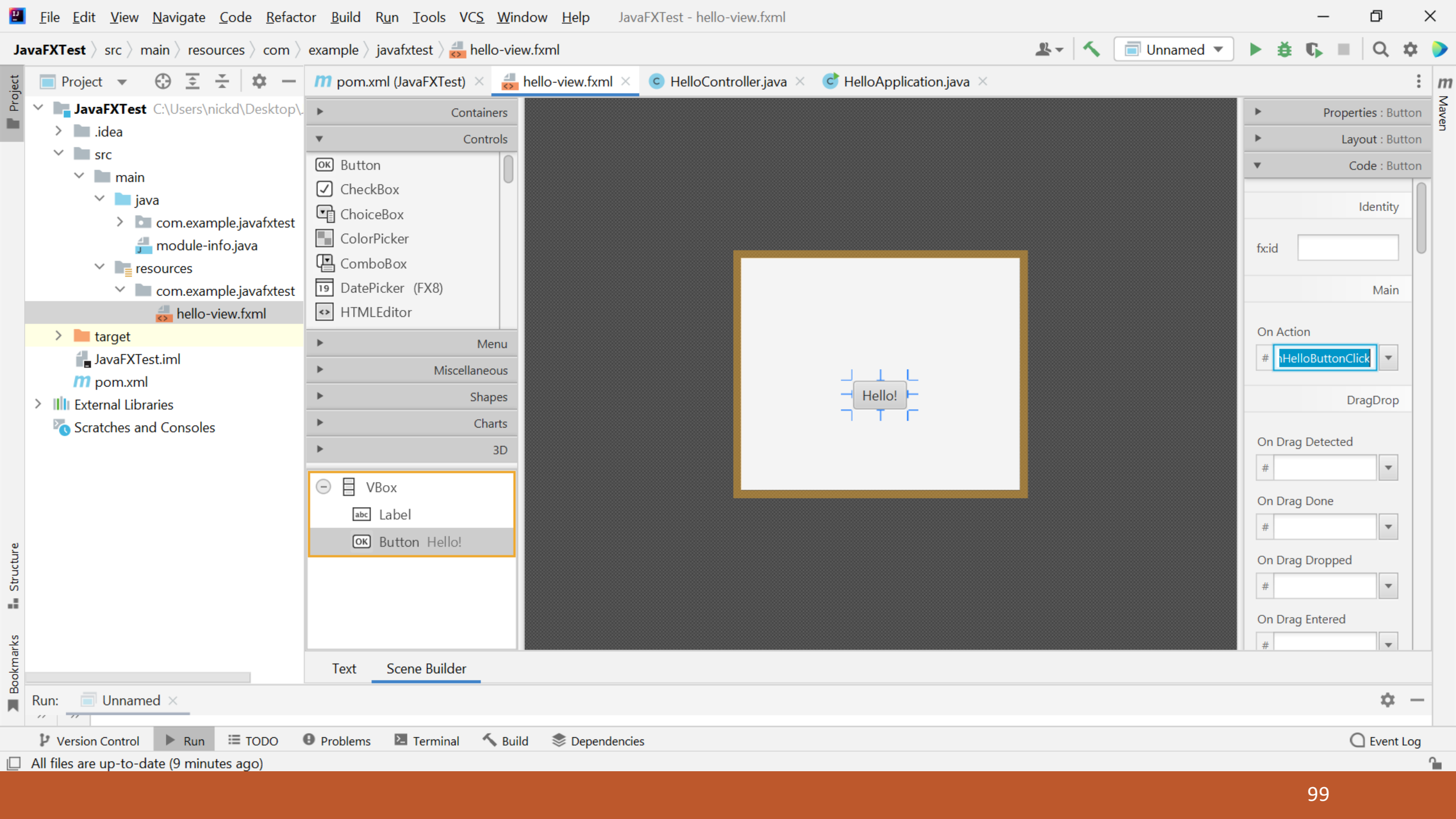
In the earliest versions of Java, graphical programming could only be achieved through use of the Abstract Window Toolkit (AWT) package.

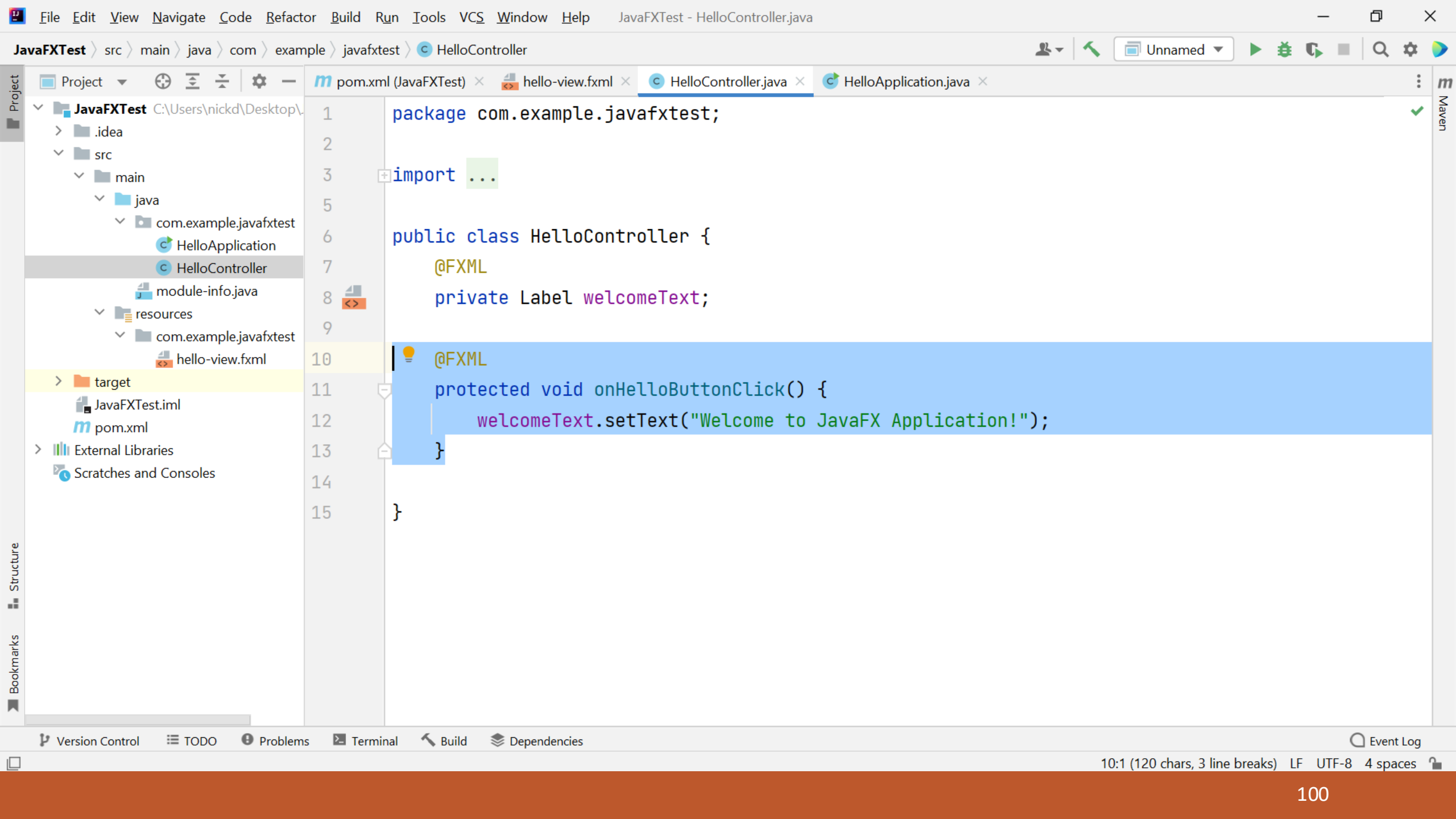
From 2000-2014, Swing was the main platform for producing Java Graphics. But has since become outdated (OS' have moved on and changed their styles).

