



# 任务三 显示天气

主讲教师：张晓诺

## 将天气显示到界面上 (1)

- (1) 在Utility类中添加一个解析天气的json数据的方法，如下所示：

- ```
/**
 * 将返回的JSON数据解析成Weather实体类
 */
public static Weather handleWeatherResponse(String response) {
    try {
        JSONObject jsonObject = new JSONObject(response);
        JSONArray jsonArray = jsonObject.getJSONArray("HeWeather");
        String weatherContent = jsonArray.getJSONObject(0).toString();
        return new Gson().fromJson(weatherContent, Weather.class);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}
```

- 本方法中先是通过JSONObject和JSONArray将天气数据中的主体内容解析出来，然后由于之前已经按照上面的数据格式定义过相应的GSON实体类，因此只需要通过fromJson () 方法就能直接将JSON数据转换成Weather对象。

(2) 在活动中去请求天气数据，以及将数据展示到界面上，修改WeatherActivity中的代码，如下所示：

- ```
package com.coolweather.android;
public class WeatherActivity extends AppCompatActivity {
    public DrawerLayout drawerLayout;
    public SwipeRefreshLayout swipeRefresh;
    private ScrollView weatherLayout;
    private Button navButton;
    private TextView titleCity;
    private TextView titleUpdateTime;
    private TextView degreeText;
    private TextView weatherInfoText;
    private LinearLayout forecastLayout;
    private TextView aqiText;
    private TextView pm25Text;
    private TextView comfortText;
    private TextView carWashText;
    private TextView sportText;
    private ImageView bingPicImg;
    private String mWeatherId;
    @Override
```

- protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    if (Build.VERSION.SDK\_INT >= 21) {  
        View decorView = getWindow().getDecorView();  
        decorView.setSystemUiVisibility(View.SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN  
View.SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE);  
        getWindow().setStatusBarColor(Color.TRANSPARENT);  
    }  
    setContentView(R.layout.activity\_weather);  
    // 初始化各控件  
    bingPicImg = (ImageView) findViewById(R.id.bing\_pic\_img);  
    weatherLayout = (ScrollView) findViewById(R.id.weather\_layout);  
    titleCity = (TextView) findViewById(R.id.title\_city);  
    titleUpdateTime = (TextView) findViewById(R.id.title\_update\_time);  
    degreeText = (TextView) findViewById(R.id.degree\_text);  
    weatherInfoText = (TextView) findViewById(R.id.weather\_info\_text);  
    forecastLayout = (LinearLayout) findViewById(R.id.forecast\_layout);  
    aqiText = (TextView) findViewById(R.id.aqi\_text);  
    pm25Text = (TextView) findViewById(R.id.pm25\_text);  
    comfortText = (TextView) findViewById(R.id.comfort\_text);  
    carWashText = (TextView) findViewById(R.id.car\_wash\_text);  
    sportText = (TextView) findViewById(R.id.sport\_text);  
    swipeRefresh = (SwipeRefreshLayout) findViewById(R.id.swipe\_refresh);  
    swipeRefresh.setColorSchemeResources(R.color.colorPrimary);  
    drawerLayout = (DrawerLayout) findViewById(R.id.drawer\_layout);  
    navButton = (Button) findViewById(R.id.nav\_button);  
}

- ```
SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(this);
String weatherString = prefs.getString("weather", null);
if (weatherString != null) {
    // 有缓存时直接解析天气数据
    Weather weather = Utility.handleWeatherResponse(weatherString);
    mWeatherId = weather.basic.weatherId;
    showWeatherInfo(weather);
} else {
    // 无缓存时去服务器查询天气
    mWeatherId = getIntent().getStringExtra("weather_id");
    weatherLayout.setVisibility(View.INVISIBLE);
    requestWeather(mWeatherId);
}
swipeRefresh.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
    @Override
    public void onRefresh() {
        requestWeather(mWeatherId);
    }
});
```

- ```
navButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        drawerLayout.openDrawer(GravityCompat.START);
    }
});
String bingPic = prefs.getString("bing_pic", null);
if (bingPic != null) {
    Glide.with(this).load(bingPic).into(bingPicImg);
} else {
    loadBingPic();
}
}
```

- ```
/**
 * 根据天气id请求城市天气信息。
 */
public void requestWeather(final String weatherId) {
    String weatherUrl = "http://guolin.tech/api/weather?cityid=" + weatherId +
"&key=bco418b57b2d4918819d3974ac1285d9";
    HttpUtil.sendOkHttpRequest(weatherUrl, new Callback() {
        @Override
        public void onResponse(Call call, Response response) throws IOException {
            final String responseText = response.body().string();
            final Weather weather = Utility.handleWeatherResponse(responseText);
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    if (weather != null && "ok".equals(weather.status)) {
                        SharedPreferences.Editor editor =
PreferenceManager.getDefaultSharedPreferences(WeatherActivity.this).edit();
                        editor.putString("weather", responseText);
                        editor.apply();
                        mWeatherId = weather.basic.weatherId;
                        showWeatherInfo(weather);
                    } else {
                        Toast.makeText(WeatherActivity.this, "获取天气信息失败", Toast.LENGTH_SHORT).show();
                    }
                }
            });
            swipeRefresh.setRefreshing(false);
        }
    });
}
```



