Title: Read docker env variable inside Spring application.yaml

Post Body:

I'm setting an environment variable inside my docker-compose.yaml file and want to use that variable's value inside my Spring Boot's application.yaml. I was told that doing something like

```
app:   auth:     tokenSecret: tokensecretvaluehere     tokenExpirationMsec: 864000000   oauth2:     sso:       url: ${SSO_URL}
```

(where SSO_URL is defined in my docker-compose.yaml) in my Spring application.yaml. However, this causes an error when I run docker-compose up --build because it can't find that variable (error is like: Could not resolve placeholder SSO_URL in value '${SSO_URL}'). This is an example of what my docker-compose.yaml:

```
api:     restart: always     ports:       - '8080:8080'     links:       - redis       - db     environment:       - SERVER_SE
```

I was asked to not uses the System.getenv functions in Java and instead set the variable like above. From there I would just use the @Value annotation to get it in my Java code as like below:

```
@Value('${app.oauth2.sso.url}') private String ssoUrl;
```

This is more of the application.yaml:

```
heb:   togglr:     jwt:       secret:         id: 101       session:         seconds: 600       tokenheader: X-TOGGLR-TOKEN lo
```

Accepted Answer:

Solution was to not use an underscore character in the variable name.

Highest Rated Answer:

In general spring boot applications are able to read the Environment variables accessible in the docker container. Its stated in the documentation (see item 10 in the list at the very beginning of the document).

So the problem might be elsewhere:

1. It might be a typo in the question, but if you're using application.yaml as opposed to application properties, then you should have something like:

   ```
   sso:     url: ${SSO_URL}
   ```

2. Make sure, that the env variable SSO_URL is indeed accessible in the container even before you start the spring boot application. In java (for debugging purposes only) you can do something like:

```
@SpringBootApplication public class MyApp {    public static void main(String [] args) {          System.out.println(System.g
```