

Date : 18/12/2020

Spring Boot 9AM

Mr. RAGHU

Docs: (Adv Java, Hibernate, Spring , Spring boot, tools, webservives)
<https://www.mediafire.com/file/w5x9w5vcmkwkkdv/RaghuSirNareshITJavaPdfs.zip/file>

YAML (__.yaml)

*) If we write both properties and yaml files, then
highest priority is properties

ie if any key is present in both properties and yaml file then
Spring container reads properties data as final.

--application.properties---
my.app.id=10

--application.yaml--
my:
 app:
 id: 99

@Value("\${my.app.id}") ---> 10

XML --> JSON
Properties --> YAML

<https://www.google.com/search?q=spring+boot+2.4.1>
<https://github.com/spring-projects/spring-boot/releases/tag/v2.4.1>

*)Note: After Spring Boot 2.4 version, priority order is changed.
1st priority is --YAML
2nd priority is --Properties

application.yml
application.properties

YAML Programming -- Configuration Properties

*)Note: When we add @ConfigurationProperties to our code, it might be
showing Warning(Yellow color underline).

=> First define variables
=> Generate set/get method
=> Gerente toString
=> Add @ConfigurationProperties(prefix="")
=> place mouse over '@ConfigurationProperties' line
=> Click on option 'Add 'spring-boot-configuration-process' to

pom.xml'

=> Now goto application.properties/application.yml
then press 'ctrl+space', now you can see all suggested keys.

--code-----

#1 Create Spring Boot Starter Project

Name : SpringBoot2YamlExOne

Package : in.nareshit.raghu

#2. Runner class

package in.nareshit.raghu.runner;

import org.springframework.boot.CommandLineRunner;

import

org.springframework.boot.context.properties.ConfigurationProperties;

import org.springframework.stereotype.Component;

@Component

@ConfigurationProperties(prefix = "my.app")

public class MyDataRunner implements CommandLineRunner {

private Integer id;

private String code;

private Double cost;

@Override

public void run(String... args) throws Exception {

System.out.println(this);

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public Double getCost() {

return cost;

}

public void setCost(Double cost) {

this.cost = cost;

}

@Override

public String toString() {

return "MyDataRunner [id=" + id + ", code=" + code +

```
    ", cost=" + cost + "];";
    }
}
```

```
}
```

#3. application.yml

```
my:
  app:
    id: 88
    code: AA
    cost: 9.9
```

#4. Open starter class and run(ctrl+F11)

```
-----
Output: MyDataRunner [id=88, code=AA, cost=9.9]
```

```
=====Ex#2=====
=====
```

YAML with List/Set/Array and Map

S#1 New Spring Starter Project

Name : SpringBoot2YamlConfigPropsCollEx

Package : in.nareshit.raghu

S#2 Runner class

```
package in.nareshit.raghu;
```

```
import java.util.Arrays;
```

```
import java.util.Map;
```

```
import org.springframework.boot.CommandLineRunner;
```

```
import
```

```
org.springframework.boot.context.properties.ConfigurationProperties;
```

```
import org.springframework.stereotype.Component;
```

```
//ctrl+shift+O
```

```
@Component
```

```
@ConfigurationProperties(prefix = "my.app")
```

```
public class MyCollDataReader implements CommandLineRunner {
```

```
    //private List<String> models;
```

```
    private String[] models;
```

```
    private Map<String,Integer> grades;
```

```
    @Override
```

```
    public void run(String... args) throws Exception {
```

```
        System.out.println(this);
```

```
    }
```

```
    public String[] getModels() {
```

```
        return models;
```

```
    }
```

```

    public void setModels(String[] models) {
        this.models = models;
    }

    public Map<String, Integer> getGrades() {
        return grades;
    }

    public void setGrades(Map<String, Integer> grades) {
        this.grades = grades;
    }

    @Override
    public String toString() {
        return "MyCollDataReader [models=" +
Arrays.toString(models) + ", grades=" + grades + "]";
    }

}

```

S#3 application.yml

```

my:
  app:
    models:
      - AA
      - BB
      - CC
    grades:
      A: 1
      B: 2
      C: 3

```

Output: MyCollDataReader [models=[AA, BB, CC], grades={A=1, B=2, C=3}]

=====

```

YAML using HAS-A Relation : using class as a DataType

```

S#1 Create Spring Starter Project

Name : SpringBoot2YamlConfigPropsHasAEx

Package: in.nareshit.raghu

S#2 Model

```

package in.nareshit.raghu.model;

```

```

public class Student {

    private Integer sid;
    private String sname;
    private Double sfee;

    public Integer getSid() {
        return sid;
    }
}

```

```

        public void setSid(Integer sid) {
            this.sid = sid;
        }
        public String getSname() {
            return sname;
        }
        public void setSname(String sname) {
            this.sname = sname;
        }
        public Double getSfee() {
            return sfee;
        }
        public void setSfee(Double sfee) {
            this.sfee = sfee;
        }
        @Override
        public String toString() {
            return "Student [sid=" + sid + ", sname=" + sname + ",
sfee=" + sfee + "]";
        }
    }
}

```

S#3 Runner class

```
package in.nareshit.raghu.runner;
```

```
import org.springframework.boot.CommandLineRunner;
```

```
import
```

```
org.springframework.boot.context.properties.ConfigurationProperties;
```

```
import org.springframework.stereotype.Component;
```

```
import in.nareshit.raghu.model.Student;
```

```
//ctrl+shift+O
```

```
@Component
```

```
@ConfigurationProperties(prefix = "my.app")
```

```
public class HasADataRunner implements CommandLineRunner {
```

```
    private Student sob;//HAS-A
```

```
    @Override
```

```
    public void run(String... args) throws Exception {
        System.out.println(sob);
    }
```

```
    public Student getSob() {
        return sob;
    }
```

```
    public void setSob(Student sob) {
        this.sob = sob;
    }
```

```
}
```

S#4 application.yml

my:

app:

sob:

sfee: 77.9

sid: 999

sname: AAA

Output: Student [sid=999, sname=AAA, sfee=77.9]

=====

Comment in properties/yml --> # symbol

Q) Why @Component and @Autowired are not required for Student class in above example?

A) When we are using data loading from properties/yml using @ConfigurationProperties , it is taken care by Spring boot.