

- Till now we have used collection that was limited to that specific property we cannot use it outside that property if we want to use we need to declared it again but in this case (Standalone collection) will declare it once and can use wherever we want.
- We use util schema for standalone collection

in/java/com/spring/SpringStandaloneCollection/Person.java - Eclipse IDE

File Edit Run Window Help

```

Person.java x  standalc.xml  App.java
1  package com.spring.SpringStandaloneCollection;
2
3  import java.util.List;
4
5  public class Person {
6      private List<String> frd;
7
8      public List<String> getFrd() {
9          return frd;
10     }
11
12     public void setFrd(List<String> frd) {
13         this.frd = frd;
14     }
15
16     @Override
17     public String toString() {
18         return "Person [frd=" + frd + "]";
19     }
20
21 }
22
23
24

```

Problems Servers Terminal Data Source Explorer Properties Console Coverage

terminated> App (7) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (29-Nov-2023, 11:22:49 am – 11:22:50 am) [pid: 5820]

erson [frd=[Omkar, Yash, Shubham, Aniket]]java.util.LinkedList

c:/main/java/com/spring/SpringStandaloneCollection/App.java - Eclipse IDE

```
Project Run Window Help
Person.java standalc.xml App.java x
1 package com.spring.SpringStandaloneCollection;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5 public class App
6 {
7     public static void main( String[] args )
8     {
9         ApplicationContext con=new ClassPathXmlApplicationContext("com/spring/SpringStandaloneCollection/standalc.xml");
10        Person p=(Person) con.getBean("p1");
11        System.out.print(p);
12        System.out.println(p.getFrd().getClass().getName());
13    }
14 }
15 }
16

Problems Servers Terminal Data Source Explorer Properties Console Coverage
<terminated> App (7) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (29-Nov-2023, 11:22:49 am – 11:22:50 am) [pid: 5820]
Person [frd=[Omkar, Yash, Shubham, Aniket]]java.util.LinkedList
```

```
Person.java standalc.xml x App.java
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd (xsi:schemaLocation) | http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xmlns:util="http://www.springframework.org/schema/util"
6     xsi:schemaLocation="http://www.springframework.org/schema/beans
7         http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
8         http://www.springframework.org/schema/context
9         http://www.springframework.org/schema/context/spring-context.xsd
10        http://www.springframework.org/schema/util
11        http://www.springframework.org/schema/util/spring-util.xsd
12    ">
13
14    <util:list list-class="java.util.LinkedList" id="myfrds">
15        <value>Omkar</value>
16        <value>Yash</value>
17        <value>Shubham</value>
18        <value>Aniket</value>
19    </util:list>
20    <bean class="com.spring.SpringStandaloneCollection.Person" name="p1" >
21        <property name="frd" ref="myfrds" />
22    </bean>
23
24 </beans>
```

We can use same list in multiple beans :

```

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd (xsi:schemaLocation) | http://www.springframework.org/schema/util/spring-
?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:util="http://www.springframework.org/schema/util"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util.xsd"
">
<util:list list-class="java.util.LinkedList" id="myfrds">
<value>Omkar</value>
<value>Yash</value>
<value>Shubham</value>
<value>Aniket</value>
</util:list>

<bean class="com.spring.SpringStandaloneCollection.Person" name="p1" >
<property name="frd" ref="myfrds" />
</bean>

<bean class="com.spring.SpringStandaloneCollection.Person" name="p2" >
<property name="frd" ref="myfrds" />
</bean>

<bean class="com.spring.SpringStandaloneCollection.Person" name="p3" >
<property name="frd" ref="myfrds" />
</bean>

</beans>

```

The screenshot shows an IDE with two panels. The left panel displays the source code for `App.java`, which is part of the `com.spring.SpringStandaloneCollection` package. The code imports `ApplicationContext` and `ClassPathXmlApplicationContext` from the Spring framework. It defines a `main` method that creates a `ClassPathXmlApplicationContext` with the configuration file `standalc.xml`. It then retrieves three `Person` objects from the context: `p1`, `p2`, and `p3`, and prints their details. The right panel shows the console output, which confirms that the application successfully loaded the configuration and retrieved the `Person` objects with the expected first names: `Omkar`, `Yash`, `Shubham`, and `Aniket`.

```

1 package com.spring.SpringStandaloneCollection;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5 public class App
6 {
7     public static void main( String[] args )
8     {
9         ApplicationContext con=new ClassPathXmlApplicationContext("com/spring/SpringSta
10         Person p=(Person) con.getBean("p1");
11         System.out.print(p);
12         System.out.println(p.getFrd().getClass().getName());
13
14
15         Person p2=(Person) con.getBean("p2");
16         System.out.println(p2);
17
18
19
20         Person p3=(Person) con.getBean("p3");
21         System.out.println(p3);
22
23     }
24 }
25

```

```

<terminated> App (7) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (2
Person [frd=[Omkar, Yash, Shubham, Aniket]]java.util.Link
Person [frd=[Omkar, Yash, Shubham, Aniket]]
Person [frd=[Omkar, Yash, Shubham, Aniket]]

