

Date : 29/12/2020

Spring Boot 9AM

Mr. RAGHU

Maven Sessions:

<https://www.youtube.com/c/NareshtIT/search?query=raghu%20maven>

Child Profiles:-

We can divide one properties file into multiple files.
Finally combine them.

=> If a properties file contains more no.of keys then
define multiple properties file and link finally as one
profile.

spring.profiles.include=<child-profile-name-1>,<child-profile-name-
2>,....

=> Child Profiles makes our application as modularized. Maintenance is
easy.

java -jar <projectName>.jar --spring.profiles.active=prod --
spring.profiles.include=proddb,prodmail

To Activate Profile:

--spring.profiles.active=prod

To Include Profile:

--spring.profiles.include=proddb,prodmail

*) If same key is present in all locations then last loaded file is
considered as first priority.

-----code-----

#1 Create one project

Name : SpringBoot2ChildProfiles

Dep : Lombok

#2. Runner class

package in.nareshtit.raghu.runner;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

import lombok.ToString;

@Component

@ToString

public class DataReadRunner implements CommandLineRunner {

 @Value("\${my.app.title}")

 private String title; //prod

 @Value("\${my.app.drivers}")

```

        private String driver; //proddb

        @Value("${my.app.host}")
        private String host; //prodmail

        @Override
        public void run(String... args) throws Exception {
            System.out.println(this);
        }
    }
}

```

#3. properties files

```

--application.properties--
my.app.title=NIT-DEFAULT

--application-prod.properties--
my.app.title=NIT-A
#spring.profiles.include=proddb,prodmail

---application-proddb.properties--
my.app.drivrs=oracle
#my.app.title=NIT-B

---application-prodmail.properties--
my.app.host=Gmail
#my.app.title=NIT-C
-----

```

#4 Run Starter class with Args

```

--spring.profiles.active=prod <space> --
spring.profiles.include=proddb,prodmail
=====

```

==

```

in pom.xml Modify version as:
<version>2.3.6.RELEASE</version>

```

> Right click on Project > Maven > Update Poject > Finish

=====

Profiles using YAML

We can define YAML files for every profile as like proeprties file.
ie

application.yml	[default profile]
application-qa.yml	[qa profile]
application-prod.yml	[prod profile]
..etc	

Or else we can define single YAML file using profile separator 3 dash (---).

application.yml

key: val

```
#default profile
```

```
---
```

```
key: val
```

```
spring:
```

```
  profiles: profilename
```

```
---
```

```
key: val
```

```
spring:
```

```
  profiles: profilename
```

```
---code-----
```

```
#1 Create Project
```

```
Name : SpringBoot2ProfileYaml
```

```
Dep   : Lombok
```

```
#2. runner class
```

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

```
import org.springframework.beans.factory.annotation.Value;
```

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

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

```
import lombok.ToString;
```

```
@Component
```

```
@ToString
```

```
public class MyDataRunner implements CommandLineRunner {
```

```
    @Value("${my.app.title}")
```

```
    private String code;
```

```
    @Override
```

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

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

```
    }
```

```
}
```

```
#3. application.yml
```

```
my:
```

```
  app:
```

```
    title: NIT-DEF
```

```
---
```

```
my:
```

```
  app:
```

```
    title: NIT-QA
```

```
spring:
```

```
  profiles: qa
```

```
---
```

```
my:
```

```
  app:
```

```
    title: NIT-PROD
spring:
  profiles: prod
```

S#4 Run Starter class with --spring.profiles.active=qa

Cloud Deployments:

<https://www.youtube.com/watch?v=QOwgiJWmZ9k>