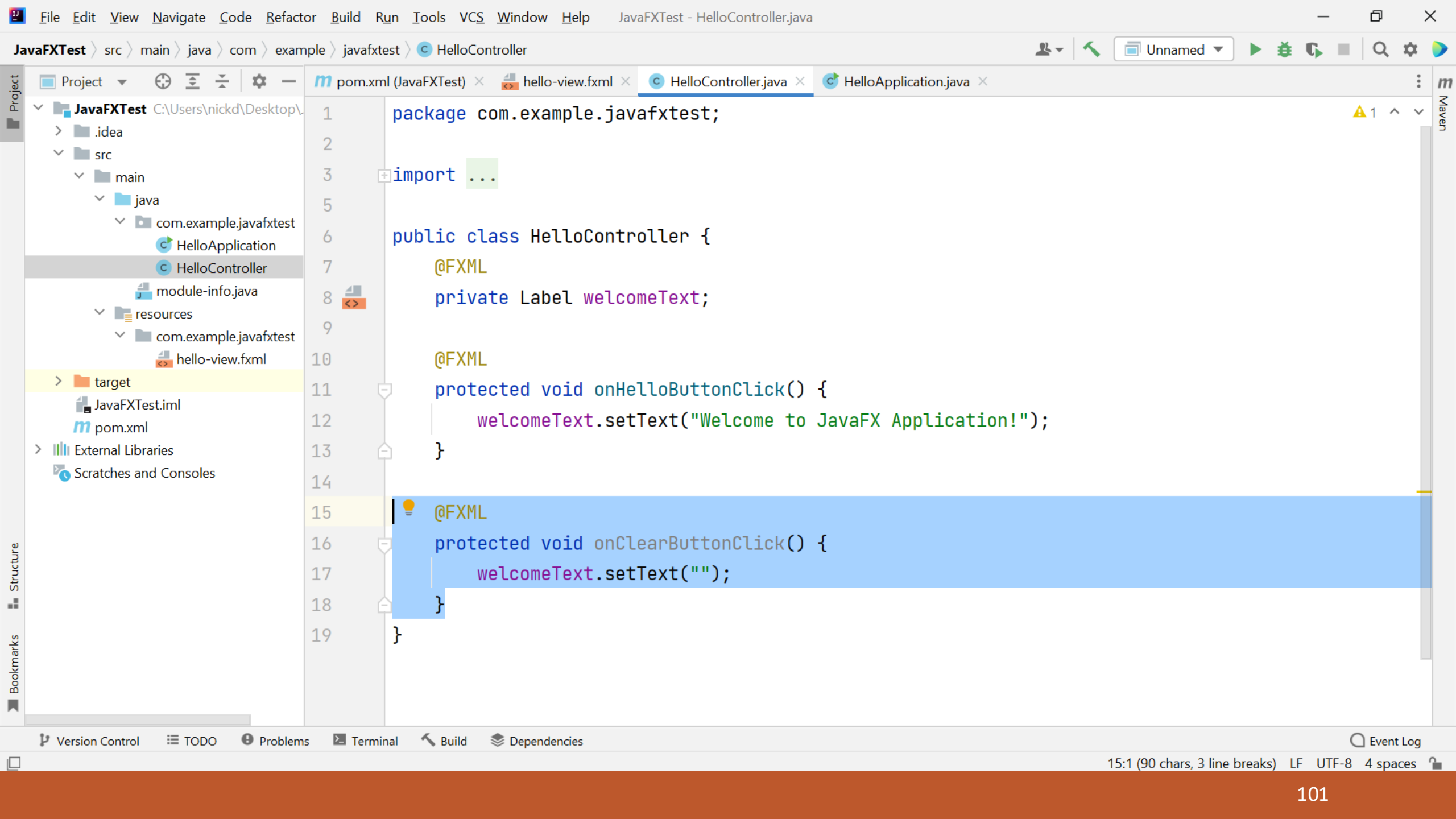
With the release of Java 8 in 2014 came JavaFX.

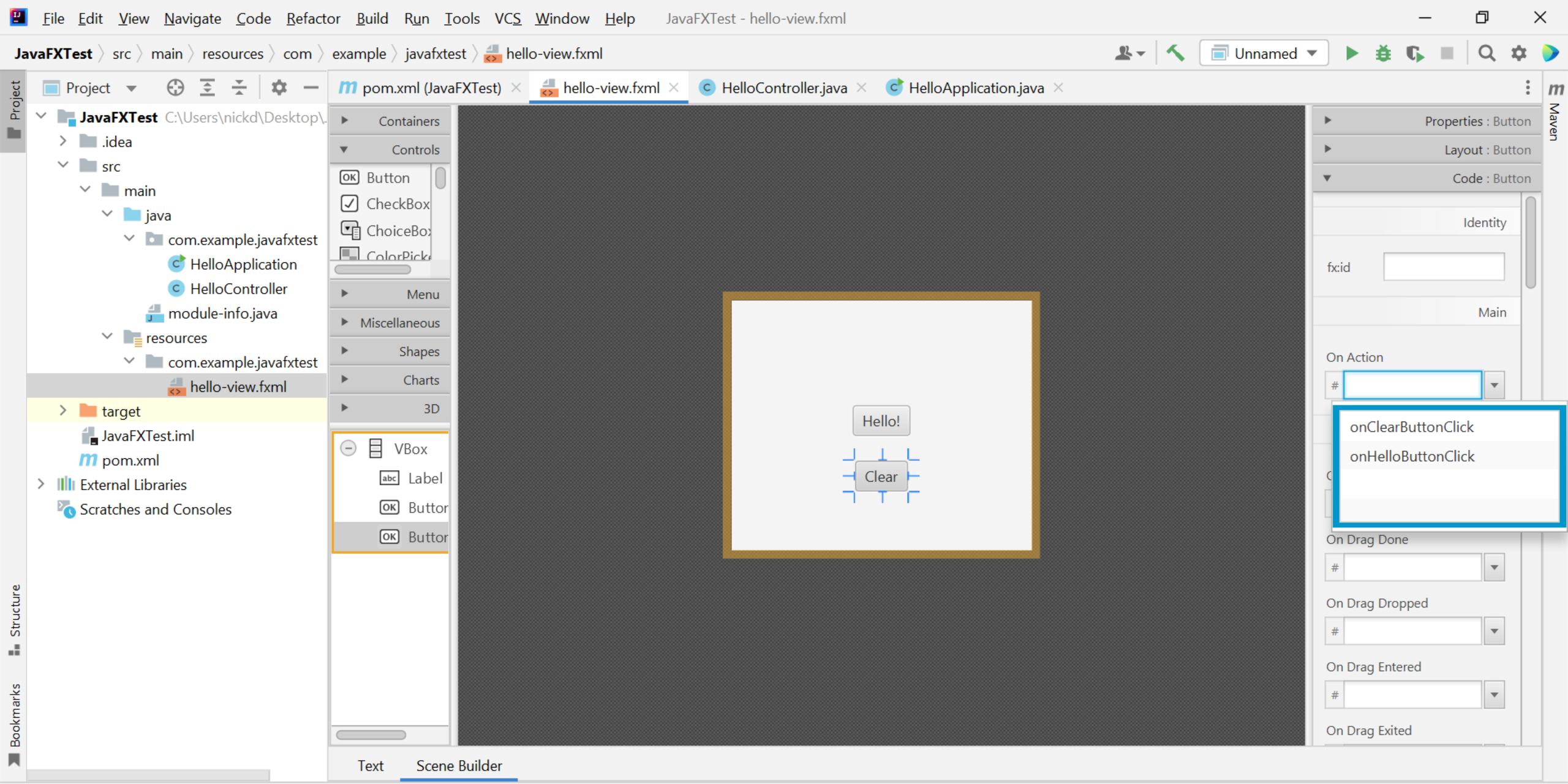

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
demo [C:\Users\nickd\Desktop\demo] - index.html
demo
Project
demo C:\Users\nickd\
.gradle
.idea
.mvn
src
main
java
resources
static
index
template
applicati
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Console
New Project
Java
Maven
Gradle
Android
IntelliJ Platform Plugin
JavaFX
Groovy
Kotlin
Multi-module Project
Empty Project
Name: JavaFXTest
Location: ~\Desktop\JavaFXTest
Language: Java Kotlin Groovy
Build system: Maven Gradle
Desktop application with JavaFX toolkit.
Test framework: JUnit TestNG
Group: com.example
Artifact: JavaFXTest
Project SDK: 17 Amazon Corretto version 17.0.2
Previous
Next
Cancel
Help
8" />
Terminal: Local
Version Control
TODO
Problems
Terminal
Dependencies
Event Log
4:59

