Programming in Java

Topic: Date Time API

Contents…

- Introduction
- Local Date
- Local Time
- Local Date Time

Introduction

- New DateTime API is introduced in jdk8.
- LocalDate, LocalTime and LocalDateTime classes are provided in java.time package.

Java Date and Time API goals

- Classes and methods should be straight forward.
- The API should support fluent API approach.
- Instances of Date and Time objects should be immutable.
- Should be thread safe.
- Use ISO standard to define Date and Time.
- API should support strong type checks.
- Allows developers to extend API.

Working with Local Date and Time

- Java.time package provides two classes for working with local Date and Time.
- LocalDate
  - Does not include time
  - A year-month-day representation
  - toString – ISO 8601 format(YYYY-MM-DD)
- LocalTime
  - Does not include date
  - Stores hours:minutes:seconds:nanoseconds
  - toString- (HH:mm:ss.SSS)

LocalDate, LocalTime and LocalDateTime

- They are local in the sense that they represent date and time from the context of one observer, in contrast to time zones.
- All the core classes in the new API are constructed by factory methods.
- When constructing a value through its fields, the factory is called *of*.
- When converting from another type, the factory is called *from*.
- There are also parse methods that take strings as parameters.

LocalDate Class

LocalDate Class

- A date without a time-zone in the ISO-8601 calendar system, such as 2007-12-03.

- LocalDate is an immutable date-time object that represents a date, often viewed as year-month-day.

- Other date fields, such as day-of-year, day-of-week and week-of-year, can also be accessed.

- This class does not store or represent a time or time-zone so its portable across time zones.

Methods of LocalDate

- public static LocalDate now()
- public static LocalDate now(ZoneId zone)

- public static LocalDate of(int year, Month month, int dayOfMonth)

Note:DateTimeException can be thrown

- public static LocalDate of(int year, int month, int dayOfMonth)

Note: *DateTimeException can be thrown.*

- public static LocalDate parse(CharSequence text)

Note: *DateTimeParseException can be thrown.*

Example (now() method)

```
// Java program to demonstrate
// LocalDate.now() method

import java.time.*;
public class Test {
    public static void main(String[] args)
    {
      // create an LocalDate object
      LocalDate lt = LocalDate.now();
      // print result
      System.out.println("LocalDate : "+ lt);
    }
}
```

Example (now(ZoneId zone))

```
// Java program to demonstrate LocalDate.now() method
import java.time.*;
public class Test {
    public static void main(String[] args)
    {
      // create a clock
      ZoneId zid = ZoneId.of(“Asia/Kolkata");
      // create an LocalDate object using now(zoneId)
      LocalDate lt = LocalDate.now(zid);
      // print result
      System.out.println("LocalDate : "+ lt);
    }
}
```

Example (of() method)
public static LocalDate of(int year,int month,int dayOfMonth)

```
// Java program to demonstrate LocalDate.of(int month) method
import java.time.*;
public class Test {
    public static void main(String[] args)
    {
     // create LocalDate object
     LocalDate localdate = LocalDate.of(2020, 5, 13);
     // print full date
     System.out.println("Date: " + localdate);
    }
}
```

Output:

```
Date:2020-05-13
```

Example (of() method)

public static LocalDate of(int year,Month month,int dayOfMonth)

```
// Java program to demonstrate
// LocalDate.of(Month month) method
import java.time.*;
public class Test {
    public static void main(String[] args)
    {
     // create LocalDate object
     LocalDate localdate = LocalDate.of(2020, Month.MAY, 13);
     // print full date
     System.out.println("Date: "+ localdate);
    }
}
```

Output:

```
Date: 2020-05-13
```

Example:parse() method

```
import java.time.*;
public class Test {
    public static void main(String[] args)
    {
     // create an LocalDate object
     LocalDate lt = LocalDate.parse("2020-05-13");
     // print result
     System.out.println("LocalDate : "+ lt);
    }
}
```

Example

```
LocalDate ldt = LocalDate.now();

ldt = LocalDate.of(2015, Month.FEBRUARY, 28);

ldt = LocalDate.of(2015, 2, 13);

ldt = LocalDate.parse("2017-02-28”);
```

LocalTime Class

LocalTime Class

- A time without a time-zone in the ISO-8601 calendar system, such as 10:15:30. 13

- LocalTime is an immutable date-time object that represents a time, often viewed as hour-minute-second.

- Time is represented to nanosecond precision.
- For example, the value "13:45:30.123" can be stored in a LocalTime.

- This class does not store or represent a date or time-zone.

Methods of LocalTime

Methods

- public static LocalTime now()
- public static LocalTime now(ZoneId zone)

- public static LocalTime of(int hour, int minute)
- public static LocalTime of(int hour, int minute, int second)
- public static LocalTime of(int hour, int min, int sec, int nsec)

- public static LocalTime parse(CharSequence text)

Example(now() method)
public static LocalTime now()

```
// Java program to demonstrate LocalTime.now() method
import java.time.*;
public class Test {
    public static void main(String[] args)
    {
     // apply now() method
     // of LocalTime class
     LocalTime time = LocalTime.now();
     // print time
     System.out.println("Time: "+ time);
    }
}
```

Output: It varies as the time passes.

```
Time: 20:43:41.453
```

Example(now(ZoneId zone)

```
// Java program to demonstrate LocalTime.now() method
import java.time.*;
public class Test {
    public static void main(String[] args)
    {
           // create a clock
     ZoneId zid = ZoneId.of("Asia/Kolkata");
     LocalTime time = LocalTime.now();
     // print time
     System.out.println("Time: "+ time);
    }
}
```

