

توسعه برنامه های

موبایل

جلسه نهم مجازی

بخش اول

سحر صادقی

به دست آوردن ساعت و تاریخ فعلی در اندروید

روش های مختلفی برای به دست آوردن تاریخ و زمان (Time & Date) وجود دارد یکی از آنها استفاده از کلاس Time برای به دست آوردن ساعت است.

```
1 Time time = new Time();
2 time.setToNow();
3 Log.d("TIME TEST", Long.toString(time.toMillis(false)));
```

اگر بخواهید تاریخ و زمان را به درست آورید می توانید از کلاس SimpleDateFormat استفاده کنید.

```
1 SimpleDateFormat sdf = new SimpleDateFormat("yyyyMMdd_HHmms");
2 String currentDateandTime = sdf.format(new Date());
```

در بالا yyyy به معنی سال MM به معنی ماه dd به معنی روز و HH به معنی ساعت در فرمت ۲۴ ساعت (اگر ۱۲ ساعت می خواهید تنظیم شود مقدار hh را جایگزین کنید.) mm ماه و ss به معنی ثانیه می باشد.

یک مثال دیگر با استفاده از کلاس Calendar و DateFormat

```
1 DateFormat df = new SimpleDateFormat("EEE, d MMM yyyy, HH:mm");
2 String date = df.format(Calendar.getInstance().getTime());
```

در بالا کارکترهای مختلفی قرار داده شده است می توانید از sample که در ادامه قرار داردمی توانید آنها را متوجه شوید.

```
1 "yyyy.MM.dd G 'at' HH:mm:ss z" ---- 2001.07.04 AD at 12:08:56 PDT
2 "hh 'o'clock' a, zzzz" ----- 12 o'clock PM, Pacific Daylight Time
3 "EEE, d MMM yyyy HH:mm:ss Z"----- Wed, 4 Jul 2001 12:08:56 -0700
4 "yyyy-MM-dd'T'HH:mm:ss.SSSZ"----- 2001-07-04T12:08:56.235-0700
5 "yyMMddHHmmsZ"----- 010704120856-0700
6 "K:mm a, z" ----- 0:08 PM, PDT
7 "h:mm a" ----- 12:08 PM
8 "EEE, MMM d, 'yy" ----- Wed, Jul 4, '01
```

سمت چپ هر کارکتر قرار داده شده است و result آن نیز روبه روی آن قرار گرفته است.

به درست آوردن زمانی و تاریخ فعلی با استفاده از کلاس Time و دسترسی به ماه و سال و روز و ساعت

```
1 Time today = new Time(Time.getCurrentTimezone());
2 today.setToNow();
3
4 String day = (today.monthDay + ""); // Day of the month (1-31)
5 String month = (today.month + ""); // Month (0-11)
6 String year = (today.year + ""); // Year
7 String time = (today.format("%k:%M:%S")); // Current time
```

ساده ترین روش برای به دست آوردن ساعت و تاریخ کد بالاست البته این کلاس دارای نقص هایی نیز هست به همین علت استفاده از آن را توصیه نمی کنند.

مثال دیگر برای SimpleDateFormat

```
SimpleDateFormat databaseDateTimeFormate = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
SimpleDateFormat databaseDateFormat = new SimpleDateFormat("yyyy-MM-dd");
1 SimpleDateFormat sdf1 = new SimpleDateFormat("dd.MM.yy");
2 SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy.MM.dd G 'at' hh:mm:ss z");
3 SimpleDateFormat sdf3 = new SimpleDateFormat("EEE, MMM d, 'yy");
4 SimpleDateFormat sdf4 = new SimpleDateFormat("h:mm a");
5 SimpleDateFormat sdf5 = new SimpleDateFormat("h:mm");
6 SimpleDateFormat sdf6 = new SimpleDateFormat("H:mm:ss:SSS");
7 SimpleDateFormat sdf7 = new SimpleDateFormat("K:mm a,z");
8 SimpleDateFormat sdf8 = new SimpleDateFormat("yyyy.MMMMM.dd GGG hh:mm
9 aaa");
10
11
12 String currentDateandTime = databaseDateTimeFormate.format(new
13 Date()); //2009-06-30 08:29:36
14 String currentDateandTime = databaseDateFormat.format(new Date()); //2009-
15 06-30
16 String currentDateandTime = sdf1.format(new Date()); //30.06.09
17 String currentDateandTime = sdf2.format(new Date()); //2009.06.30 AD at
18 08:29:36 PDT
19 String currentDateandTime = sdf3.format(new Date()); //Tue, Jun 30, '09
20 String currentDateandTime = sdf4.format(new Date()); //8:29 PM
21 String currentDateandTime = sdf5.format(new Date()); //8:29
22 String currentDateandTime = sdf6.format(new Date()); //8:28:36:249
String currentDateandTime = sdf7.format(new Date()); //8:29 AM,PDT
String currentDateandTime = sdf8.format(new Date()); //2009.June.30 AD 08:29
AM
```

