Beans Constructor Arguments

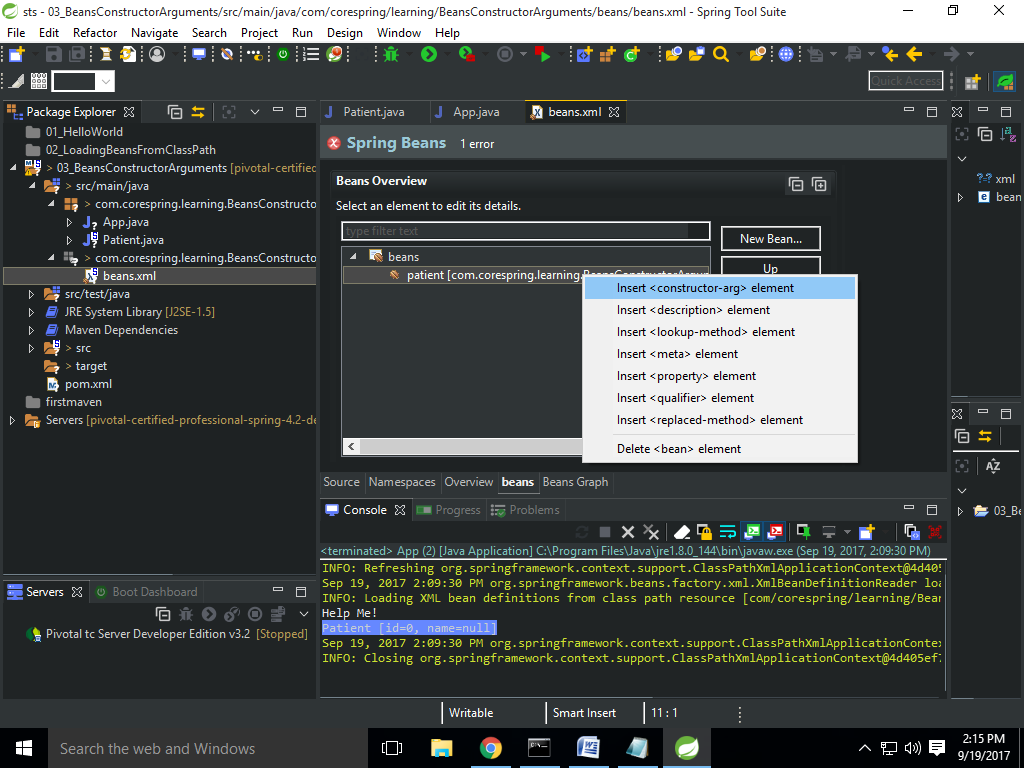
Create a project named 03\_BeansConstructorArguments and copy the previous project files ( 02\_Loading BeansFromClassPath ) at our new project and we will work from there.

1. Add id and name attribute at Patient class
2. Generate getter and setter for id and name attribute
3. Generate constructor using fields
4. Generate toString() method

So our Patient.java file looks like this

|  |
| --- |
| **package** com.corespring.learning.BeansConstructorArguments;  **public** **class** Patient {    **private** **int** id;  **private** String name;      **public** **Patient**(**int** id, String name) {  **super**();  **this**.id = id;  **this**.name = name;  }  **public** **int** **getId**() {  **return** id;  }  **public** **void** **setId**(**int** id) {  **this**.id = id;  }  **public** String **getName**() {  **return** name;  }  **public** **void** **setName**(String name) {  **this**.name = name;  }    ***@Override***  **public** String **toString**() {  **return** "Patient [id=" + id + ", name=" + name + "]";  }  **public** **void** **speak**() {  System.***out***.println("Help Me!");  }  } |

1. If we run the program then BeanCreationException will be thrown because it cannot find the constructor.
2. Insert empty constructor at Patient class and it is error free.
3. Now we will add System.***out***.println(patient);
4. It will print Patient [id=0, name=null]
5. If we want to initialize the id and name field through bean configuration
6. Go to patient bean and right click on the patient bean
7. Select Insert <constructor –arg> element



1. Type value field = 1 and it will be the first arg of constructor
2. Select Insert <constructor –arg> element and Type value field = Shekhar and it will be the second arg of constructor
3. Then run the App.java and now it will print Patient [id=1, name=Shekhar]