获取当前城市并更新：

由于获取当前城市需要定位，因此我们必须在 Manifest.xml 里面添加 ACCESS_FINE_LOCATION 或者 ACCESS_COARSE_LOCATION 这两个权限之一：

```xml
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
```

接下来我们编写 LocationUtil 类，用于获取当前所在地的名称

```
public class LocationUtil {
    private LocationManager locationManager;
    private Context context;

    private LocationUtil(Context context) {
        this.context = context;
        locationManager = (LocationManager) context.getSystemService(Context.LOCATION_SERVICE);
    }

    /**
     * 用于获取当前所在地的城市名称
     *
     * @return cityName
     */
    public String getCurrentLocation() {
        // 有些用户的手机默认应用关闭定位权限，这条判断语句用于判断这种情形
        // 在用户没有开启定位权限的时候提示用户开启该权限
        if (ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
                && ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(context, text: "请允许赋予程序定位权限", Toast.LENGTH_LONG).show();
            return null;
        }

        // 使用网络提供商所提供的经纬度地址
        Location currentLocation = locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
        if (currentLocation == null) {
            Toast.makeText(context, text: "无法获取当前位置", Toast.LENGTH_SHORT).show();
            return null;
        }

        // 根据经纬度来获取城市名称
        return getCityByLocation(currentLocation);
    }
```

根据经纬度来获取当前城市名称的代码如下：

```
private String getCityByLocation(Location location) {
    // Android 自带的位置解析包，可以根据经纬度来解析地理位置
    Geocoder geocoder = new Geocoder(context, Locale.getDefault());

    try {
        List<Address> addresses = geocoder.getFromLocation(location.getLatitude(),
                location.getLongitude(), maxResults: 1);
        if (addresses.size() > 0) {     // 如果找到了城市信息
            Address address = addresses.get(0);
            String city = address.getLocality();
            // 获取的城市名称带有“市县乡州村”，使用正则式除掉
            city = city.replaceAll( regex: "([市县乡州村])", replacement: "");
            Toast.makeText(context, String.format("当前位置 %s", city), Toast.LENGTH_SHORT).show();
            return city;
        }
    } catch (IOException e) {
        e.printStackTrace();
    }

    return null;
}
```

之后为定位按键添加监听，用于更新当前城市天气：

```
locationBtn.setOnClickListener((v) -> {
        //  在获取信息的时候先要上锁，避免重复获取
        if (failedToSetCannotRefreshNow()) return;

        LocationUtil locationUtil = LocationUtil.getInstance(ShowWeatherActivity.this);
        String city = locationUtil.getCurrentLocation();
        if (city == null) {
            return;
        }

        WeatherPredicationApp app = (WeatherPredicationApp) getApplication();
        List<CityEntity> cityEntities = app.getCityList();

        //  通过遍历城市列表，找到当前城市的cityCode，并刷新=
        for (CityEntity cityEntity : cityEntities) {
            if (cityEntity.getCityName().equals(city)) {
                String cityCode = cityEntity.getCityCode();
                setCurrentCity(cityCode);
                refreshWeatherByCityCode(cityCode);
            }
        }
});
```

获取定位信息之前要上锁，上锁代码如下：

```
private boolean failedToSetCannotRefreshNow() {
    // 使用synchronized关键字处理线程并发
    // isRefreshing 定义如下： private volatile boolean isRefreshing = false;
    synchronized (ShowWeatherActivity.class) {
        if (isRefreshing) {
            return true;
        }
        isRefreshing = true;
    }

    // 如果上锁成功，那么设置一个timer，防止获取信息的时候卡死，锁无法解除的情况
    refreshTimer = new Timer();
    refreshTimer.schedule(() -> {
            setCanRefreshNow(); // 解除锁
            Toast.makeText( context: ShowWeatherActivity.this,  text: "天气获取失败", Toast.LENGTH_SHORT).show();
    },  delay: 5000);

    // 让更新图标旋转
    refreshBtn.startAnimation(rotate);
    return false;
}
```

有上锁就对应有解除锁，解除锁的代码如下，解除锁的操作防止数据更新之后即可。

```
private void setCanRefreshNow() {
    synchronized (ShowWeatherActivity.class) {
        isRefreshing = false;

        // 将计时的 timer 关闭，由于timer取消后不能够再次使用，所以手动置为null，释放资源，防止MemoryLeak
        if (refreshTimer != null) {
            refreshTimer.cancel();
            refreshTimer = null;
        }

        // 停止旋转
        refreshBtn.clearAnimation();
    }
}
```