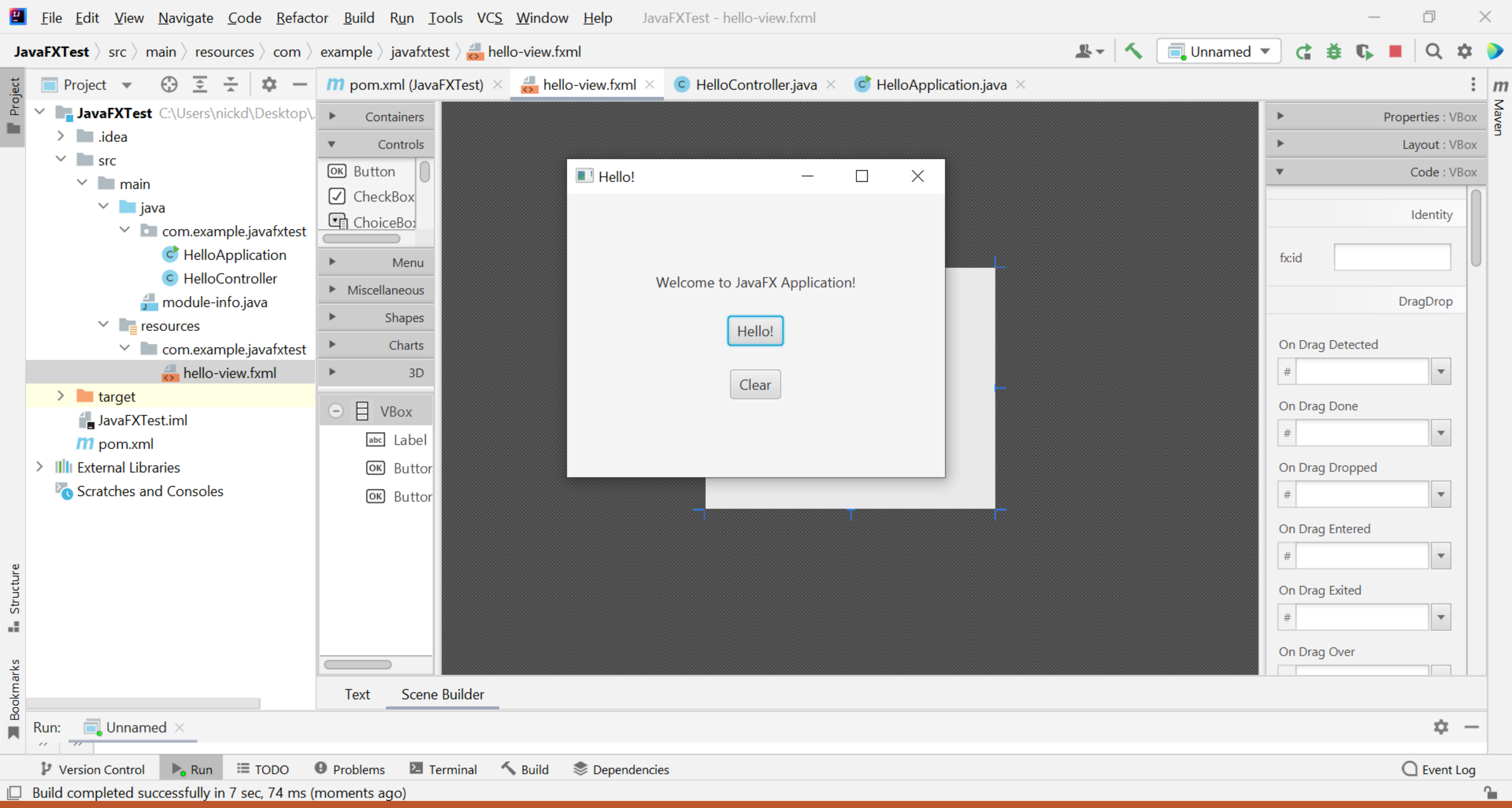

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
JavaFXTest - HelloApplication.java
JavaFXTest src main java com example javafxtest HelloApplication start
Unnamed
Project
pom.xml (JavaFXTest)
hello-view.fxml
HelloController.java
HelloApplication.java
JavaFXTest C:\Users\nickd\Desktop\
.idea
src
main
java
com.example.javafxtest
module-info.java
resources
com.example.javafxtest
hello-view.fxml
target
JavaFXTest.iml
pom.xml
External Libraries
Scratches and Consoles
12 public void start(Stage stage) throws IOException {
13 FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource(name: "hello-vi
14 Scene (), v: 320, v1: 240);
15 stage
16 stage
17 stage
18 }
19
20 public st
21 launc
22 }
23 }
Hello!
Welcome to JavaFX Application!
Hello!
Run: Unnamed
"C:\Program Files\Amazon Corretto\jdk17.0.2_8\bin\java.exe" ...
Version Control
Run
TODO
Problems
Terminal
Build
Dependencies
Event Log
All files are up-to-date (moments ago)
16:31 LF UTF-8 4 spaces

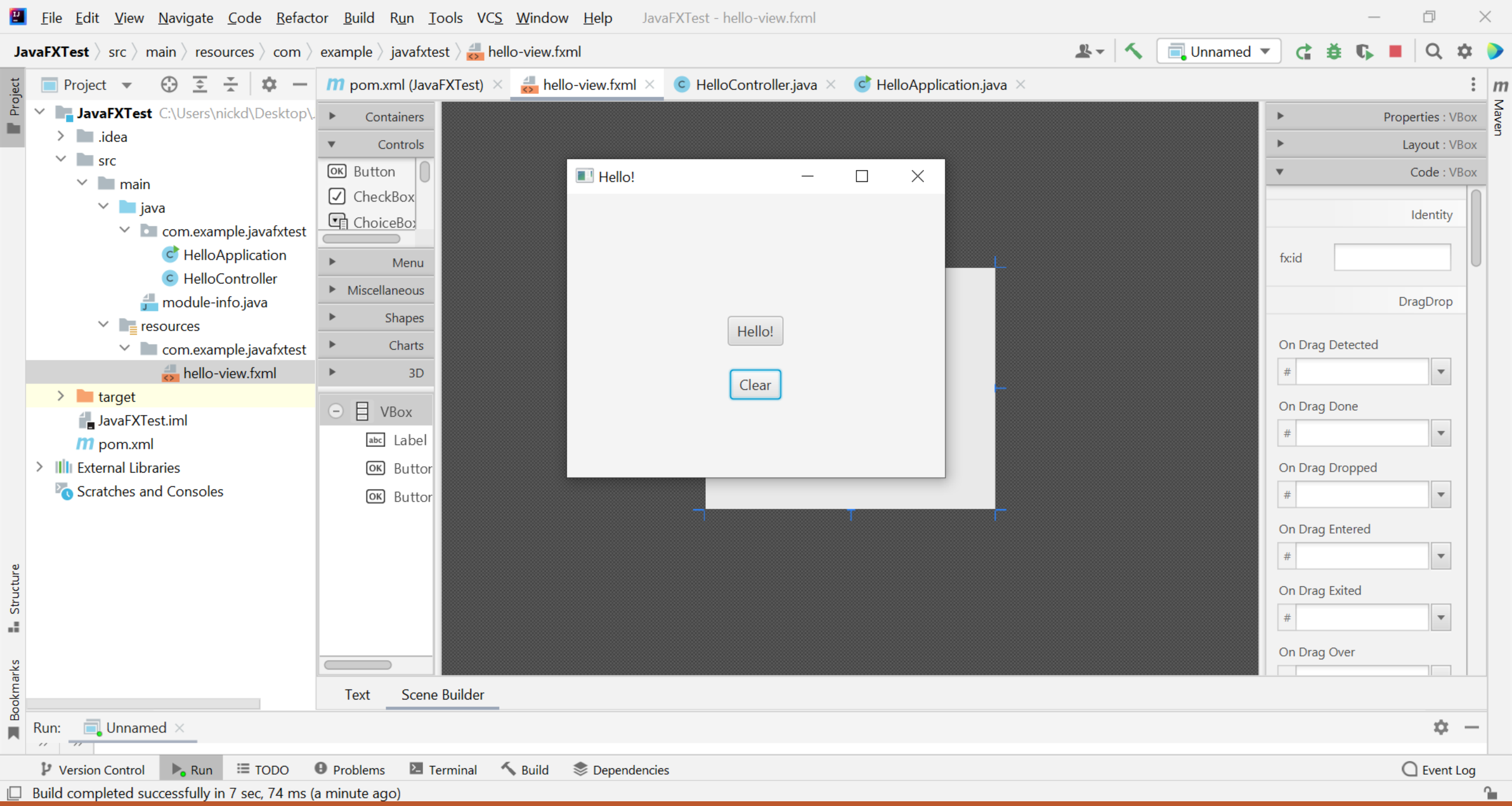

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
JavaFXTest - hello-view.fxml
JavaFXTest src main resources com example javafxtest hello-view.fxml
Unnamed
Project
JavaFXTest C:\Users\nickd\Desktop\
.idea
src
main
java
com.example.javafxtest
module-info.java
resources
com.example.javafxtest
hello-view.fxml
target
JavaFXTest.iml
pom.xml
External Libraries
Scratches and Consoles
pom.xml (JavaFXTest)
hello-view.fxml
HelloController.java
HelloApplication.java
Containers
Controls
Button
CheckBox
ChoiceBox
ColorPicker
ComboBox
DatePicker (FX8)
HTMLEditor
Menu
Miscellaneous
Shapes
Charts
3D
VBox
Label
Button Hello!
Hello!
Properties : Button
Layout : Button
Code : Button
Identity
fx:id
Main
On Action
#HelloButtonClick
DragDrop
On Drag Detected
On Drag Done
On Drag Dropped
On Drag Entered
Text
Scene Builder
Run:
Unnamed
Version Control
Run
TODO
Problems
Terminal
Build
Dependencies
Event Log
All files are up-to-date (9 minutes ago)
Project
Structure
Bookmarks
Maven


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
JavaFXTest - HelloController.java
JavaFXTest src main java com example javafxtest HelloController
Unnamed
Project
JavaFXTest C:\Users\nickd\Desktop\
.idea
src
main
java
com.example.javafxtest
HelloApplication
HelloController
module-info.java
resources
com.example.javafxtest
hello-view.fxml
target
JavaFXTest.iml
pom.xml
External Libraries
Scratches and Consoles
pom.xml (JavaFXTest)
hello-view.fxml
HelloController.java
HelloApplication.java
1 package com.example.javafxtest;
2
3 import ...
5
6 public class HelloController {
7 @FXML
8 private Label welcomeText;
9
10 @FXML
11 protected void onHelloButtonClick() {
12 welcomeText.setText("Welcome to JavaFX Application!");
13 }
14
15 }
Maven
Structure
Bookmarks
Version Control
TODO
Problems
Terminal
Build
Dependencies
Event Log
10:1 (120 chars, 3 line breaks) LF UTF-8 4 spaces