به دست آوردن تاریخ با استفاده از کلاس Calendar

```
1 final Calendar c = Calendar.getInstance();
2 int mYear = c.get(Calendar.YEAR);
3 int mMonth = c.get(Calendar.MONTH);
4 int mDay = c.get(Calendar.DAY_OF_MONTH);
```

آموزش تایمر در اندروید

این آموزش هر چند دیده نمی شود ولی کارایی بسیار بالایی دارد. خب برویم به سراغ بخش کد برنامه شیرین ترین بخش! ²

```
1 Timer timerObj = new Timer();
2 TimerTask timerTaskObj = new TimerTask() {
3     public void run() {
4         //perform your action here
5     }
6 };
7 timerObj.schedule(timerTaskObj, 0, 15000);
```

ابتدا یک object از تایمر ایجاد کرده سپس باید یک TimerTask برای زمان بندی تایمر ایجاد شده ایجاد کنیم بعد از آن یک void ایجاد کرده سپس در این void که از نام آن run است را اجرا می کنیم و در آخر بعد از بسته شدن void زمان زمانی بندی کردن تایمر ایجاد شده می کنیم برای این کار از ویژگی Schedule (به معنی زمان بندی) را فراخوانی کرده و در آخر "زمان" را برای تایمر ست می کنیم تایمر را باید به میلی ثانیه بدهیم به طور مثال یک ثانیه برابر است با ۱۰۰۰ میلی ثانیه.

شاید بخواهید یک تایمر تکرار ایجاد کنید برای این کار می توانیم از کد زیر استفاده کنیم.

```
1 Timer timer = new Timer();
2 TimerTask t = new TimerTask() {
3     int sec = 0;
4     @Override
5     public void run() {
6     }
7 };
8 timer.scheduleAtFixedRate(t, 1000, 1000);
```

کد بالا نیز شبیه کد قبلی است با این تفاوت که این تایمر ۱۰۰۰ بار اجرا می شود یعنی بعد از هر یک ثانیه تا ۱۰۰۰ بار اجرا می شود.

برویم به سراغ شمارنده (counter) معکوس به کد زیر نگاه کنید.

```
1 new CountdownTimer(30000, 1000) {
2
3     public void onTick(long millisUntilFinished) {
```

```

4     mTextField.setText("Seconds remaining: " + millisUntilFinished / 1000);
5 }
6
7 public void onFinish() {
8     mTextField.setText("Done");
9 }
10
11 }.start();

```

از کد بالا برای ایجاد تایمر معکوس استفاده کنیم و کنترلی بیشتری نسبت به تایمر های قبلی داریم به این صورت که در صورت تموم شدن این تایمر پیغام Done نمایش داده می شود.

این بخش مخصوص تایمر است ولی گفتم شاید بعضی از دوستان به کد زیر نیاز داشته باشند کد زیر به کار به صورت پشت سر هم بعد از یک فاصله زمانی اجرا کند. به طور مثال می خواهید اینترنت را هر ۱۵ ثانیه یک بار چک کنید برای این کار می توانید از کد زیر استفاده کنید.

```

1 final Handler handler = new Handler();
2 handler.postDelayed(new Runnable() {
3     @Override
4     public void run() {
5         //Do something after 100ms
6     }
7 }, 1000);

```

کد بالا بعد از فاصله زمانی یک ثانیه ای یک دکد را اجرا می کند ولی استفاده بیش از حد از کد بالا سبب memory leak می شود.

استفاده از کد بالا به صورت ساده شده مانند زیر

```

1 new Handler().postDelayed(new Runnable() {
2     @Override
3     public void run() {
4         //Do something after 100ms
5     }
6 }, 1000);

```

کدی قوی تر از کد های بالا کد زیر از api ورژن ۱ تا ۲۴ کار می کند و کد قدرتمندی است.

```

1 new Handler(Looper.getMainLooper()).postDelayed(new Runnable() {
2     @Override
3     public void run() {
4         //Do something here

```

```
5 }  
6 }, 5000);
```

کد بالا بعد هر ۵ ثانیه یک کار را انجام می دهد.

در صورت استفاده از کد بالا ممکن است به ارور بر بخورید زیرا کدهای بالا در یک ترد دیگر است به طور مثال اگر بخواهید مقدار متن یک دکمه را تغییر دهید باید از `RunOnUiThread` استفاده کنید همانند زیر

```
1 runOnUiThread(new Runnable() {  
2     @Override  
3     public void run() {  
4         new Handler().postDelayed(new Runnable() {  
5             @Override  
6             public void run() {  
7                 //Do something after 1 second  
8             }  
9         }, 1000);  
10    }  
11 });
```

کدهای مختلفی از این دست برای تایمر وجود دارد همانند زیر

```
1 new Timer().schedule(new TimerTask() {  
2     @Override  
3     public void run() {  
4         // this code will be executed after 2 seconds  
5     }  
6 }, 2000);
```