

Property Name	Purpose
<code>ThankGivingDay</code>	Returns the Date of Thanks Giving Day of the current year.
<code>BlackFriday</code>	Returns the Date of Black Friday of the current year.
<code>WomenDay</code>	Returns the Date of Women's of the current year.
<code>IndependenceDay</code>	Returns 4 th July of current year.
<code>NewYearsDay</code>	Returns 1 st January of current year.
<code>MartinLutherKingDay</code>	Returns the Date of Martin Luther King Day of the current year.
<code>WashingtonsBirthDay</code>	Returns the Date of Washington's Birthday of the current year.
<code>MemorialDay</code>	Returns the Date of Memorial Day of the current year.
<code>LaborDay</code>	Returns the Date of Labor Day of the current year.
<code>ColumbusDay</code>	Returns the Date of Columbus Day of the current year.
<code>RemembranceDay</code>	Returns the Date of Remembrance Day of the current year.

FirstQuarterStartDate

Sometimes the year is divided into four segments popularly known as quarters. Moreover in different organizations and institutes these quarters are different. This property is added to mark the start date of the first quarter of the year. (This may be different from the calendar start as 1st January)

WeekStartDate

Some organizations work in different time frame than others. For some of them Sunday could be the first date of Working Week. This property allows the programmer to set a day name as the week start day. For example "Monday".

Method Name	Brief Introduction
<code>tomorrow</code>	It returns the date of tomorrow of the passed argument date.
<code>yesterday</code>	It returns the date of yesterday of the passed argument date
<code>nextNthXday</code>	It returns the date of next Nth X day. Where X stands for days of the week (Sun, Mon, Tue etc). So this method is helpful when you need to find out something like "the date of 4 th Monday from Thanks Giving Day of this year."
<code>prevNthXday</code>	It returns the date of previous Nth X day. Where X stands for days of the week (Sun, Mon, Tue etc). So this method is helpful when you need to find out something like "the date of 4 th Monday before Thanks Giving Day of this year."
<code>nextSunday</code>	This is just a wrapper function of <code>nextNthXday</code> and it returns the date of the next Sunday starting from the argument date.
<code>nextMonday</code>	This is just a wrapper function of <code>nextNthXday</code> and it returns the date of the next Monday starting from the argument date.
<code>nextTuesday</code>	This is just a wrapper function of

	<p>nextNthXday and it returns the date of the next Tuesday starting from the argument date.</p>
<code>nextWednesday</code>	<p>This is just a wrapper function of nextNthXday and it returns the date of the next Wednesday starting from the argument date.</p>
<code>nextThursday</code>	<p>This is just a wrapper function of nextNthXday and it returns the date of the next Thursday starting from the argument date.</p>
<code>nextFriday</code>	<p>This is just a wrapper function of nextNthXday and it returns the date of the next Friday starting from the argument date.</p>
<code>nextSaturday</code>	<p>This is just a wrapper function of nextNthXday and it returns the date of the next Saturday starting from the argument date.</p>
<code>lastSunday</code>	<p>This is just a wrapper function of prevNthXday and it returns the date of the last Sunday just before the argument date.</p>
<code>lastMonday</code>	<p>This is just a wrapper function of prevNthXday and it returns the date of the last Monday just before the argument date.</p>
<code>lastTuesday</code>	<p>This is just a wrapper function of prevNthXday and it returns the date of the last Tuesday just before the argument date.</p>

