JSON, Jackson, and Maven example

Posted on October 20, 2015 by Arulkumaran Kumaraswamipillai

0 Like Share Tweet 0 G+1 Share

Step 1: Create a new simple-json project using Maven CLI.

```
mvn archetype:generate -DgroupId=com.mytutorial -
```

press “Enter” for all the prompts. This creates a new “simple-json” folder with MVN artefacts like pom.xml, etc.

Step 2: Import the maven project into your IDE. E.g. in eclipse File –> Import –> Existing Maven Projects and browse the “simple-json” folder with the pom.xml you created in Step 1. Press finish.

Step 3: Make changes to the pom.xml to include JSON libraries

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all

⊞ Ice Breaker Interview
⊞ Core Java Interview (
⊞ JEE Interview Q&A (3
⊞ Pressed for time? Jav
⊟ SQL, XML, UML, JSC
⊞ ERD (1)
⊞ NoSQL (2)
⊞ Regex (2)
⊞ SQL (7)
⊞ UML (1)
⊟ JSON (2)
JSON interview (
JSON, Jackson,
⊞ XML (2)
⊞ XSD (2)
⊞ YAML (2)
⊞ Hadoop & BigData Int
⊞ Java Architecture Inte
⊞ Scala Interview Q&As
⊞ Spring, Hibernate, & I
⊞ Testing & Profiling/Sa

```

<project xmlns="http://maven.apache.org/POM/4.0.(
    xsi:schemaLocation="http://maven.apache.org/
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.mytutorial</groupId>
    <artifactId>simple-json</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>simple-json</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</pro
        <jackson.version>2.4.4</jackson.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>

        <!-- JSON library  -->
        <dependency>
            <groupId> com.fasterxml.jackson.core
            <artifactId>jackson-databind</artifa
            <version>${jackson.version}</version
        </dependency>

        <dependency>
            <groupId> com.fasterxml.jackson.core
            <artifactId>jackson-core</artifactId
            <version>${jackson.version}</version
        </dependency>

        <dependency>
            <groupId> com.fasterxml.jackson.core
            <artifactId>jackson-annotations</art
            <version>${jackson.version}</version
        </dependency>
    </dependencies>
</project>
```

Step 4: Write the domain Java object with the annotations to be able to Marshall & Unmarshall. Aslo, not the serialization classes to covert LocalDate object to String and vice-versa.

```
package com.mytutorial.model;

import java.io.IOException;
import java.io.Serializable;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Objects;

import com.fasterxml.jackson.annotation.JsonInc
import com.fasterxml.jackson.core.JsonGeneratio
```

- ⊞ Other Interview Q&A f
- ⊞ ▶Free Java Interviev

16 Technical Key Areas

open all | close all

- ⊞ Best Practice (6)
- ⊞ Coding (26)
- ⊞ Concurrency (6)
- ⊞ Design Concepts (7)
- ⊞ Design Patterns (11)
- ⊞ Exception Handling (3
- ⊞ Java Debugging (21)
- ⊞ Judging Experience I
- ⊞ Low Latency (7)
- ⊞ Memory Managemen
- ⊞ Performance (13)
- ⊞ QoS (8)
- ⊞ Scalability (4)
- ⊞ SDLC (6)
- ⊞ Security (13)
- ⊞ Transaction Managen

80+ step by step Java Tutorials

open all | close all

- ⊞ Setting up Tutorial (6)
- ⊞ Tutorial - Diagnosis (2
- ⊞ Akka Tutorial (9)
- ⊞ Core Java Tutorials (2
- ⊞ Hadoop & Spark Tuto
- ⊞ JEE Tutorials (19)
- ⊞ Scala Tutorials (1)
- ⊞ Spring & HIbernate Tu
- ⊞ Tools Tutorials (19)
- ⊞ Other Tutorials (45)

```
import com.fasterxml.jackson.core.JsonGenerator
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.core.JsonProcessin
import com.fasterxml.jackson.databind.Deseriali
import com.fasterxml.jackson.databind.JsonDeser
import com.fasterxml.jackson.databind.JsonSeria
import com.fasterxml.jackson.databind.ObjectMap
import com.fasterxml.jackson.databind.Serialize
import com.fasterxml.jackson.databind.annotatio
import com.fasterxml.jackson.databind.annotatio

//Java 8 code
public class ClientEvent implements Serializabl

    private static final long serialVersionUID
    public static final String DATE_FORMAT = "d

    private String clientCode;

    @JsonDeserialize(using = CustomLocalDateDes
    @JsonSerialize(using = CustomLocalDateSeria
    private LocalDate eventDate;

    public ClientEvent() {
        super();
    }

    public String getClientCode() {
        return clientCode;
    }

    public void setClientCode(String clientCode
        this.clientCode = clientCode;
    }

    public LocalDate getEventDate() {
        return eventDate;
    }

    public void setEventDate(LocalDate eventDat
        this.eventDate = eventDate;
    }

    @Override
    public boolean equals(Object obj) {
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
        final ClientEvent other = (ClientEvent)

        return Objects.equals(this.clientCode,
                && Objects.equals(this.eventDat
    }

    @Override
    public int hashCode() {
        return Objects.hash(this.clientCode, th
    }

    @Override
    public String toString() {
        return Objects.toString(this);
    }
```

