1. **package** com.example.property.util;
3. **import** java.text.ParseException;
4. **import** java.text.SimpleDateFormat;
5. **import** java.util.Date;
6. **import** java.util.Locale;
8. **import** android.text.format.Time;
10. /\*\*
11. \*
12. \* @author cj 2016-5-26
13. \*
14. \*/
15. **public** **class** TimeUtil {
17. **private** SimpleDateFormat sf = **null**;
19. /\*\*
20. \* 获取当前时间
21. \*
22. \* @return
23. \*/
24. **public** String getNowTime() {
25. String timeString = **null**;
26. Time time = **new** Time();
27. time.setToNow();
28. String year = thanTen(time.year);
29. String month = thanTen(time.month + 1);
30. String monthDay = thanTen(time.monthDay);
31. String hour = thanTen(time.hour);
32. String minute = thanTen(time.minute);
34. timeString = year + "-" + month + "-" + monthDay + " " + hour + ":"
35. + minute;
36. // System.out.println("-------timeString----------" + timeString);
37. **return** timeString;
38. }
40. **public** **int** calculate(**int** year, **int** month) {
42. **boolean** yearleap = judge(year);
43. **int** day;
44. **if** (yearleap && month == 2) {
45. day = 29;
46. } **else** **if** (!yearleap && month == 2) {
47. day = 28;
48. } **else** **if** (month == 4 || month == 6 || month == 9 || month == 11) {
49. day = 30;
50. } **else** {
51. day = 31;
52. }
53. **return** day;
54. }
56. **public** **boolean** judge(**int** year) {
57. **boolean** yearleap = (year % 400 == 0) || (year % 4 == 0)
58. && (year % 100 != 0);// 采用布尔数据计算判断是否能整除
59. **return** yearleap;
60. }
62. /\*\*
63. \* 十一下加零
64. \*
65. \* @param str
66. \* @return
67. \*/
68. **public** String thanTen(**int** str) {
70. String string = **null**;
72. **if** (str < 10) {
73. string = "0" + str;
74. } **else** {
76. string = "" + str;
78. }
79. **return** string;
80. }
82. /\*\*
83. \* 计算时间差
84. \*
85. \* @param starTime
86. \*            开始时间
87. \* @param endTime
88. \*            结束时间
89. \* @param type
90. \*            返回类型 ==1----天，时，分。 ==2----时
91. \* @return 返回时间差
92. \*/
93. **public** String getTimeDifference(String starTime, String endTime) {
94. String timeString = "";
95. SimpleDateFormat dateFormat = **new** SimpleDateFormat("yyyy-MM-dd HH:mm");
97. **try** {
98. Date parse = dateFormat.parse(starTime);
99. Date parse1 = dateFormat.parse(endTime);
101. **long** diff = parse1.getTime() - parse.getTime();
103. **long** day = diff / (24 \* 60 \* 60 \* 1000);
104. **long** hour = (diff / (60 \* 60 \* 1000) - day \* 24);
105. **long** min = ((diff / (60 \* 1000)) - day \* 24 \* 60 - hour \* 60);
106. **long** s = (diff / 1000 - day \* 24 \* 60 \* 60 - hour \* 60 \* 60 - min \* 60);
107. **long** ms = (diff - day \* 24 \* 60 \* 60 \* 1000 - hour \* 60 \* 60 \* 1000
108. - min \* 60 \* 1000 - s \* 1000);
109. // System.out.println(day + "天" + hour + "小时" + min + "分" + s +
110. // "秒");
111. **long** hour1 = diff / (60 \* 60 \* 1000);
112. String hourString = hour1 + "";
113. **long** min1 = ((diff / (60 \* 1000)) - hour1 \* 60);
114. timeString = hour1 + "小时" + min1 + "分";
115. // System.out.println(day + "天" + hour + "小时" + min + "分" + s +
116. // "秒");
118. } **catch** (ParseException e) {
119. // TODO Auto-generated catch block
120. e.printStackTrace();
121. }
122. **return** timeString;
124. }
126. /\*\*
127. \* 计算相差的小时
128. \*
129. \* @param starTime
130. \* @param endTime
131. \* @return
132. \*/
133. **public** String getTimeDifferenceHour(String starTime, String endTime) {
134. String timeString = "";
135. SimpleDateFormat dateFormat = **new** SimpleDateFormat("yyyy-MM-dd HH:mm");
137. **try** {
138. Date parse = dateFormat.parse(starTime);
139. Date parse1 = dateFormat.parse(endTime);
141. **long** diff = parse1.getTime() - parse.getTime();
142. String string = Long.toString(diff);
144. **float** parseFloat = Float.parseFloat(string);
146. **float** hour1 = parseFloat / (60 \* 60 \* 1000);
148. timeString = Float.toString(hour1);
149. } **catch** (ParseException e) {
150. // TODO Auto-generated catch block
151. e.printStackTrace();
152. }
153. **return** timeString;
155. }
157. /\*\*
158. \* 获取时间中的某一个时间点
159. \*
160. \* @param str
161. \* @param type
162. \* @return
163. \*/
164. **public** String getJsonParseShiJian(String str, **int** type) {
165. String shijanString = **null**;
166. String nian = str.substring(0, str.indexOf("-"));
167. String yue = str.substring(str.indexOf("-") + 1, str.lastIndexOf("-"));
