FAQ: How to read Dates with Hibernate

FAQ: Handling Dates with Hibernate

How can I read date strings from the command-line and store them as dates in the database?

Answer:

You can make use of a combination of Java's date formatting class and Hibernate annotations.

Sample output:

```
Student [id=50, firstName=Paul, lastName=Doe,
email=paul@luv2code.com, dateOfBirth=null]
Student [id=51, firstName=Daffy, lastName=Duck,
email=daffy@luv2code.com, dateOfBirth=null]
Student [id=52, firstName=Paul, lastName=Doe, email=paul@luv.com,
dateOfBirth=31/12/1998]
```

Development Process Overview

1. Alter database table for student
2. Add a date utils class for parsing and formatting dates
3. Add date field to Student class
4. Add toString method to Student class
5. Update CreateStudentDemo

Detailed steps

1. Alter database table for student

We need to alter the database table to add a new column for "date_of_birth".

Run the following SQL in your MySQL Workbench tool.

```
ALTER TABLE `hb_student_tracker`.`student`
ADD COLUMN `date_of_birth` DATETIME NULL AFTER `last_name`;
```

--

2. Add a date utils class for parsing and formatting dates

We need to add a DateUtils class to handle parsing and formatting dates. The source code is here. The class should be placed in the package: com.luv2code.hibernate.demo.

The date formatter uses special symbols for formatting/parsing.

- dd: day in month (number)
- MM: month in year (number)
- yyyy: year

See this link for details: https://docs.oracle.com/javase/tutorial/i18n/format/simpleDateFormat.html

```
package com.luv2code.hibernate.demo;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class DateUtils {

    // The date formatter
    // - dd:   day in month (number)
    // - MM:   month in year (number)
    // - yyyy: year
    //
    // See this link for details: https://docs.oracle.com/javase/tutorial/i18n/format/simpleDateFormat.html
    //
    //
    private static SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");

    // read a date string and parse/convert to a date
    public static Date parseDate(String dateStr) throws ParseException {
        Date theDate = formatter.parse(dateStr);

        return theDate;
    }

    // read a date and format/convert to a string
    public static String formatDate(Date theDate) {

        String result = null;

        if (theDate != null) {
            result = formatter.format(theDate);
        }

        return result;
    }
}
```

---

3. Add date field to Student class

We need to add a date field to the Student class. We map this field to the database column, "date_of_birth". Also, we make use of the @Temporal annotation. This is a Java annotation for storing dates.

```
    @Column(name="date_of_birth")
    @Temporal(TemporalType.DATE)
    private Date dateOfBirth;
```

Here's the full source code.

```
package com.luv2code.hibernate.demo.entity;

import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

import com.luv2code.hibernate.demo.DateUtils;

@Entity
@Table(name="student")
public class Student {

    @Id
    @Column(name="id")
    private int id;

    @Column(name="first_name")
    private String firstName;

    @Column(name="last_name")
    private String lastName;

    @Column(name="email")
    private String email;

    @Column(name="date_of_birth")
    @Temporal(TemporalType.DATE)
    private Date dateOfBirth;

    public Student() {

    }

    public Student( String firstName, String lastName, String email, Date
theDateOfBirth) {

        this.firstName = firstName;
        this.lastName = lastName;
        this.email = email;
        this.dateOfBirth = theDateOfBirth;

    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }
```

```
	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

	public String getEmail() {
		return email;
	}

	public void setEmail(String email) {
		this.email = email;
	}

	public Date getDateOfBirth() {
		return dateOfBirth;
	}

	public void setDateOfBirth(Date dateOfBirth) {
		this.dateOfBirth = dateOfBirth;
	}

	@Override
	public String toString() {
		return "Student [id=" + id + ", firstName=" + firstName + ", lastName=" + lastName + ", email=" + email
				+ ", dateOfBirth=" + DateUtils.formatDate(dateOfBirth) + "]";
	}

}
```

---

4. Add toString method to Student class

We will make an update to the toString method in our Student class. It will use the formatter from our DateUtils.class. This code is already included in Student.java from the previous step. I'm just highlighting it here for clarity.

```
        return "Student [id=" + id + ", firstName=" + firstName + ", lastName=" + lastName + ", email=" + email
                + ", dateOfBirth=" + DateUtils.formatDate(dateOfBirth) + "]";
```

Note the use of DateUtils above.

---

5. Update CreateStudentDemo

Now for the grand finale. In the main program, read the date as a String and parse/convert it to a date. Here's the snippet of code.

```
            String theDateOfBirthStr = "31/12/1998";
            Date theDateOfBirth = DateUtils.parseDate(theDateOfBirthStr);

            Student tempStudent = new Student("Pauly", "Doe", "paul@luv.com", theDateOfBirth);
```

Here's the full code:

```
package com.luv2code.hibernate.demo;

import java.text.ParseException;
import java.util.Date;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import com.luv2code.hibernate.demo.entity.Student;

public class CreateStudentDemo {

    public static void main(String[] args) {

        // create session factory
        SessionFactory factory = new Configuration().configure("hibernate.cfg.xml").addAnnotatedClass(Student.class)
                .buildSessionFactory();

        // create a session
        Session session = factory.getCurrentSession();

        try {
            // create a student object
            System.out.println("creating a new student object ...");

            String theDateOfBirthStr = "31/12/1998";

            Date theDateOfBirth = DateUtils.parseDate(theDateOfBirthStr);

            Student tempStudent = new Student("Pauly", "Doe", "paul@luv.com", theDateOfBirth);

            // start transaction
            session.beginTransaction();

            // save the student object
            System.out.println("Saving the student ...");
            session.save(tempStudent);

            // commit transaction
            session.getTransaction().commit();

            System.out.println("Success!");
        } catch (Exception exc) {
            exc.printStackTrace();
        } finally {
            factory.close();
        }
    }

}
```