- `@Override`  
    `public void onFailure(Call call, IOException e) {`  
        `e.printStackTrace();`  
        `runOnUiThread(new Runnable() {`  
            `@Override`  
            `public void run() {`  
                `Toast.makeText(WeatherActivity.this, "获取天气信息失败",`  
                `Toast.LENGTH_SHORT).show();`  
                `swipeRefresh.setRefreshing(false);`  
            `}`  
        `});`  
    `}`  
};  
`loadBingPic();`  
}  
/\*\*

- \* 加载必应每日一图

```
*/  
private void loadBingPic() {  
    String requestBingPic = "http://guolin.tech/api/bing_pic";  
    HttpUtil.sendOkHttpRequest(requestBingPic, new Callback() {  
        @Override  
        public void onResponse(Call call, Response response) throws IOException {  
            final String bingPic = response.body().string();  
            SharedPreferences.Editor editor =  
PreferenceManager.getDefaultSharedPreferences(WeatherActivity.this).edit();  
            editor.putString("bing_pic", bingPic);  
            editor.apply();  
            runOnUiThread(new Runnable() {  
                @Override  
                public void run() {  
                    Glide.with(WeatherActivity.this).load(bingPic).into(bingPicImg);  
                }  
            });  
        }  
        @Override  
        public void onFailure(Call call, IOException e) {  
            e.printStackTrace();  
        }  
    });  
}
```

- ```
/**
 * 处理并展示Weather实体类中的数据。
 */
private void showWeatherInfo(Weather weather) {
    String cityName = weather.basic.cityName;
    String updateTime = weather.basic.update.updateTime.split(" ")[1];
    String degree = weather.now.temperature + "°C";
    String weatherInfo = weather.now.more.info;
    titleCity.setText(cityName);
    titleUpdateTime.setText(updateTime);
    degreeText.setText(degree);
    weatherInfoText.setText(weatherInfo);
    forecastLayout.removeAllViews();
    for (Forecast forecast : weather.forecastList) {
        View view = LayoutInflater.from(this).inflate(R.layout.forecast_item, forecastLayout, false);
        TextView dateText = (TextView) view.findViewById(R.id.date_text);
        TextView infoText = (TextView) view.findViewById(R.id.info_text);
        TextView maxText = (TextView) view.findViewById(R.id.max_text);
        TextView minText = (TextView) view.findViewById(R.id.min_text);
        dateText.setText(forecast.date);
        infoText.setText(forecast.more.info);
        maxText.setText(forecast.temperature.max);
        minText.setText(forecast.temperature.min);
        forecastLayout.addView(view);
    }
}
```

- ```
if (weather.aqi != null) {
    aqiText.setText(weather.aqi.city.aqi);
    pm25Text.setText(weather.aqi.city.pm25);
}
String comfort = "舒适度: " + weather.suggestion.comfort.info;
String carWash = "洗车指数: " + weather.suggestion.carWash.info;
String sport = "运行建议: " + weather.suggestion.sport.info;
comfortText.setText(comfort);
carWashText.setText(carWash);
sportText.setText(sport);
weatherLayout.setVisibility(View.VISIBLE);
Intent intent = new Intent(this, AutoUpdateService.class);
startService(intent);
}
}
```

- 在onCreate () 方法中仍然先是获取一些控件的实例，然后会尝试从本地缓存中读取天气数据，如果是第一次是没有缓存，因此就会从网上取出天气id，并调用requestWeather () 方法来从服务器请求天气数据。注意请求数据的时候先将ScrollView进行隐藏，不然空数据的界面看上去会很奇怪。
- requestWeather () 方法中先是使用了参数中传入的天气id和知情权申请好的API Key拼装出一个接口地址，接着调用HttpUtil.sendOkHttpRequest () 方法来向该地址发出，服务器会将相应城市的天气信息以JSON格式返回。

- 然后在onResponse () 方法中回调中先调用Utility.handleWeatherResponse () 方法将返回的Json格式数据转换成Weather对象，再将当前线程切换到主线程，然后进行判断如果服务器返回的status状态是ok，就说明请求天气成功了，此时将返回的数据缓存到SharedPreferences当中，并调用showWeatherInfo () 方法进行内容显示。
- showWeatherInfo () 方法中的逻辑比较简单，其实就是从Weather对象中获取数据，然后显示到相应的控件上。注意在未来几天天气预报的部分使用了一个for循环来处理每天的天气信息，在循环中动态加载forecast\_item.xml布局并设置相应的数据，然后添加到父布局当中，设置完所有数据之后，将ScrollView重新变成可见。
- 当下一次再进入WeatherActivity时，由于缓存已经存在，因此会直接解析并显示天气数据，不会再发起网络请求。



谢 谢