

Setting Properties in Profiles

Passing env variables to our java test.

Integration tests deploying to multiple targets and target host is testing against api, the target host can change environment and use maven build profiles to control it.

By setting up environment variables we can get it in unit test or integration test

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.2.2</version>
      <configuration>
        <environmentVariables>
          <TEST_HOST>test.example.com</TEST_HOST>
        </environmentVariables>
      </configuration>
    </plugin>
  </plugins>
</build>
```

```
package org.pratik.example;
```

```
import org.junit.jupiter.api.Test;
```

```
import static org.junit.jupiter.api.Assertions.*;
```

```
class ExampleClassTest {
    @Test
    void getHello(){
        System.out.println("-----");
        System.out.println(System.getenv("TEST_HOST"));
    }
}
```

We can do this with build

```
<profiles>
  <profile>
    <id>test</id>
    <properties>
      <TEST_HOST>test.example.com</TEST_HOST>
    </properties>
  </profile>
  <profile>
    <id>uat</id>
```

```
<properties>
  <TEST_HOST>test.example.com</TEST_HOST>
</properties>
</profile>
</profiles>
```

Now if we activate either profile we will get the value of TEST_HOST for that. If we do not do then build one will get executed.

Depending on different env active we will get that value.