100+ Java pre-interview coding tests

open all | close all

- ⊞ Can you write code? (
- ⊞ ♦ Complete the given
- ⊞ Converting from A to I
- ⊞ Designing your classe
- ⊞ Java Data Structures
- ⊞ Passing the unit tests
- ⊞ What is wrong with th
- ⊞ Writing Code Home A
- ⊞ Written Test Core Jav
- ⊞ Written Test JEE (1)

How good are your .....?

open all | close all

- ⊞ Career Making Know-
- ⊞ Job Hunting & Resum

```

    public String toJson() {
        ObjectMapper mapper = new ObjectMapper(
        mapper.setSerializationInclusion(JsonIn

        try {
            String json = mapper.writeValueAsSt
            return json;
        } catch (JsonProcessingException e) {
            e.printStackTrace();
            throw new RuntimeException(e);
        }
    }

    public static ClientEvent fromJson(String j
        ObjectMapper mapper = new ObjectMapper(
        mapper.setSerializationInclusion(JsonIn
        try {
            ClientEvent clientEvent = mapper.re
            return clientEvent;
        } catch (IOException e) {
            e.printStackTrace();
            throw new RuntimeException(e);
        }
    }

}

/**
 * Custom LocalDate deserializer from JSON usin
 * format.
 */
class CustomLocalDateDeserializer extends JsonD

    public CustomLocalDateDeserializer() {
    }

    @Override
    public LocalDate deserialize(JsonParser jp,
            throws IOException, JsonProcessingE
        String stringValue = jp.getValueAsStrin
        LocalDate dateTime = LocalDate.parse(st
                DateTimeFormatter.ofPattern(Cli
        return dateTime;
    }
}

/**
 * Custom LocalDate serializer to JSON using th
 */
class CustomLocalDateSerializer extends JsonSer

    public CustomLocalDateSerializer() {
    }

    @Override
    public void serialize(LocalDate value, Json
            SerializerProvider provider) throws
            JsonGenerationException {
        DateTimeFormatter formatter = DateTimeF
                .ofPattern(ClientEvent.DATE_FOR
        jgen.writeString(formatter.format(value
    }
}

```

Step 5: Finally, the JUnit test class to Marshall & Unmarshall.

```
 package com.mytutorial.model;

import java.io.IOException;
import java.io.StringWriter;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

import org.junit.Assert;
import org.junit.Test;

import com.fasterxml.jackson.core.JsonGeneration
import com.fasterxml.jackson.databind.JsonMappin
import com.fasterxml.jackson.databind.ObjectMapp

public class ClientEventTest {

    private static final String JSON_TEXT = "{\"

    @Test
    public void testMarshall() throws JsonGenera
        ClientEvent ce = new ClientEvent();
        ce.setClientCode("CLIENT-123");
        ce.setEventDate(LocalDate.now());
        ObjectMapper objMapper = new ObjectMappe
        StringWriter sw = new StringWriter();
        objMapper.writeValue(sw, ce);

        Assert.assertNotNull(sw.toString());
        Assert.assertEquals(JSON_TEXT, sw.toStri
    }

    @Test
    public void testUnMarshall() throws JsonGene

        ClientEvent ce = ClientEvent.fromJson(JS

        Assert.assertNotNull(ce);
        Assert.assertEquals("CLIENT-123", ce.get
        DateTimeFormatter df = DateTimeFormatter
        Assert.assertEquals("20/10/2015", df.for
    }

}
```

Step 6: Run “ClientEventTest” as a Junit test.

Popular Posts

- ♦ 11 Spring boot interview questions & answers

  828 views

- ♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

  768 views

18 Java scenarios based interview Questions and Answers

400 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

389 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

296 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

286 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

280 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

240 views

001B: ♦ Java architecture & design concepts interview questions & answers

202 views

Bio **Latest Posts**

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ Preventing XSS attacks in Java

Spring, JavaConfig (i.e @Configuration) & JDBC by example ›

Posted in JSON**,** member-paid

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

↑