

Title: SpringBoot project, Environment Variable in application.yml always empty

Post Body:

I'm trying to run a SpringBoot project that runs with docker-compose. i started its dependencies (Redis, MongoDB ,and RabbitMQ) with `docker-compose up`

and i'm building the project and running it with these commands `mvn clean package -DskipTests && mvn spring-boot:run`

I keep having these errors :

Error processing condition on `org.springframework.boot.autoconfigure.security.oauth2.client.OAuth2RestOperationsConfiguration$RequestScopedConfiguration`

Caused by: `java.lang.IllegalArgumentException: Could not resolve placeholder 'OAUTH_CLIEN`  
`TID' in value`  
`'${OAUTH_CLIEN`  
`TID}'`

In the docker-compose file, the values of the environment variables are defined in the environment section.

```
environment:
  - RABBIT_HOST=rabbitstomp
  - RABBIT_USER=guest
  - RABBIT_PASS=user
  - MONGO_HOST=mongodb://
```

This is how application.yml looks like :

```
spring.data.mongodb:
  database: ${DB_NAME} #notificationdb
  uri: ${MONGO_HOST}
  security:
    oauth2:
      resource:
```

So when running the project without docker-compose, am i supposed to put the values in the application.yml ?

i also tried `mvn spring-boot:run -Dspring-boot.run.arguments=--path.to.value=value1`, but i'm not sure how the path should be with variables like `key-set-uri: ${auth-server:${SSO_HOST}}/keys`

Accepted Answer:

Spring has support for providing default values in the configuration via the [PlaceholderConfigurerSupport](#). The default value is what comes after the `:`. In your case, you should write:

```
client-id: ${OAUTH_CLIEN
```

If you use the `@Value` annotation to inject the values, you have support to SpEL for using expressions of the type: `#{someExpression}` for more complex cases.

**UPDATE:**

In your case, I believe you are reversing the position of the variables. The correct should be:

```
key-set-uri: ${SSO_HOST:auth-server}/keys
```

Here is what it means: first, it will try to use the `SSO_HOST` environment variable, which is provided to the container through docker-compose. In case this variable is not provided to the process, Spring will use `auth-server` as the address of the server. It seems to me that this address is visible only inside the docker-compose network, so if you are running your app outside this network, the `auth-server` address will not be visible. Do you know where is the auth server? Is it another docker container? Is it running on localhost?

Some interesting reference: <https://www.baeldung.com/spring-value-defaults>

Highest Rated Answer:

Pass env variables in docker compose as object not a list:

```
environment:
  RABBIT_HOST: rabbitstomp
  RABBIT_USER: guest
  RABBIT_PASS: user
  MONGO_HOST: mongodb://localhost:27017
  OAU
```