<code>lastWednesday</code>	This is just a wrapper function of <code>prevNthXday</code> and it returns the date of the last Sunday just before the argument date.
<code>lastThursday</code>	This is just a wrapper function of <code>prevNthXday</code> and it returns the date of the last Thursday just before the argument date
<code>lastFriday</code>	This is just a wrapper function of <code>prevNthXday</code> and it returns the date of the last Friday just before the argument date
<code>lastSaturday</code>	This is just a wrapper function of <code>prevNthXday</code> and it returns the date of the last Saturday just before the argument date
<code>lastday</code>	This method returns the last date of the month of which the argument is one date. For example if the argument date passed to <code>lastday</code> is February 2,2008 then it will return 29-February-2008 as 2008 is a leap year.
<code>getWorkingDaysInRange</code>	This method returns a <code>List<></code> of <code>DateTime</code> objects that represents the working days in the given range. To specify the range this function takes two dates as start and end date, a list of holidays as a <code>List<></code> of <code>DateTime</code> objects and an integer that marks the weekend holiday style.

getWorkingDaysFor

This method is a wrapper which calls the above function to get the Working Days for a particular year. It takes three arguments. The year for which the list of working days needs to be extracted, a list of holidays as a list of date time objects and an integer marking the weekend holiday style. If only Sunday is holiday then this integer is 1, if both, Saturday and Sunday are holidays then this is 2 and otherwise this integer is 0. These two methods assume that the working days are Monday to Friday, Monday to Saturday or all days of week excluding the specified holidays. Any special shift holidays are out of the scope for this version of DateDef.

getFirstXDayOf

This function returns the date of the first X day of the year specified as argument. For example, if X is Sunday then it will return the first Sunday of the argument year.

getLastXDayOf

This function returns the date of the last X day of the year specified as argument. For example, if X is Sunday then it will return the last Sunday of the argument year.

ThanksGivingDayOf

This method returns the date of Thanks Giving Day of the specified year.

BlackFridayOf	This method returns the date of the Black Friday of the specified year.
MemorialDayOf	This method returns the date of the Memorial Day of the specified year.
WomenDayOf	This method returns the date of the Women Day of the specified year.
NewYearsDayOf	This method returns the date of the new year's day of the specified year.
IndependenceDayOf	This method returns the Independence Day of the specified year.
MartinLutherKingDayOf	This method returns the Martin Luther King Day Of the specified Year.
WashingtonsBirthDayOf	This method returns Washington's Birthday of the specified year.
LaborDayOf	This method returns the Labor Day of the specified year.
ColombusDayOf	This method returns the Columbus Day of the specified year.
RemembranceDayOf	This method returns the Remembrance Day of the specified year.
GetFirstQuarterDays	This method returns the list of first quarter days of the financial year starting from the FirstQuarterStartDate

<code>GetSecondQuarterDays</code>	This method returns the list of second quarter days of the financial year.
<code>GetThirdQuarterDays</code>	This method returns the list of third quarter days of the financial year.
<code>GetFourthQuarterDays</code>	This method returns the list of fourth quarter days of the financial year.
<code>QuarterTodayIsIn</code>	This method returns an integer which represents which quarter Today belongs to.
<code>QuarterOfThisDate</code>	This method returns an integer representing the quarter in which the argument date belongs.

METHOD

```
public DateTime tomorrow(DateTime d)
```

DESCRIPTION

This is a wrapper function which returns the day which is tomorrow of the passed date. For example if it is called with 31st December of any year it will return the 1st day of January of the next year. Leap years are handled.

EXAMPLE CODE SNIPPET

```
//The following code will return 29th February as 2008 is a leap year  
DateTime d = tomorrow(new DateTime(2008,2,28));
```

METHOD

```
public DateTime yesterday(DateTime d)
```

DESCRIPTION

This is a wrapper function which returns the day which is a day older than the passed date. For example if it is called with 1st day of January of a year then it will return the 31st day of December of the previous year.

