



Pubsale 2015

ANDROID APP AND SERVERSIDE

מבנה הפרויקט

הפרויקט הוא אפליקציית אנדרואיד שמתקשרת עם שרת REST מרוחק שבתורו מתקשר עם מסד נתונים MySQL



הפעלת הפרויקט

ניתן להוריד את הפרויקט וכל הקבצים הקשורים מהקישור הבא:

<https://drive.google.com/folderview?id=0BxlaxUoVu3MoamZDZVRhTzRTRzQ&usp=sharing>

הקוד נמצא ב תיקייה PubSale2015.zip

משום שאין ברשותי שרת ההפעלה ההפעלה כרגע דורשת

לערוך את הקובץ PubServiceClient ולהגדיר נכון את הכתובת של השרת ב שורה 25

גרדע האפליקציה מכוונת לעשרות רק דרך שרת פרוקסי של [Fiddler](#) שדרכו עבדתי רוב הזמן כדי לנתר את בקשות ה http.

אני מציע שעבור הצגה של פעולת הפרויקט תבוצע התחברות מרחוק ע"י TeamViewer למכונת הפיתוח שלי שכן היא נייחת.

טכנולוגיות

התוכנית פותחה ב סביבת IntelliJ Idea ultimate 15.0.1 שיש לה רשיון סטודנט. ואיפשרה לי לפתח את צד השרת והלקוח באותה סביבה באותו בפרויקט.

בקרת ה תצוגה נעשתה באמצעות GIT ו BitBucket

השתמשתי ב JAVA 8 בצד השרת

וב JAVA7 בצד הלקוח (גרסה 16 של ה API של אנדרואיד עוד לא תומכת ב JAVA8 ורציתי גרסה שתתמוך במכשיר שיש לי))

ניהול התלויות נעשה באמצעות [MAVEN](#) אל אף שאנדרואיד מעדיפים שיעבדו עם [GRADLE](#)

מבנה הקוד

- Interfaces – ממשקים כלליים ואובייקטים POJO
- DAL – גישה לדאטאבייס + לוגיקת צד שרת שאמורה להיות בפרוייקט נפרד אבל התעייפתי
- Android – קוד צד לקוח
 - App.Activities לוגיקות של מסכים
 - App.Fragments לוגיקה של קומפונטות של מסכים שמשתמשים בהם מספר פעמים
 - Client-לוגיקת תקשורת
 - Res.layouts עיצוב המסכים
 - Res.strings כל המחזורות המוצגות בתוכנה (שיהיה קל לתרגם)
- Webservice – מימוש של RS-JAX ופניה ל DAL

תיאור הזרימת הנתונים

כאשר הלקוח מבצע פעולה באנדרואיד שדורשת קריאה לשרת מתבצעת קריאת HTTP ע"י ספריית [RETROFIT](#). הספרייה מאפשרת לכתוב interface ובעזרת annotations ליצור קוד לקוח ל שירות REST HTTP. הבקשה עוברת סריליזציה ל JSON באמצעות [JACKSON](#) ונשלחת לשרת. השרת המארח הוא מסוג [GLASSFISH](#) ו מממש JAX-RS ע"י ספריית [Jersey](#) הבקשות שמגיעות מהלקוח עוברות דה-סריליזציה. ומתורגמות במקרה הצורך עי [ModelMapper](#) ממבני הנתונים של פרוטוקול התקשורת למבני נתונים המוכרים ע"י DB. כעת מתרחשת הלוגיקה של השרת ומתבצעת הפעולה המתאימה מול מסך הנתונים. העבודה מול מסד הנתונים מתבצעת באמצעות ספריית [hibernate](#) שהיא המימוש הידוע ביותר ל JPA2.0 ובעצם ותיקה יותר מ JPA בכללותו. (בהתחלה השתמשתי ב hibernate ללא JPA ואז גיליתי כמה יתרונות בזה. הכתיבה נעשתה בשיטת code-first כשאר הקוד יוצר את סכמת מסד הנתונים מתוך ה annotations על מבני הנתונים שמייצגים את טבלאות ה db ולסיכום השתמשתי בספריית [JINQ](#) המאפשרת לכתוב שאילתות בצורה פונקציונלית כדי לקבל עבודה יותר נוחה מול מסד הנתונים כמו ב [entity framework](#) המייקרוסופטי. השאילתות נעמו מול שרת [mysql](#) סטנדרטי לגמרי