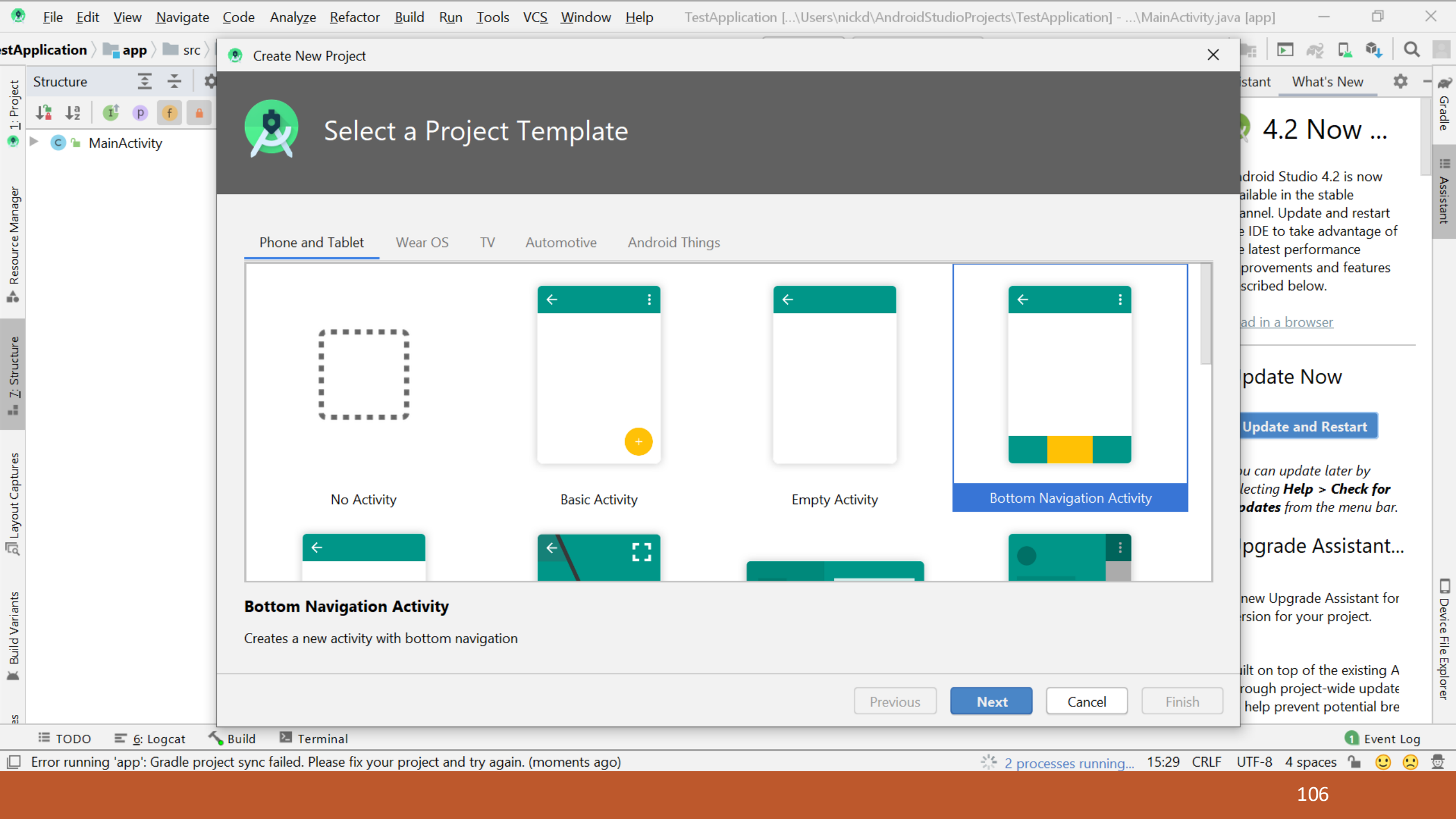

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
JavaFXTest - HelloController.java
JavaFXTest src main java com example javafxtest HelloController
Unnamed
Project
pom.xml (JavaFXTest)
hello-view.fxml
HelloController.java
HelloApplication.java
JavaFXTest C:\Users\nickd\Desktop\
.idea
src
main
java
com.example.javafxtest
HelloApplication
HelloController
module-info.java
resources
com.example.javafxtest
hello-view.fxml
target
JavaFXTest.iml
pom.xml
External Libraries
Scratches and Consoles
1 package com.example.javafxtest;
2
3 import ...
5
6 public class HelloController {
7 @FXML
8 private Label welcomeText;
9
10 @FXML
11 protected void onHelloButtonClick() {
12 welcomeText.setText("Welcome to JavaFX Application!");
13 }
14
15 @FXML
16 protected void onClearButtonClick() {
17 welcomeText.setText("");
18 }
19 }
Maven
Structure
Bookmarks
Version Control
TODO
Problems
Terminal
Build
Dependencies
Event Log
15:1 (90 chars, 3 line breaks) LF UTF-8 4 spaces


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
JavaFXTest - hello-view.fxml
JavaFXTest src main resources com example javafxtest hello-view.fxml
Unnamed
Project
JavaFXTest C:\Users\nickd\Desktop\
.idea
src
main
java
com.example.javafxtest
HelloApplication
HelloController
module-info.java
resources
com.example.javafxtest
hello-view.fxml
target
JavaFXTest.iml
pom.xml
External Libraries
Scratches and Consoles
pom.xml (JavaFXTest)
hello-view.fxml
HelloController.java
HelloApplication.java
Containers
Controls
Button
CheckBox
ChoiceBox
Menu
Miscellaneous
Shapes
Charts
3D
VBox
Label
Buttor
Buttor
Hello!
Clear
Properties : Button
Layout : Button
Code : Button
Identity
fx:id
Main
On Action
onClearButtonClick
onHelloButtonClick
On Drag Done
On Drag Dropped
On Drag Entered
On Drag Exited
Text
Scene Builder
Version Control
TODO
Problems
Terminal
Build
Dependencies
Event Log
Maven
Structure
Bookmarks


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
JavaFXTest - hello-view.fxml
JavaFXTest src main resources com example javafxtest hello-view.fxml
Unnamed
Project
JavaFXTest C:\Users\nickd\Desktop\
.idea
src
main
java
com.example.javafxtest
HelloApplication
HelloController
module-info.java
resources
com.example.javafxtest
hello-view.fxml
target
JavaFXTest.iml
pom.xml
External Libraries
Scratches and Consoles
pom.xml (JavaFXTest)
hello-view.fxml
HelloController.java
HelloApplication.java
Containers
Controls
Button
CheckBox
ChoiceBox
Menu
Miscellaneous
Shapes
Charts
3D
VBox
Label
Buttor
Buttor
Hello!
Welcome to JavaFX Application!
Hello!
Clear
Properties : VBox
Layout : VBox
Code : VBox
Identity
fx:id
DragDrop
On Drag Detected
On Drag Done
On Drag Dropped
On Drag Entered
On Drag Exited
On Drag Over
Text
Scene Builder
Run:
Unnamed
Version Control
Run
TODO
Problems
Terminal
Build
Dependencies
Event Log
Build completed successfully in 7 sec, 74 ms (moments ago)


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
JavaFXTest - hello-view.fxml
JavaFXTest src main resources com example javafxtest hello-view.fxml
Unnamed
Project
JavaFXTest C:\Users\nickd\Desktop\
.idea
src
main
java
com.example.javafxtest
HelloApplication
HelloController
module-info.java
resources
com.example.javafxtest
hello-view.fxml
target
JavaFXTest.iml
pom.xml
External Libraries
Scratches and Consoles
pom.xml (JavaFXTest)
hello-view.fxml
HelloController.java
HelloApplication.java
Containers
Controls
Button
CheckBox
ChoiceBox
Menu
Miscellaneous
Shapes
Charts
3D
VBox
Label
Buttor
Buttor
Hello!
Hello!
Clear
Properties : VBox
Layout : VBox
Code : VBox
Identity
fx:id
DragDrop
On Drag Detected
On Drag Done
On Drag Dropped
On Drag Entered
On Drag Exited
On Drag Over
Text
Scene Builder
Run:
Unnamed
Version Control
Run
TODO
Problems
Terminal
Build
Dependencies
Event Log
Build completed successfully in 7 sec, 74 ms (a minute ago)

Android Studio


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
TestApplication [...\Users\nickd\AndroidStudioProjects\TestApplication] - ...\MainActivity.java [app]
Create New Project
Select a Project Template
Phone and Tablet
Wear OS
TV
Automotive
Android Things
No Activity
Basic Activity
Empty Activity
Bottom Navigation Activity
Bottom Navigation Activity
Creates a new activity with bottom navigation
Previous
Next
Cancel
Finish
What's New
4.2 Now ...
Update and Restart
Upgrade Assistant...
Structure
MainActivity
TODO
6: Logcat
Build
Terminal
Event Log
Error running 'app': Gradle project sync failed. Please fix your project and try again. (moments ago)
2 processes running...
15:29
CRLF
UTF-8
4 spaces

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help TestApp [C:\Users\nickd\AndroidStudioProjects\TestApp] - ...\res\layout\fragment_home.xml [app]

TestApp > app > src > main > res > layout > fragment_home.xml

app | Pixel 2 Pie 9.0 - API 28

Android

- app
  - manifests
  - java
  - java (generated)
  - res
    - drawable
    - layout
      - activity_main.xml
      - fragment_dashboard.xml
      - fragment_home.xml
      - fragment_notifications.xml
    - menu
      - bottom_nav_menu.xml
    - mipmap
    - navigation
    - values
  - Gradle Scripts

mobile_navigation.xml | fragment_home.xml | activity_main.xml

Palette

Common: TextView, Button, ImageView, RecyclerView, <fragment>, ScrollView, Switch

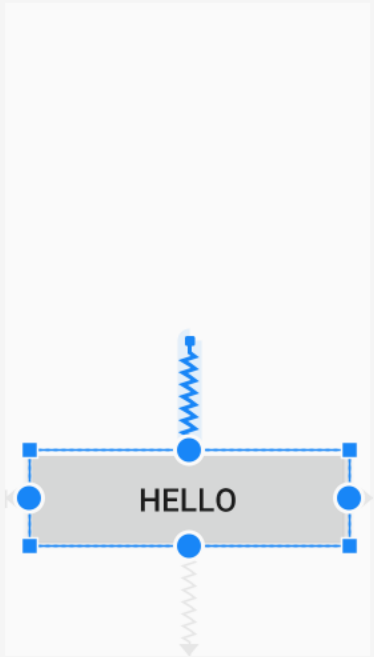
Text | Buttons | Widgets | Layouts | Containers | Google | Legacy

Component Tree

- ConstraintLayout
  - Ab text_home
  - helloButton "Hello"

Pixel | 29 | AppTheme | Default (en-us)

0dp



Attributes

helloButton — Button

End → EndOf **parent** (0dp)

Top → BottomOf **text_home** (0dp)