Output:

```
Time: 06:30:45.936
```

Output may vary with the passage of time

Example(of()) public static LocalTime of(int hour,int minute)

```
// Java program to demonstrate LocalTime of(int hour, int minute) method
import java.time.*;
public class Main {
    public static void main(String[] args)
    {
      // Create LocalTime object
      LocalTime localtime = LocalTime.of(6, 5);
      // Print time
      System.out.println("TIME: "+ localtime);
    }
}
```

Output:

```
TIME: 06:05
```

Example:public static LocalTime of(int hour,int minute,int second)

```
// Java program to demonstrate LocalTime of(int hour, int minute, int second) method
import java.time.*;
public class Main {
    public static void main(String[] args)
    {
      // Create LocalTime object
      LocalTime localtime = LocalTime.of(6, 5, 40);
      // Print time
      System.out.println("TIME: "+ localtime);
    }
}
```

Output:

```
TIME: 06:05:40
```

Example(public static LocalTime of(int hour,int minute,int second,int nanosecond))

```
// Java program to demonstrate LocalTime of(int hour, int minute, int second, int nanosecond) method
import java.time.*;
public class Main {
    public static void main(String[] args)
    {
      // Create LocalTime object
      LocalTime localtime = LocalTime.of(6, 5, 40, 50);
      // Print time
      System.out.println("TIME: "+ localtime);
    }
}
```

Output:

```
TIME: 06:05:40.000000050
```

Example(public static LocalTime parse(CharSequence text))

```
// Java program to demonstrate LocalTime.parse() method
import java.time.*;
public class Main {
    public static void main(String[] args)
    {
     // create an LocalTime object
     LocalTime lt = LocalTime.parse("10:15:45");
     // print result
     System.out.println("LocalTime : "+ lt);
    }
}
```

Output:

```
LocalTime : 10:15:45
```

LocalDateTime Class

LocalDateTime Class

- A date-time without a time-zone in the ISO-8601 calendar system, such as 2007-12-03T10:15:30.

- LocalDateTime is an immutable date-time object that represents a date-time, often viewed as year-month-day-hour-minute-second.
- Other date and time fields, such as day-of-year, day-of-week and week-of-year, can also be accessed.

- Time is represented to nanosecond precision.
- For example, the value "2nd October 2007 at 13:45.30.123456789" can be stored in a LocalDateTime.

Methods of LocalDateTime

Methods

- public static LocalDateTime now()
- public static LocalDateTime now(ZoneId zone)

- public static LocalDateTime of(int year, int mnth, int day, int hour, int mint)
- public static LocalDateTime of(int year, int mnth, int day, int hour, int mint, int sec)
- public static LocalDateTime of(int year, int mnth, int day, int hour, int mint, int sec, int nsec)
- public static LocalDateTime of(LocalDate d, LocalTime t)
- public static LocalDateTime parse(CharSequence text)

Example—now()

```
// Java program to demonstrate LocalDateTime.now() method
import java.time.*;
public class Test {
    public static void main(String[] args)
    {
     // create an LocalDateTime object
     LocalDateTime lt = LocalDateTime.now();
     // print result
     System.out.println("LocalDateTime : "+ lt);
    }
}
```

Sample output:

```
LocalDateTime : 2021-02-19T10:03:55.356
```

Example—now()

```
// Java program to demonstrate LocalDateTime.now() method
import java.time.*;
public class Main {
    public static void main(String[] args)
    {
     // create a clock
     ZoneId zid = ZoneId.of(“Asia/Kolkata");
     // create an LocalDateTime object using now(zoneId)
     LocalDateTime lt   = LocalDateTime.now(zid);
     // print result
     System.out.println("LocalDateTime : "+ lt);
    }
}
```

Sample Output:

```
LocalDateTime : 2021-02-20T09:37:12.068
```

Example—of()

```
import java.time.*;
 public class Main {
     public static void main(String[] args)
     {
      // create LocalDateTime object
      LocalDateTime localdatetime1 = LocalDateTime.of(2020, 5, 13, 6, 30);
      // print full date and time
      System.out.println("DateTime: "+ localdatetime1); //DateTime: 2020-05-13T06:30
      LocalDateTime localdatetime2 = LocalDateTime.of(2020, 5, 13, 6, 30,45);
      // print full date and time
      System.out.println("DateTime: "+ localdatetime2); //DateTime: 2020-05-13T06:30:45
      // create LocalDateTime object
       LocalDateTime localdatetime3 = LocalDateTime.of(2020, 5, 13, 6, 30, 45, 20000);
       // print full date and time
       System.out.println("DateTime: "+ localdatetime3); //DateTime: 2020-05-
     13T06:30:45.000020
     }
}
```

Example-of()

```
// Java program to demonstrate LocalDateTime.of(LocalDate date, LocalTime time)
    method
import java.time.*;
public class Main {
    public static void main(String[] args)
    {
     // Create LocalDate object using LocalDate.of() method
     LocalDate date = LocalDate.of(2020, 5, 13);
     // Create LocalTime object using LocalTime.of() method
     LocalTime time = LocalTime.of(6, 30);
     // Create LocalDateTime object
     LocalDateTime localdatetime = LocalDateTime.of(date, time);
     // Print full date and time
     System.out.println( "DateTime: " + localdatetime); //DateTime: 2020-05-13T06:30
    }
}
```

Example-parse()

```
// Java program to demonstrate LocalDateTime.parse() method
import java.time.*;
public class Main {
    public static void main(String[] args)
    {
    // create an LocalDateTime object
     LocalDateTime lt = LocalDateTime .parse("2018-12-30T19:34:50.63");
     // print result
     System.out.println("LocalDateTime : "+ lt);
    }
}
```

Output:

```
LocalDateTime : 2018-12-30T19:34:50.630
```