```

2)Map collection

```
Person.java × standalc.xml App.java
1 package com.spring.SpringStandaloneCollection;
2
3 import java.util.Map;
4
5 public class Person {
6     private Map<String,String> classfees;
7
8     public Map<String, String> getClassfees() {
9         return classfees;
10    }
11
12    public void setClassfees(Map<String, String> classfees) {
13        this.classfees = classfees;
14    }
15
16    @Override
17    public String toString() {
18        return "Person [classfees=" + classfees + "]";
19    }
20
21
22
23
24

```

```
Person.java × standalc.xml App.java
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd (xsi:schemaLocation) | http:
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xmlns:util="http://www.springframework.org/schema/util"
6     xsi:schemaLocation="http://www.springframework.org/schema/beans
7
8     http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
9     http://www.springframework.org/schema/context
10    http://www.springframework.org/schema/context/spring-context.xsd
11    http://www.springframework.org/schema/util
12    http://www.springframework.org/schema/util/spring-util.xsd
13    ">
14    <util:map map-class="java.util.HashMap" id="fee">
15        <entry key="java" value="8000" />
16        <entry key="python" value="5000" />
17        <entry key="Django" value="2000" />
18        <entry key="Hibernate" value="1000" />
19
20
21    </util:map>
22
23
24
25    <bean class="com.spring.SpringStandaloneCollection.Person" name="cls1" >
26        <property name="classfees" ref="fee" />
27    </bean>
28
29 </beans>

```

```
1 package com.spring.SpringStandaloneCollection;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5 public class App
6 {
7     public static void main( String[] args )
8     {
9         ApplicationContext con=new ClassPathXmlApplicationContext("com/
10         Person p=(Person) con.getBean("cls1");
11         System.out.println(p);
12         System.out.println(p.getClassfees().getClass().getName());
13
14
15
16
17     }
18 }
19
```

```
<terminated> App (7) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (29-Nov-2023, 11:43:16 ar
Person [classfees={python=5000, Hibernate=1000, java=8000, Django=2000}]
java.util.HashMap
```

In case of TreeMap we get sorted output:

```
Person.java  standalc.xml x App.java
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd (xsi:schemaLocation) | http://ww
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xmlns:util="http://www.springframework.org/schema/util"
6     xsi:schemaLocation="http://www.springframework.org/schema/beans
7
8     http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
9     http://www.springframework.org/schema/context
10    http://www.springframework.org/schema/context/spring-context.xsd
11    http://www.springframework.org/schema/util
12    http://www.springframework.org/schema/util/spring-util.xsd
13    ">
14    <util:map map-class="java.util.TreeMap" id="fee">
15        <entry key="java" value="8000" />
16        <entry key="python" value="5000" />
17        <entry key="Django" value="2000" />
18        <entry key="Hibernate" value="1000" />
19
20    </util:map>
21
22
23
24
25    <bean class="com.spring.SpringStandaloneCollection.Person" name="cls1" >
26        <property name="classfees" ref="fee" />
27    </bean>
28
29 </beans>
```

```
1 package com.spring.SpringStandaloneCollection;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5 public class App
6 {
7     public static void main( String[] args )
8     {
9         ApplicationContext con=new ClassPath
10         Person p=(Person) con.getBean("cls1"
11         System.out.println(p);
12         System.out.println(p.getClassfees().
13
14
15
16
17     }
18 }
19
```

```
<terminated> App (7) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (29-Nov-2023, 11:47:47 am - 11:47:48 am) [pid: 14956]
Person [classfees={Django=2000, Hibernate=1000, java=8000, python=5000}]
java.util.TreeMap
```

3) Properties is also like key value pair and in this case we dont need declare class it automatically inject data from bean.

Properties is class that extends hashtable

```
Person.java × standalc.xml App.java
1 package com.spring.SpringStandaloneCollection;
2
3 import java.util.Map;
4 import java.util.Properties;
5
6 public class Person {
7     private Properties pro;
8
9     public Properties getPro() {
10         return pro;
11     }
12
13     public void setPro(Properties pro) {
14         this.pro = pro;
15     }
16
17     @Override
18     public String toString() {
19         return "Person [pro=" + pro + "]";
20     }
21
22 }
```

src/main/java/com/spring/SpringStandaloneCollection/standalc.xml - Eclipse IDE

Run Window Help

```
Person.java × standalc.xml × App.java
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd (xsi:schemaLocation)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xmlns:util="http://www.springframework.org/schema/util"
6     xsi:schemaLocation="http://www.springframework.org/schema/beans
7         http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
8         http://www.springframework.org/schema/context
9         http://www.springframework.org/schema/context/spring-context.xsd
10        http://www.springframework.org/schema/util
11        http://www.springframework.org/schema/util/spring-util.xsd
12    ">
13    <util:properties id="prop">
14        <prop key="1" >One</prop>
15        <prop key="2" >Two</prop>
16        <prop key="3" >Three</prop>
17    </util:properties>
18
19    <bean class="com.spring.SpringStandaloneCollection.Person" name="pp" >
20        <property name="pro" ref="prop" />
21    </bean>
22
23 </beans>
```