Bottom → BottomOf **parent** (0dp)

layout_width: 358dp

layout_height: 107dp

visibility

visibility

Common Attrib

style

onClick

background

text

text

contentDescription

textAppearance

alpha

AppCompat

Body1

Body2

Display1

Display2

Display3

Display4

Large

Medium

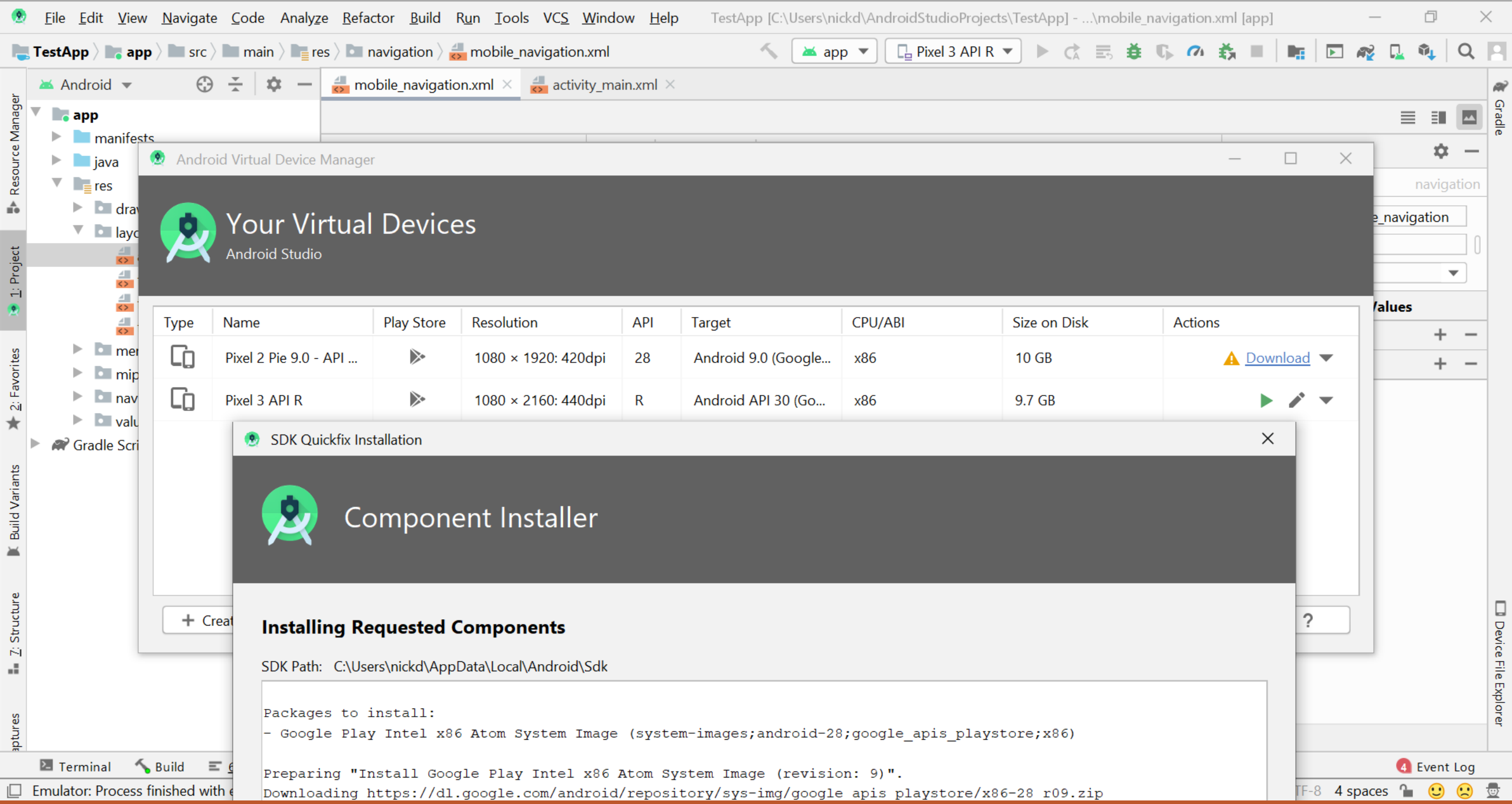
All Attributes

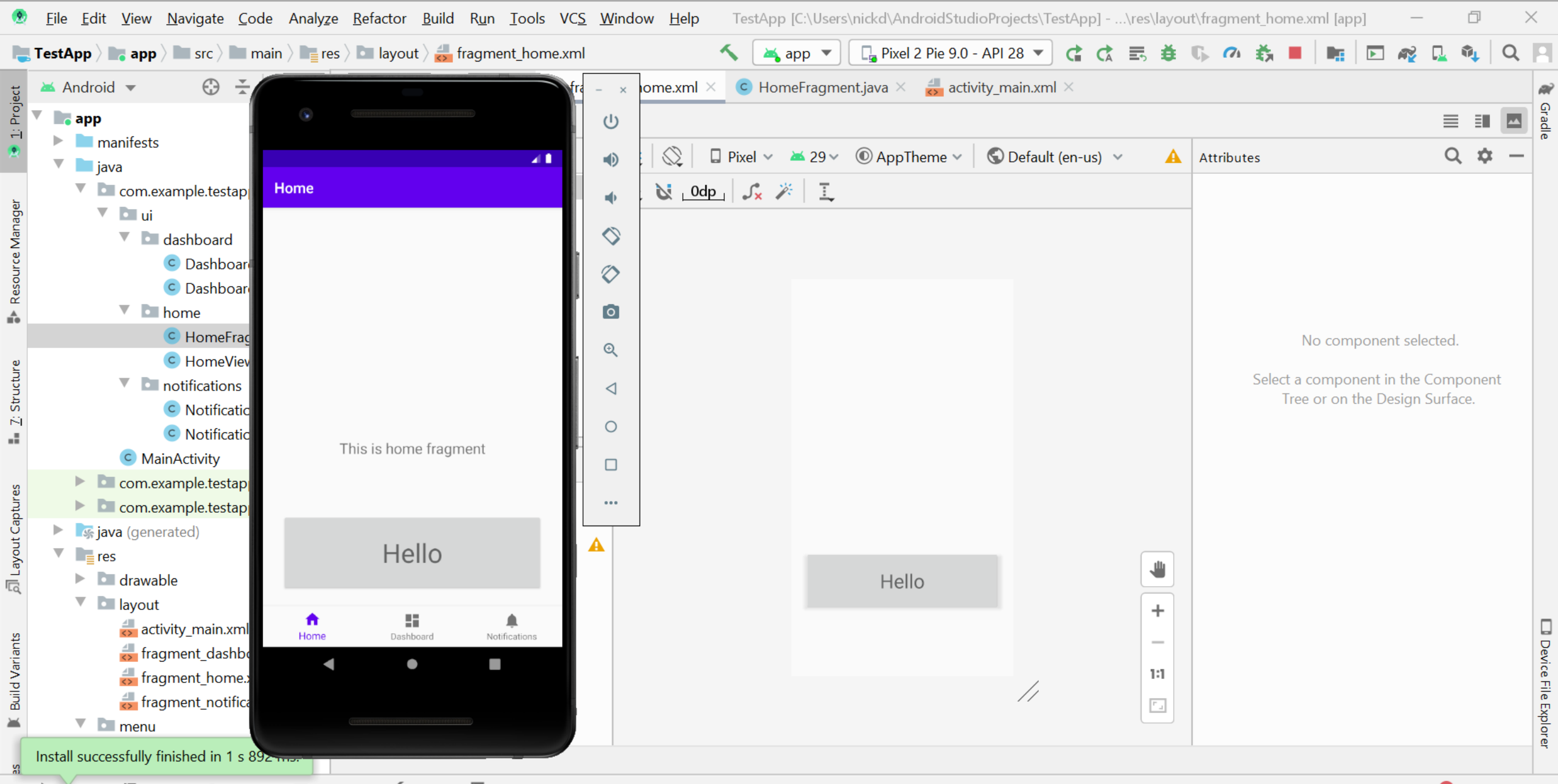
alpha

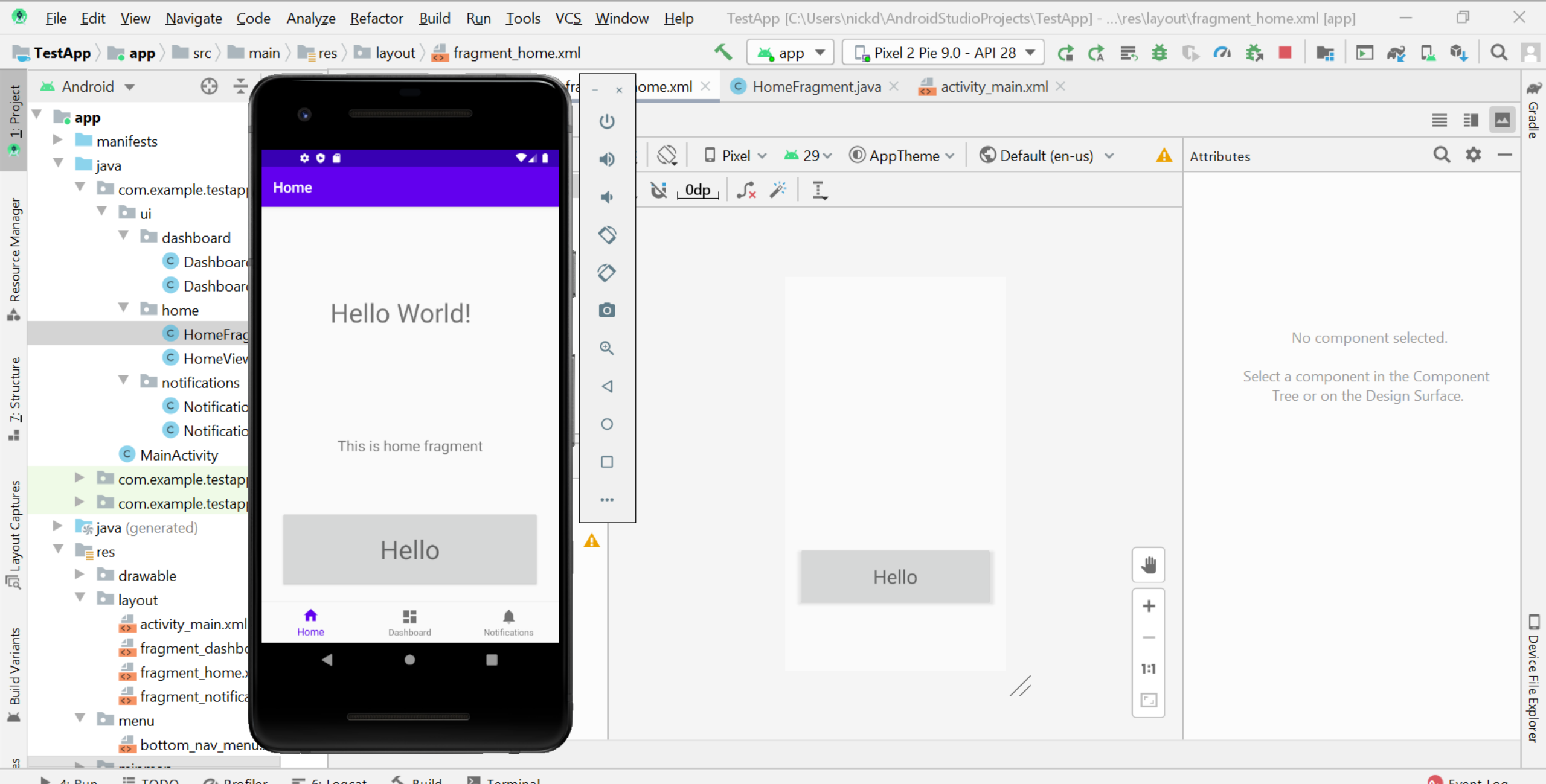
4: Run | TODO | Profiler | 6: Logcat | Build | Terminal | Event Log

Install successfully finished in 337 ms.: App restart successful without requiring a re-install. (11 minutes ago)

CRLF UTF-8 4 spaces


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
TestApp [C:\Users\nickd\AndroidStudioProjects\TestApp] - ...\mobile_navigation.xml [app]
TestApp app src main res navigation mobile_navigation.xml
app
Pixel 3 API R
Android
mobile_navigation.xml
activity_main.xml
app
manifests
java
res
Android Virtual Device Manager
Your Virtual Devices
Android Studio
Type Name Play Store Resolution API Target CPU/ABI Size on Disk Actions
Pixel 2 Pie 9.0 - API ... 1080 × 1920: 420dpi 28 Android 9.0 (Google... x86 10 GB Download
Pixel 3 API R 1080 × 2160: 440dpi R Android API 30 (Go... x86 9.7 GB
SDK Quickfix Installation
Component Installer
Installing Requested Components
SDK Path: C:\Users\nickd\AppData\Local\Android\Sdk
Packages to install:
- Google Play Intel x86 Atom System Image (system-images;android-28;google_apis_playstore;x86)
Preparing "Install Google Play Intel x86 Atom System Image (revision: 9)".
Downloading https://dl.google.com/android/repository/sys-img/google_apis_playstore/x86-28_r09.zip
Terminal
Build
Event Log
Emulator: Process finished with
4 spaces


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
TestApp [C:\Users\nickd\AndroidStudioProjects\TestApp] - ...\res\layout\fragment_home.xml [app]
TestApp app src main res layout fragment_home.xml
app
Pixel 2 Pie 9.0 - API 28
Android
app
manifests
java
com.example.testap
ui
dashboard
Dashboar
Dashboar
home
HomeFrag
HomeView
notifications
Notificatic
Notificatic
MainActivity
com.example.testap
com.example.testap
java (generated)
res
drawable
layout
activity_main.xml
fragment_dashbo
