HowToDoInJava

Spring Boot – Inject Application Arguments in @Bean and @Compoment

Last Updated: October 24, 2019 By: Lokesh Gupta Spring Boot 2

Learn to retrieve and access the **application runtime arguments** in `@Component` annotated classes and `@Bean` annotated methods in a Spring boot application using org.springframework.boot.ApplicationArguments class.

1. ApplicationArguments as constructor injection

This is fairly simple way to gain access to application arguments where we need to use them in constructor itself.

```
ArgsComponent.java

@Component
public class ArgsComponent
{
  @Autowired
  public ArgsComponent(ApplicationArguments args)
  {
    //Aceess arguments using args
  }
}
```

2. ApplicationArguments as autowired dependency

If we do not specifically require arguments in constructor, autowiring is more cleaner way to inject `ApplicationArguments` class in any spring component or configuration class.

```
AppConfiguration.java

@Configuration
public class AppConfiguration
{
  @Autowired
  private ApplicationArguments args;

  @Bean
  public ArgsComponent argsComponent() {

    //access args

    return new ArgsComponent();
  }
}
```

3. Inject Command Line Arguments – Demo

Let's run a quick demo to understand it's usage.

In this demo, we are passing two arguments while running the spring boot application from command prompt.

- `—emailClient`=*Java*
- `disableDataService`

The first argument (*emailClient*) is assigned a value. Second argument (*disableDataService*) is non-option argument. We will access both argument in the application.

> Each word after the run command is an argument. The arguments that start with `'-'` are option argument; and others are non-option arguments.

```
ArgsComponent.java
```

```
@Component
public class ArgsComponent
{
  public ArgsComponent()
  {
  }
}
```

```
AppConfiguration.java

import java.util.List;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.ApplicationArguments;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class AppConfiguration
{
  private static final Logger log = LoggerFactory.getLogger(AppConfiguration.class

  @Autowired
  private ApplicationArguments args;

  @Bean
  public AppConfiguration argsComponent() {

    verifyArguments();

    return new ArgsComponent(args);
  }

  public void verifyArguments()
  {
    //Check is argument is present
    if(args.containsOption("emailClient"))
    {
      //Get argument values
      List<String> values = args.getOptionValues("emailClient");

      log.info("Email clients are :: " + values);
    }

    //////////////

    List<String> nonOptionArgs = args.getNonOptionArgs();
```

```
      log.info("Non Option Args List ...");

      if (!nonOptionArgs.isEmpty())
      {
        nonOptionArgs.forEach(file -> log.info(file));
      }
    }
  }
```

Start the application and verify output

To Start the application with arguments, run following command from command prompt.

```
$ java -jar target\SpringBoot2Demo-0.0.1-SNAPSHOT.jar –emailClient=Java
disableDataService
```

Observe the startup logs.

```
Console

2019-05-06 16:31:36.443  INFO com.howtodoinjava.demo.ArgsComponent     : Email cli
2019-05-06 16:31:36.443  INFO com.howtodoinjava.demo.ArgsComponent     : Non Optio
2019-05-06 16:31:36.445  INFO com.howtodoinjava.demo.ArgsComponent     : disableDa

2019-05-06 16:31:36.461  INFO com.howtodoinjava.demo.AppConfiguration  : Email cli
2019-05-06 16:31:36.476  INFO com.howtodoinjava.demo.AppConfiguration  : Non Optio
2019-05-06 16:31:36.477  INFO com.howtodoinjava.demo.AppConfiguration  : disableDa
```

Drop me your questions related to this **spring boot command line arguments example** to demonstrate access *command line arguments while executing spring boot jar* application.

Happy Learning !!

Was this post helpful?

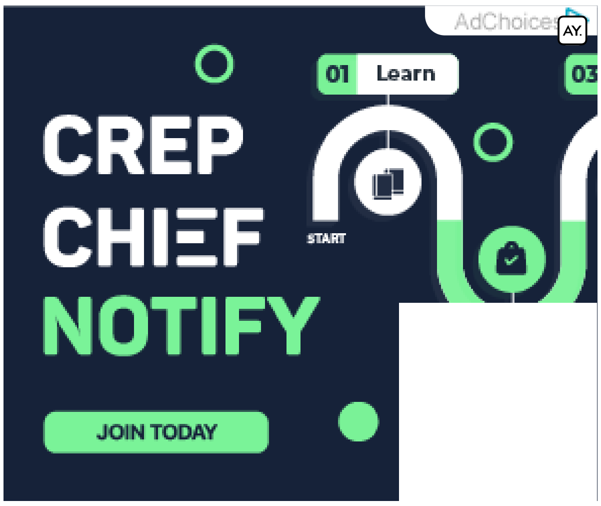
Let us know if you liked the post. That's the only way we can improve.

Yes

No

Recommended Reading:

1. Inject Spring dependencies in Quartz Job
2. Python unpack tuple into variables or arguments
3. Java Command Line Arguments
4. Mockito – Verify multiple method calls with different arguments
5. Spring Boot – CRUD Application
6. Spring boot logging with application.properties
7. Spring Boot Logging with application.yml
8. How to Deploy Spring Boot Application to Cloud Foundry Platform
9. Spring – Application events
0. Java Web Application Performance Improvement



Join 7000+ Awesome Developers

Get the latest updates from industry, awesome resources, blog updates and much more.

Email Address

Subscribe

** We do not spam !!*

Leave a Comment

Name *

Email *

Website

☐ Add me to your newsletter and keep me updated whenever you publish new blog posts

Post Comment

Search …

HowToDoInJava

A blog about Java and related technologies, the best practices, algorithms, and interview questions.

Meta Links

- About Me
- Contact Us
- Privacy policy
- Advertise
- Guest Posts

Blogs

REST API Tutorial