

JSON

```
<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>org.pratik</groupId>
  <artifactId>Json</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>Json</name>
  <url>https://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>
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-lang3</artifactId>
      <version>3.12.0</version>
    </dependency>
    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>2.9.1</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.jsonschema2pojo</groupId>
        <artifactId>jsonschema2pojo-maven-plugin</artifactId>
        <version>0.5.1</version>
        <configuration>
          <sourceDirectory>${basedir}/src/main/resources/schema</
sourceDirectory>
```

```

        <targetPackage>pratik.maven.model</targetPackage>
        <useCommonsLang3>true</useCommonsLang3>
    </configuration>
    <executions>
        <execution>
            <goals>
                <goal>generate</goal>
            </goals>
        </execution>
    </executions>
</plugin>
</plugins>
</build>
</project>

```

Keep the following in resources/schema folder

```
package pratik.maven.model;
```

```

import java.util.HashMap;
import java.util.Map;
import com.fasterxml.jackson.annotation.JsonAnyGetter;
import com.fasterxml.jackson.annotation.JsonAnySetter;
import com.fasterxml.jackson.annotation.JsonIgnore;
import com.fasterxml.jackson.annotation.JsonInclude;
import com.fasterxml.jackson.annotation.JsonProperty;
import com.fasterxml.jackson.annotation.JsonPropertyOrder;
import org.apache.commons.lang3.builder.EqualsBuilder;
import org.apache.commons.lang3.builder.HashCodeBuilder;
import org.apache.commons.lang3.builder.ToStringBuilder;

```

```
@JsonInclude(JsonInclude.Include.NON_NULL)
```

```

@JsonPropertyOrder({
    "post-office-box",
    "extended-address",
    "street-address",
    "locality",
    "region",
    "postal-code",
    "country-name"
})

```

```
public class Address {
```

```

    @JsonProperty("post-office-box")
    private String postOfficeBox;

```

```

@JsonProperty("extended-address")
private String extendedAddress;
@JsonProperty("street-address")
private String streetAddress;
/**
 *
 * (Required)
 */
@JsonProperty("locality")
private String locality;
/**
 *
 * (Required)
 */
@JsonProperty("region")
private String region;
@JsonProperty("postal-code")
private String postalCode;
/**
 *
 * (Required)
 */
@JsonProperty("country-name")
private String countryName;
@JsonIgnore
private Map<String, Object> additionalProperties = new HashMap<String,
Object>();

```

```

@JsonProperty("post-office-box")
public String getPostOfficeBox() {
    return postOfficeBox;
}

```

```

@JsonProperty("post-office-box")
public void setPostOfficeBox(String postOfficeBox) {
    this.postOfficeBox = postOfficeBox;
}

```

```

@JsonProperty("extended-address")
public String getExtendedAddress() {
    return extendedAddress;
}

```

```

@JsonProperty("extended-address")

```

```
public void setExtendedAddress(String extendedAddress) {  
    this.extendedAddress = extendedAddress;  
}
```

```
@JsonProperty("street-address")  
public String getStreetAddress() {  
    return streetAddress;  
}
```

```
@JsonProperty("street-address")  
public void setStreetAddress(String streetAddress) {  
    this.streetAddress = streetAddress;  
}
```

```
/**  
 *  
 * (Required)  
 */  
@JsonProperty("locality")  
public String getLocality() {  
    return locality;  
}
```

```
/**  
 *  
 * (Required)  
 */  
@JsonProperty("locality")  
public void setLocality(String locality) {  
    this.locality = locality;  
}
```

```
/**  
 *  
 * (Required)  
 */  
@JsonProperty("region")  
public String getRegion() {  
    return region;  
}
```

```
/**  
 *  
 * (Required)
```

```

*
*/
@JsonProperty("region")
public void setRegion(String region) {
    this.region = region;
}

@JsonProperty("postal-code")
public String getPostalCode() {
    return postalCode;
}

@JsonProperty("postal-code")
public void setPostalCode(String postalCode) {
    this.postalCode = postalCode;
}

/**
 *
 * (Required)
 */
@JsonProperty("country-name")
public String getCountryName() {
    return countryName;
}

/**
 *
 * (Required)
 */
@JsonProperty("country-name")
public void setCountryName(String countryName) {
    this.countryName = countryName;
}

@JsonProperty("country-name")
public void setCountryName(String countryName) {
    this.countryName = countryName;
}

@JsonProperty("country-name")
public void setCountryName(String countryName) {
    this.countryName = countryName;
}

@JsonAnyGetter
public Map<String, Object> getAdditionalProperties() {
    return this.additionalProperties;
}

@JsonAnySetter
public void setAdditionalProperty(String name, Object value) {
    this.additionalProperties.put(name, value);
}

```

```

@Override
public String toString() {
    return new ToStringBuilder(this).append("postOfficeBox",
postOfficeBox).append("extendedAddress",
extendedAddress).append("streetAddress", streetAddress).append("locality",
locality).append("region", region).append("postalCode",
postalCode).append("countryName",
countryName).append("additionalProperties", additionalProperties).toString();
}

@Override
public int hashCode() {
    return new
HashCodeBuilder().append(postOfficeBox).append(streetAddress).append(pos
talCode).append(locality).append(countryName).append(additionalProperties).
append(extendedAddress).append(region).toHashCode();
}

@Override
public boolean equals(Object other) {
    if (other == this) {
        return true;
    }
    if ((other instanceof Address) == false) {
        return false;
    }
    Address rhs = ((Address) other);
    return new EqualsBuilder().append(postOfficeBox,
rhs.postOfficeBox).append(streetAddress,
rhs.streetAddress).append(postalCode, rhs.postalCode).append(locality,
rhs.locality).append(countryName,
rhs.countryName).append(additionalProperties,
rhs.additionalProperties).append(extendedAddress,
rhs.extendedAddress).append(region, rhs.region).isEquals();
}
}

```