EXAMPLE CODE SNIPPET

```
DateTime d = yesterday(new DateTime(2008,3,1));
```

METHOD

```
public DateTime nextNthXday(DateTime start,  
  
    int n,  
  
    string downname)
```

DESCRIPTION

Sometimes we need to calculate what will be the date of the third Sunday from now or what will be the fourth Thursday in November. In such situations this function will be useful. The first parameter is the start date, the second one is the number of days we are looking for of type downname. For example if we are looking for 4th Friday from a date represented by start, then the call to this function will be

EXAMPLE CODE SNIPPET

```
//This will return 10th Sunday of the year 2008 or in other words 10th  
//Sunday starting from 1st January 2008.
```

```
DateTime d = new DateTime(2008,1,1);  
DateTime t = nextNthXday(d,10,"Sunday");
```

METHOD

```
public DateTime prevNthXday(DateTime start, int n,  
  
    string downname)
```

DESCRIPTION

Sometimes we need to calculate what the date of the third Sunday from now was or what the date of fourth Thursday in November was. In such situations this function will be useful. The first parameter is the start date, the second one is the number of days we are looking for of type downame.

EXAMPLE CODE SNIPPET

```
//This will return last 2nd Sunday from the date 1st January 2008  
DateTime start = new DateTime(2008,1,1);  
DateTime last = prevNthXday(start,2,"Sunday");
```

METHOD

```
public DateTime lastday(DateTime start)
```

DESCRIPTION

This function returns the date of the month of which start is a date.

EXAMPLE CODE SNIPPET

```
//This will return 29th February as 2008 is leap year.  
DateTime d = new DateTime(2008,2,28);  
DateTime t = lastday(d);
```

METHOD

```
public List<DateTime> getXDaysFor(int month,  
                                  int year,string downame)
```

DESCRIPTION

This function returns the X days of a month of a year. Month is the month and year is the year. downame is the name of the weekday.

EXAMPLE CODE SNIPPET

```
//This will return all the Fridays of January 2008  
List<DateTime> list = getXDaysFor(1,2008,"Friday");
```

METHOD

```
public List<DateTime> getWorkingDaysFor (int year,  
                                         List<DateTime> holidays, int weekendStyle)
```

DESCRIPTION

This function returns all the working days for the year “year” and where the holidays of the year are listed as “holidays” and the weekend holiday style is denoted by the last argument weekendStyle. If there is only one holiday (which is Sunday) then weekendStyle is 1. If both Saturday and Sunday are holidays then the weekendStyle is 2. But if none is a holiday and 7 days are working except the holidays then weekendStyle is 0.

EXAMPLE CODE SNIPPET

```
//This will return all the working days of 2008 where holidays is a  
//list of DateTime which contains the public holidays.  
  
List<DateTime> list = getWorkingDaysFor(2008,holidays,2);
```

METHOD

```
public List<DateTime> getWorkingDaysInRange(DateTime start,  
                                             DateTime end,List<DateTime> holidays,int weekendStyle)
```

DESCRIPTION

This function returns all the working days within a given range and where the holidays of the year are listed as “holidays” and the weekend holiday style is denoted by the last argument weekendStyle. If there is only one holiday (which is Sunday) then weekendStyle is 1. If both Saturday and Sunday are holidays then the weekendStyle is 2. But if none is a holiday and 7 days are working except the holidays then weekendStyle is 0.

EXAMPLE CODE SNIPPET

```
//This will return all the working days between 1-Jan-2008 and  
//1-Apr-2008 where holidays is a list of DateTime which contains the  
//public holidays.  
  
DateTime d1 = new DateTime(2008,1,1);
```

```
DateTime d2 = new DateTime(2008,4,1);  
List<DateTime> list = getWorkingDaysInRange(d1,d2,holidays,2);
```

METHOD

```
public DateTime getLastXdayOf(int year,string x)
```

DESCRIPTION

This function returns the last X day of the year passed. For example if we pass 2008 and "Sunday" then the method will return the date of the last Sunday of 2008.

EXAMPLE CODE SNIPPET

```
//This will return the last Thursday of 2008  
DateTime d = getLastXdayOf(2008,"Thursday");
```

The other method are kind of self explanatory of a wrapper of the discussed methods. If you have any trouble understanding the usages of any of these properties or methods, please let us know by mailing at api@consulttoday.com