fragment_home.x
fragment_notifica
menu
Home
This is home fragment
Hello
Home
Dashboard
Notifications
HomeFragment.java
activity_main.xml
Pixel
29
AppTheme
Default (en-us)
0dp
Hello
Attributes
No component selected.
Select a component in the Component Tree or on the Design Surface.
Install successfully finished in 1 s 892
4: Run
TODO
Profiler
6: Logcat
Build
Terminal
Event Log
Install successfully finished in 1 s 892 ... (moments ago)
176 chars, 4 line breaks
35:1
CRLF
UTF-8
4 spaces


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
TestApp [C:\Users\nickd\AndroidStudioProjects\TestApp] - ...\res\layout\fragment_home.xml [app]
TestApp app src main res layout fragment_home.xml
app
Pixel 2 Pie 9.0 - API 28
Android
app
manifests
java
com.example.testapp
ui
dashboard
home
HomeFrag
HomeView
notifications
MainActivity
java (generated)
res
drawable
layout
activity_main.xml
menu
bottom_nav_menu
HomeFragment.java
activity_main.xml
Pixel
29
AppTheme
Default (en-us)
0dp
Attributes
No component selected.
Select a component in the Component Tree or on the Design Surface.
Hello
Home
Hello World!
This is home fragment
Hello
Home
Dashboard
Notifications
1:1
4: Run
TODO
Profiler
6: Logcat
Build
Terminal
Event Log
Install successfully finished in 926 ms. (2 minutes ago)
37:22
CRLF
UTF-8
4 spaces

SpringBoot / SpringMVC

Extension: Spring Initializr Java Support

Spring Initializr Java Support v0.9.0

Microsoft | 1,434,201 | ★★★★½☆ (5)

A lightweight extension based on Spring Initializr to generate quick start Spring Boot Java projects.

Disable | Uninstall | Switch to Pre-Release Version

Details | Feature Contributions | Changelog

Spring Initializr Java Support

VS Marketplace v0.9.2022032503 | installs 1.43M | rating 3.4/5 (5)

Overview

Spring Initializr is a lightweight extension to quickly generate a Spring Boot project in Visual Studio Code (VS Code). It helps you to customize your projects with configurations and manage Spring Boot dependencies.

Feature List

- Generate a Maven/Gradle Spring Boot project
- Customize configurations for a new project (language, Java version, group id, artifact id, boot version and dependencies)
- Search for dependencies

