**Maven ve Kafka Consumer Code**

**Kafka’da herhangi bir Java projesi oluşturacağımızda bazen gerekli kütüphaneleri bulamayabiliriz.Bunun en kolay çözümü Maven projesi oluşturmaktır.Öncelikle Mavenden bahsetmek gerekirse :**

**Maven Nedir?**

Maven, geliştirilen java projelerinin geliştirme süreçlerini basitleştirmek, standartlaştırmak,dökümantasyon hazırlamak, kütüphane bağımlılıklarını kullanıcı zahmetinden kurtarmak  için kullanılan bir araçtır. Maven ile hemen hemen bütün java projelerinin iskeleti oluşturulabilir. Bu iskelet standartlaştığı için, siz oluşturduğunuz projeyi geliştirme ortamınızdan bağımsız olarak geliştirirsiniz. Proje iskeleti standart olduğu için java projenizi ister eclipse ile ister netbeans ile isterseniz başka ide lerle geliştirebilirsiniz. Bir java projesi oluşturuyorsunuz ve bazı jar kütüphanelerine ihtiyacınız var maven bu kütüphaneleri kendi repository sinden indirerek projeye dahil ediyor. **Bağımlı kaynaklar,** projede kullanılacak tüm kütüphaneler ve eklentiler **POM**(Project Object Model) dosyasından kolayca yönetilebilmektedir.

**Maven Kurulumu?**

After verifying java version on you system. Download Apache maven from its official website or use following command to download Apache Maven 3.3.9.

$ cd /opt

$ wget http://www-eu.apache.org/dist/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz

Now extract downloaded archive using following command.

$ sudo tar xzf apache-maven-3.3.9-bin.tar.gz

$ sudo ln -s apache-maven-3.3.9 maven

**Step 3 – Setup Environment Variables**

As you have downloaded pre compiled Apache Maven files on your system. Now set the environments variables by creating new file /etc/profile.d/maven.sh.

$ sudo vi /etc/profile.d/maven.sh

and add following content.

export M2\_HOME=/opt/maven

export PATH=${M2\_HOME}/bin:${PATH}

Now load the environment variables in current shell using following command.

$ source /etc/profile.d/maven.sh

## Example POM

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.mkyong</groupId>

<artifactId>NumberGenerator</artifactId>

<packaging>jar</packaging>

<version>1.0-SNAPSHOT</version>

<name>NumberGenerator</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

POM içerisinde kullanabilecegimiz bazı tag’lere bakalım;

* packaging : Paketleme tipini belirtir, default olarak jar’dır. Jar, War, Rar vs gibi tipler kullanılabilir.
* name : Projeye isim vermek içindir. Verilmesinden yanayım.
* url : Proje’nin adres niteliğindedir.
* dependencies : Uygulama bağımlılıklarının tanımlandığı kısımdır.
* scope : Proje ortamını belirler, compile, test, runtime olabilir.
* properties, mailingList, parent vb gibi kısımlara ilerleyen zamanlarda yazılar içerisinde değinecegim.

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.companyname.project-group</groupId>

<artifactId>project</artifactId>

<version>1.0</version>

</project>

Pom’da olması zorulunlu olan groupId, artifactId ve version’dur, bunları proje olusturma aşamasında zaten set ediyoruz. Version defalut olarak 1.0.0-SNAPSHOT’dır. GroupId -> company name, articactId -> project name’dir.

|  |  |
| --- | --- |
|  | <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>com.wora</groupId>  <artifactId>caysever-app</artifactId>  <packaging>jar</packaging>  <version>1.0.0</version>  <name>caysever-app @javaman</name>  <url>alicanakkus.com</url>  <developers>  <developer>  <id>caysever</id>  <name>AlicanAkkus</name>  <email>aakkus@alicanakkus.com</email>  <url>http://alicanakkus/about</url>  <roles>  <role>developer</role>  </roles>  </developer>  </developers>  <dependencies>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>3.8.1</version>  <scope>test</scope>  </dependency>  </dependencies>  </project> |

Maven ile Basit Java Projesi

Projenin pom.xml kısmını yazmaya başlıyoruz.

**pom.xml:**

<project xmlns="<http://maven.apache.org/POM/4.0.0>" xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"

  xsi:schemaLocation="<http://maven.apache.org/POM/4.0.0> <http://maven.apache.org/xsd/maven-4.0.0.xsd>">

  <modelVersion>4.0.0</modelVersion>

  <groupId>6</groupId>

  <artifactId>TestApp</artifactId>

  <version>1.0-SNAPSHOT</version>

  <packaging>jar</packaging>

  <name>TestApp</name>

  <url>[http://maven.apache.org](http://maven.apache.org/)</url>

  <properties>

    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

  </properties>

  <dependencies>

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>3.8.1</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

</project>

**App.java:**

package com.ozgul.gozacanlar;

public class App

{

    public static void main( String[] args )

    {

        System.out.println( "Hello Word! First Maven Project :)" );

    }

}

Maven projesini derlemek ve çıktısını almak için:

java -cp target/TestApp-1.0-SNAPSHOT.jar com.ozgul.gozacanlar.App