

Android Thread

appSchool.co.il

AsyncTask

Android responsiveness

- responsiveness היא דאגה עיקרית של אנדרואיד
- אנדרואיד נותן 5 שניות כדי להפעיל יישום או מציג שגיאה.

אסטרטגיות לשמירה על responsiveness

- AsyncTask
- thread strategies handler
- Use services

AssynTask

1. המחלקה מורכבת מטיפוסים גנריים - `params`, `Progress`, `Result`

`AsynTasks` שיפעל יותר מכמה שניות יכול להיות שיהרג על ידי מערכת הפעלה.

AssynTask

ל- AsyncTask יש 4 פונקציות עיקריות

1. onPreExecute - תהליך שמטפל במשימות גדולות. **לא ניתן**

לבצע בו עדכון ui

2. doInBackground - תהליך שנעשה ברקע **אך לא ניתן לעדכן בו**

את ממשק המשתמש

3. onProgressUpdate - **ניתן לשנות בעזרתו את ממשק**

המשתמש

4. onPostExecute - מקבל כפרמטר את result שמוחזר על ידי

הפונקציה doInBackground. **ומעדכן את ממשק המשתמש.**

ל- AsyncTask יש גם פעולת עזר publishProgress שתפקידה

לבצע את העדכונים.

AsyncTask<Params,Progress,Result>

params - זה מערך של נתונים שנקבל ביצירה של האובייקט. לדוגמה מערך של תמונות להורדה.

Progress - סוג הנתונים של יחידות ההתקדמות

Result - סוג הנתונים של התוצאה. סוג הנתונים שתחזיר פונקציית doBackground.

במידה ונסמן Void - זהו טיפוס ללא שימוש

במידה ואנו רואים סימון של ... (שלוש נקודות) זה מסמל מערך.

הכרזה על class AsyncTask

```
public class loadfilebackground extends AsyncTask<String, Integer,  
Bitmap[]>
```

params progress Result

```
protected void onPreExecute() {}
```

```
protected Bitmap[] doInBackground(String... params) { return bitmaps}
```

```
protected void onProgressUpdate(Integer... values) {}
```

```
protected void onPostExecute(Bitmap[] result) {}
```

על מנת לבצע ביטול של AsyncTask ניתן לעשות זאת על ידי isCancelled

Demo for example

Handler

UI Thread

1. לכל אפליקציה קיים main thread שאחראי לביצוע כל האירועים של האפליקציה
2. ה-main thread אחראי לאינטראקציה עם ה-ui ולכן הוא נקרא ui thread
3. במידה וה-main thread ימתין יותר מ-5 שניות האפליקציה תקרוס
4. לכן כאשר יהיו לנו משימות מורכבות לא נעשה אותן ב-thread הראשי אלה נפנה אותו ל-thread נפרד שיבצע זאת במקביל ל-ui thread.

Multi threading

1. ל-thread אסור לבצע שינויים בממשק המשתמש -
אנדרואיד לא מאפשר זאת.
2. רק ה-thread הראשי רשאי לבצע שינוי ב-ui.
3. כל יישום מורכב מ-
 - a. Main thread
 - b. thread'ים נוספים
4. אינטראקציה בין ה-thread'ים מתבצעת ב-2
 - אפשרויות
 - a. אובייקט Handler
 - b. AsyncTask

Handler

1. מתווך בין ה-main thread לשאר ה-threads.
2. ה-handler מכיל תור של מסרים. כמו כן ה-handler מקושר לתור המסרים שיצר אותו. תפקידו של ה-handler לתזמן את המסרים ולהכניס פעולות לתור המסרים על מנת שיתבצעו ב-thread הראשי.
3. כל Thread שולח הודעות לתור ההודעות של ה-handler.
4. תפקידו של ה-handler לשלח מסרים ואובייקטי runnable ל-main thread ולבצע את מסר שקיבל מכל thread ו-thread ה-handler נעזר בשתי פונקציות
 - a. sendMessage
 - b. obtainMessage
- 5.

סדר פעולות שימוש ב-handler

1. ניצור אובייקט handler או אובייקט שיורש מ-mainthread handler
2. נממש את הפונקציה handleMessage שתטפל בהודעה ובממשק המשתמש ב-main thread.
3. ניצור thread משני כלשהו או אובייקט שיורש מ-thread
4. ניצור אובייקט message שאותו נרצה לשלח לתור ההודעות.
5. ונשלח את ההודעה ע"י handler.sendMessage (message)

יום ללא תכנות הוא יום מבוזבז!