168. String tian = str.substring(str.lastIndexOf("-") + 1, str.indexOf(" "));
169. String shi = str.substring(str.indexOf(" ") + 1, str.lastIndexOf(":"));
170. String fen = str.substring(str.lastIndexOf(":") + 1, str.length());
172. **switch** (type) {
173. **case** 1:
174. shijanString = nian;
175. **break**;
176. **case** 2:
177. shijanString = yue;
178. **break**;
179. **case** 3:
180. shijanString = tian;
181. **break**;
182. **case** 4:
183. shijanString = shi;
184. **break**;
185. **case** 5:
186. shijanString = fen;
187. **break**;
189. }
190. **return** shijanString;
191. }
193. /\*\*
194. \* Sring变int
195. \*
196. \* @param str
197. \* @return
198. \*/
199. **public** **int** strToInt(String str) {
200. **int** value = 0;
201. value = Integer.parseInt(str);
202. **return** value;
203. }
205. /\*\*
206. \* 与当前时间比较早晚
207. \*
208. \* @param time
209. \*            需要比较的时间
210. \* @return 输入的时间比现在时间晚则返回true
211. \*/
212. **public** **boolean** compareNowTime(String time) {
213. **boolean** isDayu = **false**;
215. SimpleDateFormat dateFormat = **new** SimpleDateFormat("yyyy-MM-dd HH:mm");
217. **try** {
218. Date parse = dateFormat.parse(time);
219. Date parse1 = dateFormat.parse(getNowTime());
221. **long** diff = parse1.getTime() - parse.getTime();
222. **if** (diff <= 0) {
223. isDayu = **true**;
224. } **else** {
225. isDayu = **false**;
226. }
227. } **catch** (ParseException e) {
228. // TODO Auto-generated catch block
229. e.printStackTrace();
230. }
232. **return** isDayu;
233. }
235. /\*\*
236. \* 把时间戳变yyyy-MM-dd HH:mm格式时间
237. \*
238. \* @param time
239. \* @return
240. \*/
241. **public** String getDateToString(**long** time) {
242. Date d = **new** Date(time);
243. sf = **new** SimpleDateFormat("yyyy-MM-dd HH:mm");
244. **return** sf.format(d);
245. }
247. /\*\*
248. \* 返回时间戳
249. \*
250. \* @param time
251. \* @return
252. \*/
253. **public** **long** dataOne(String time) {
254. SimpleDateFormat sdr = **new** SimpleDateFormat("yyyy-MM-dd HH:mm",
255. Locale.CHINA);
256. Date date;
257. **long** l = 0;
258. **try** {
259. date = sdr.parse(time);
260. l = date.getTime();
261. } **catch** (ParseException e) {
262. e.printStackTrace();
263. }
264. **return** l;
265. }
267. /\*\*
268. \* 比较两个时间
269. \*
270. \* @param starTime
271. \*            开始时间
272. \* @param endString
273. \*            结束时间
274. \* @return 结束时间大于开始时间返回true，否则反之֮
275. \*/
276. **public** **boolean** compareTwoTime(String starTime, String endString) {
277. **boolean** isDayu = **false**;
278. SimpleDateFormat dateFormat = **new** SimpleDateFormat("yyyy-MM-dd");
280. **try** {
281. Date parse = dateFormat.parse(starTime);
282. Date parse1 = dateFormat.parse(endString);
284. **long** diff = parse1.getTime() - parse.getTime();
285. **if** (diff >= 0) {
286. isDayu = **true**;
287. } **else** {
288. isDayu = **false**;
289. }
290. } **catch** (ParseException e) {
291. // TODO Auto-generated catch block
292. e.printStackTrace();
293. }
294. **return** isDayu;
296. }
298. **public** **boolean** compareTwoTime2(String starTime, String endString) {
299. **boolean** isDayu = **false**;
300. SimpleDateFormat dateFormat = **new** SimpleDateFormat("yyyy-MM-dd HH:mm");
302. **try** {
303. Date parse = dateFormat.parse(starTime);
304. Date parse1 = dateFormat.parse(endString);
306. **long** diff = parse1.getTime() - parse.getTime();
307. **if** (diff >= 0) {
308. isDayu = **true**;
309. } **else** {
310. isDayu = **false**;
311. }
312. } **catch** (ParseException e) {
313. // TODO Auto-generated catch block
314. e.printStackTrace();
315. }
316. **return** isDayu;
318. }
320. /\*\*
321. \* 获取年
322. \*
323. \* @param time
324. \* @return
325. \*/
326. **public** String getTimeYear(String time) {
328. String substring = time.substring(0, time.lastIndexOf(" "));
329. **return** substring;
331. }
333. /\*\*
334. \* 换算小时，0.5小时-->0小时30分
335. \*
336. \* @param hour
337. \* @return
338. \*/
339. **private** String convertTime(String hour) {
341. String substring = hour.substring(0, hour.lastIndexOf("."));
342. String substring2 = hour.substring(hour.lastIndexOf(".") + 1,
343. hour.length());
344. substring2 = "0." + substring2;
345. **float** f2 = Float.parseFloat(substring2);
346. f2 = f2 \* 60;
347. String string = Float.toString(f2);
348. String min = string.substring(0, string.lastIndexOf("."));
349. **return** substring + "小时" + min + "分";
351. }
352. }