Categories

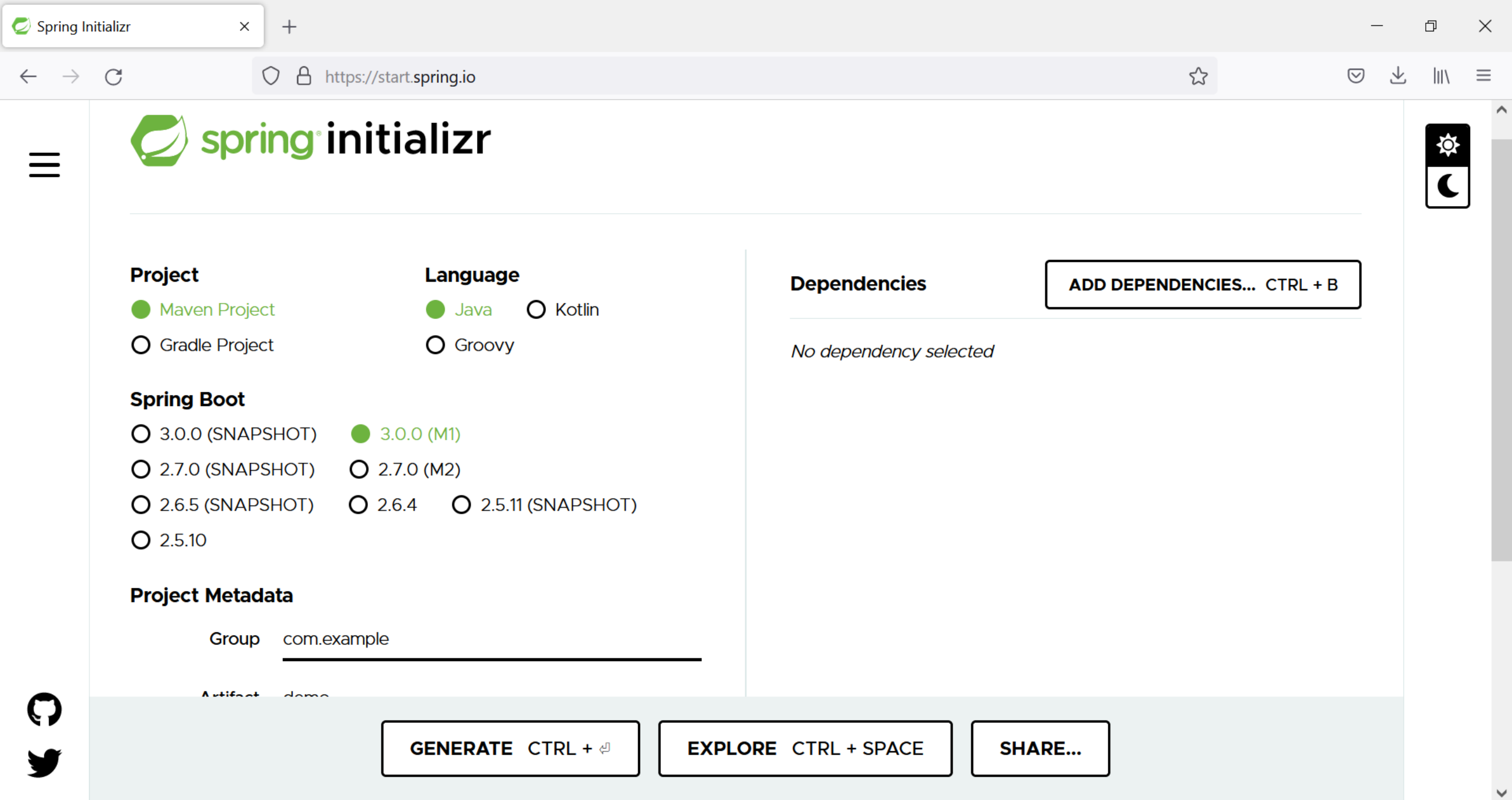
Other

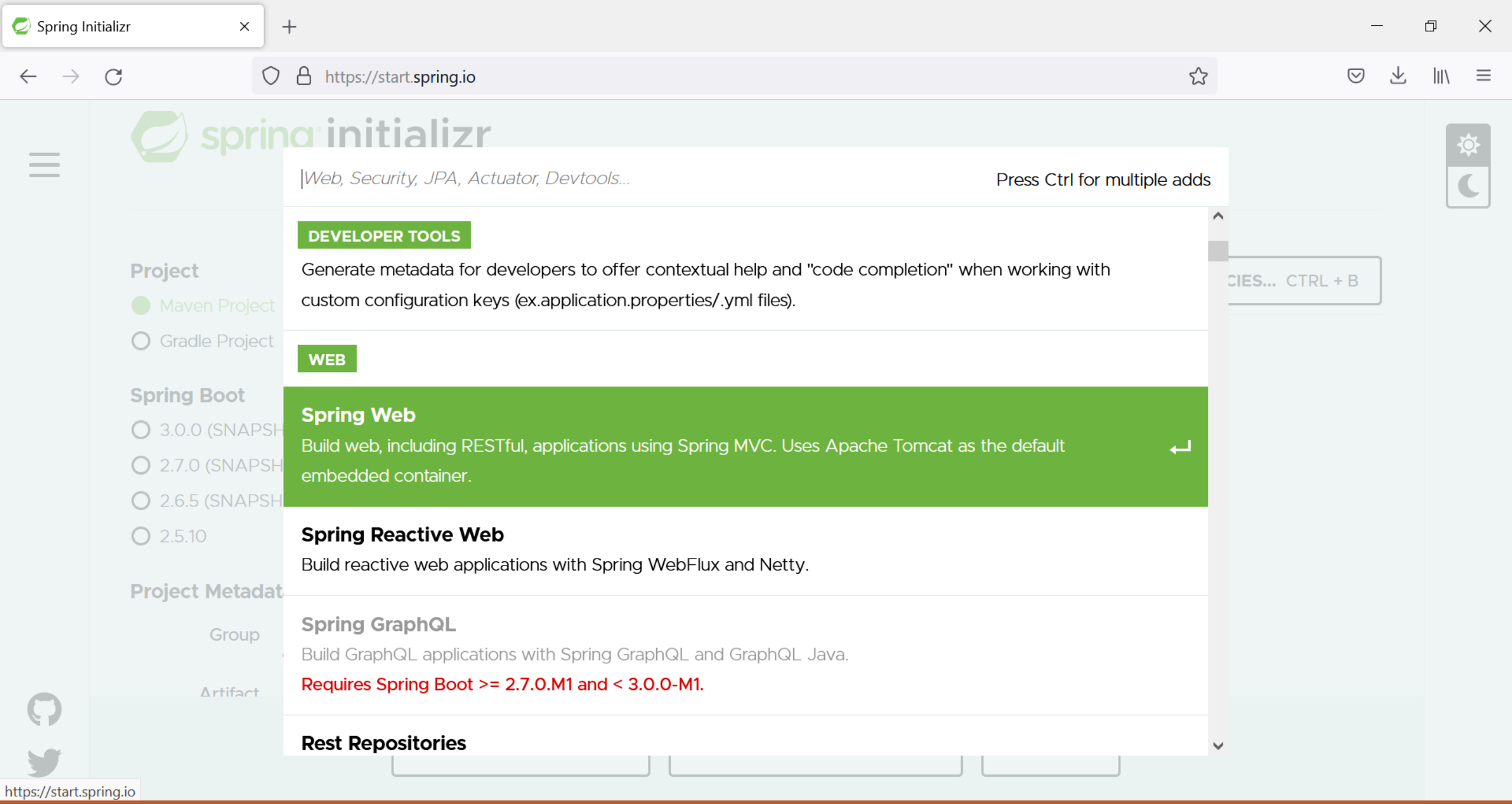
Resources

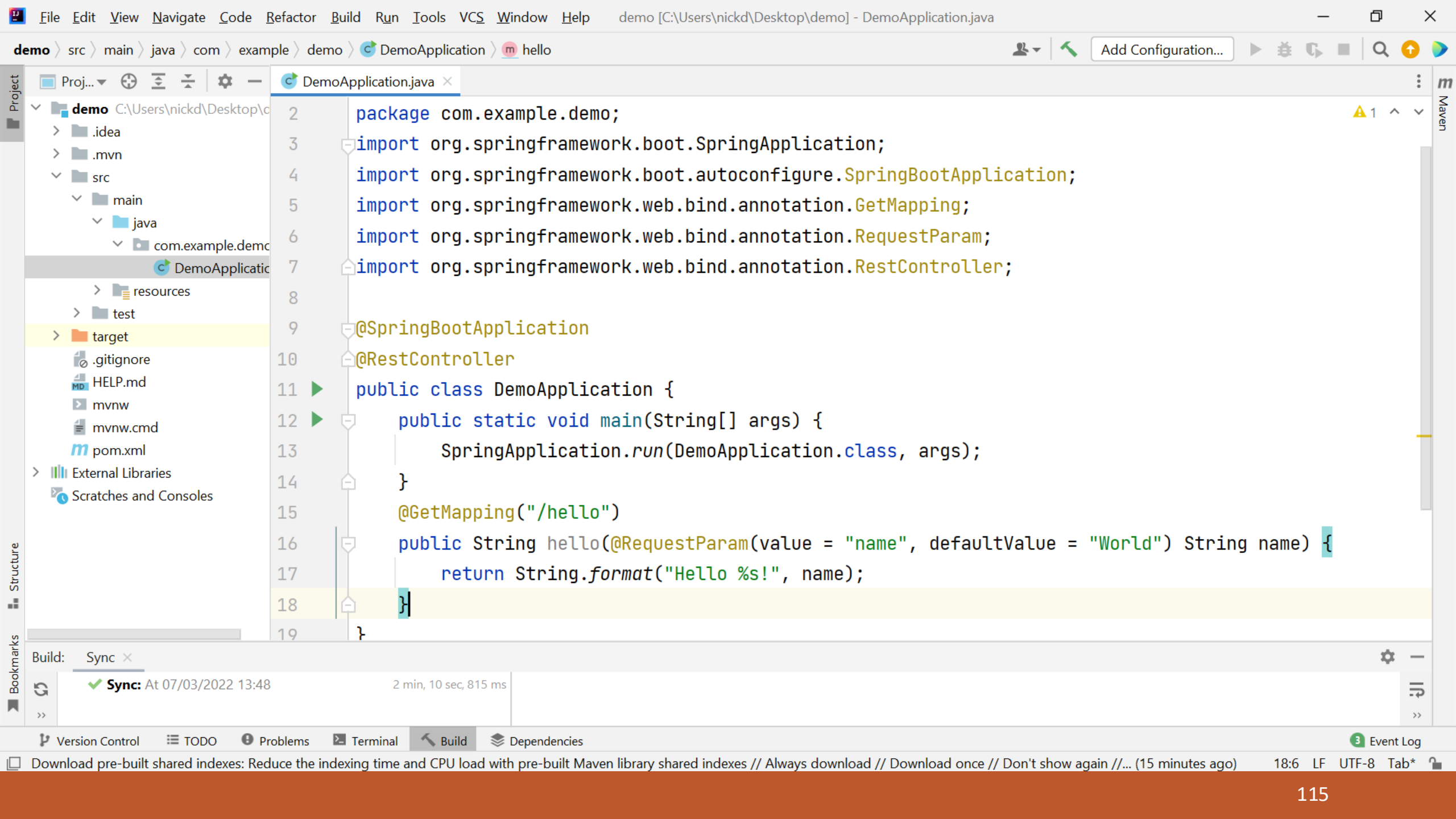
Marketplace
Repository
License
microsoft.com

Marketplace Info

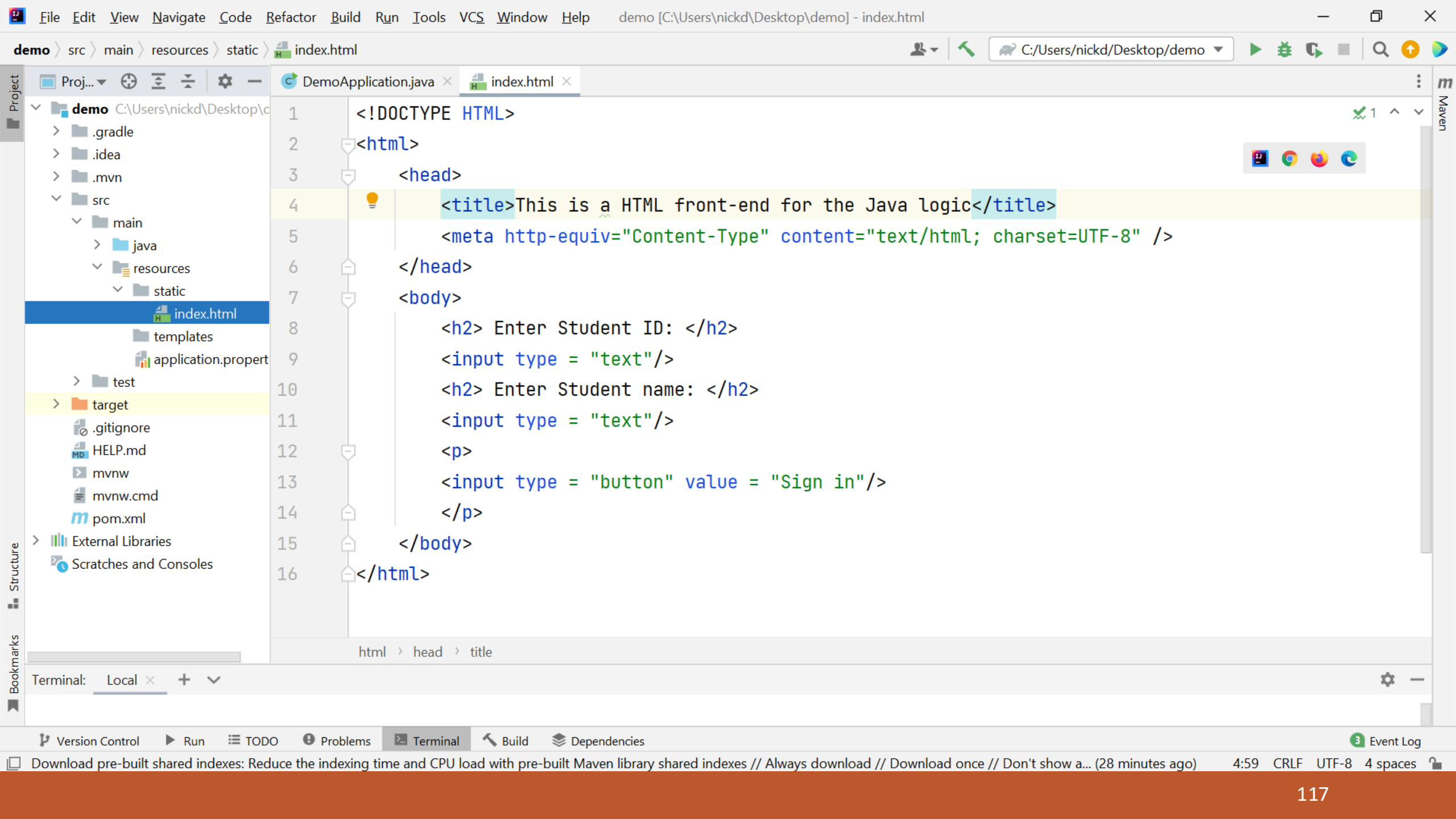
Released on	17/01/2018, 07:11:18
Last updated	25/03/2022, 03:05:16
Identifier	vscjava.vscode-spring-initializr

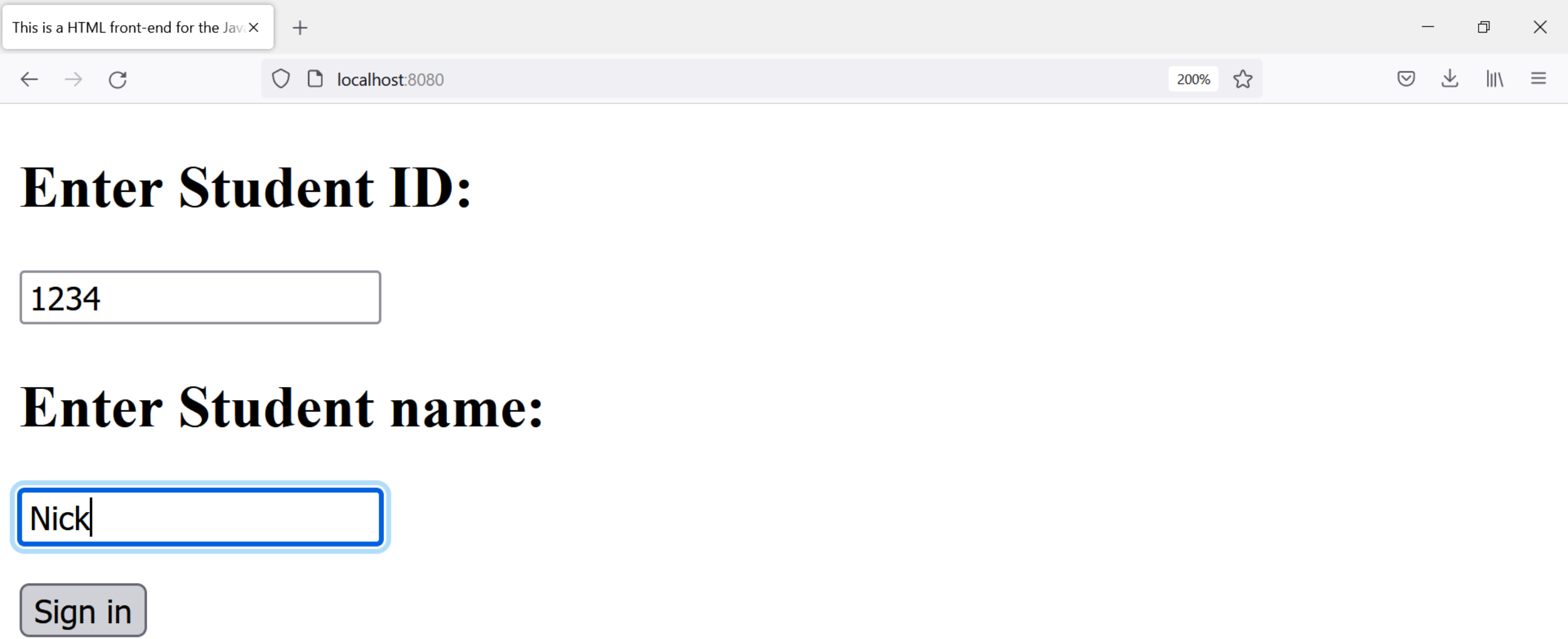

Spring Initializr
https://start.spring.io
spring initializr
Project
Maven Project
Gradle Project
Language
Java
Kotlin
Groovy
Spring Boot
3.0.0 (SNAPSHOT)
3.0.0 (M1)
2.7.0 (SNAPSHOT)
2.7.0 (M2)
2.6.5 (SNAPSHOT)
2.6.4
2.5.11 (SNAPSHOT)
2.5.10
Project Metadata
Group
com.example
Dependencies
ADD DEPENDENCIES... CTRL + B
No dependency selected
GENERATE CTRL + ⏎
EXPLORE CTRL + SPACE
SHARE...


Spring Initializr
https://start.spring.io
spring initializr
Web, Security, JPA, Actuator, Devtools...
Press Ctrl for multiple adds
DEVELOPER TOOLS
Generate metadata for developers to offer contextual help and "code completion" when working with custom configuration keys (ex.application.properties/.yml files).
WEB
Spring Web
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
Spring Reactive Web
Build reactive web applications with Spring WebFlux and Netty.
Spring GraphQL
Build GraphQL applications with Spring GraphQL and GraphQL Java.
Requires Spring Boot >= 2.7.0.M1 and < 3.0.0-M1.
Rest Repositories
Project
Maven Project
Gradle Project
Spring Boot
3.0.0 (SNAPSH
2.7.0 (SNAPSH
2.6.5 (SNAPSH
2.5.10
Project Metadat
Group
Artifact
CIES... CTRL + B
https://start.spring.io


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
demo [C:\Users\nickd\Desktop\demo] - DemoApplication.java
demo src main java com example demo DemoApplication hello
Add Configuration...
Proj...
DemoApplication.java
demo C:\Users\nickd\Desktop\c
.idea
.mvn
src
main
java
com.example.demc
DemoApplicatic
resources
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
1
2 package com.example.demo;
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.web.bind.annotation.GetMapping;
6 import org.springframework.web.bind.annotation.RequestParam;
7 import org.springframework.web.bind.annotation.RestController;
8
9 @SpringBootApplication
10 @RestController
11 public class DemoApplication {
12 public static void main(String[] args) {
13 SpringApplication.run(DemoApplication.class, args);
14 }
15 @GetMapping("/hello")
16 public String hello(@RequestParam(value = "name", defaultValue = "World") String name) {
17 return String.format("Hello %s!", name);
18 }
19 }
Build: Sync
Sync: At 07/03/2022 13:48 2 min, 10 sec, 815 ms
Version Control TODO Problems Terminal Build Dependencies Event Log
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Maven library shared indexes // Always download // Download once // Don't show again //... (15 minutes ago)
18:6 LF UTF-8 Tab*
Project
Structure
Bookmarks
Maven


localhost:8080/hello
localhost:8080/hello
410%
Hello World!


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
demo [C:\Users\nickd\Desktop\demo] - index.html
demo src main resources static index.html
C:/Users/nickd/Desktop/demo
Proj...
DemoApplication.java
index.html
demo C:\Users\nickd\Desktop\c
.gradle
.idea
.mvn
src
main
java
resources
static
index.html
templates
application.propert
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <title>This is a HTML front-end for the Java logic</title>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6 </head>
7 <body>
8 <h2> Enter Student ID: </h2>
9 <input type = "text"/>
10 <h2> Enter Student name: </h2>
11 <input type = "text"/>
12
13 <input type = "button" value = "Sign in"/>
14
15 </body>
16 </html>
html head title
Terminal: Local
Version Control Run TODO Problems Terminal Build Dependencies Event Log
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Maven library shared indexes // Always download // Download once // Don't show a... (28 minutes ago)
4:59 CRLF UTF-8 4 spaces
Project
Structure
Bookmarks
Maven


This is a HTML front-end for the Jav
localhost:8080
200%
Enter Student ID:
1234
Enter Student name:
Nick
Sign in