

תיעוד

מבנה כללי

המבנה הכללי של הפרויקט הינו 3 תוכניות: אפליקציה הבנויה בשיטת, שרת, ואתר web. בכל אחד מהפרויקטים הנ"ל מופיעים ה packages הבאים:

• אפליקציה:

- control – טיפול בלוגיקה של חלונות האפליקציה
- entities – ישויות
- entities.enums – סדרות המבטאות תכונות של ישויות
- entities.json – ישויות שנועדו לעזור בשימוש בטכנולוגית json.
- Model.backend – ישויות המשומשות בגישה ל DB.
- Model.datasource – מימוש הגישה ל DB.

• שירות:

- entities – כנ"ל
- entities.enums – כנ"ל
- entities.json – כנ"ל
- java5775_5366_9373Service – מימוש הגישה ל DB.

• אתר:

- entities – כנ"ל
- entities.enums – כנ"ל
- entities.json – כנ"ל
- java5775_5366_9373Web – מימוש הגישה ל DB.

תיאור תפיסה כללית

התפיסה הכללית שלנו בפרויקט הייתה לבנות את הפרויקט ע"פ הוראות ה OSF, ולהוסיף אליו רק במקרים בהם מצאנו את הכתוב ב OSF כבעייתי ב context של הפרויקט שלנו.

בעיות ופתרונות טכניים

1. מעבר אובייקטים מורכבים דרך extras בין activities:
השימוש ב interface Serializable ובפונקציה `intent.putExtra(string, serializable)` לא פעל בניסיון שלי להעביר extra של ישות המוגדרת ב entities. בכדי לפתור בעיה זו לקחתי מהאינטרנט מחלקה `ObjectSerializer.java` והיא נמצאת ב package entities. מחלקה זו מקבלת אובייקט מורכב ועושה לו סיריאליזציה ל string ואז ניתן להעביר אותו דרך `intent.putExtra(string, string)`
2. בחירת תאריך באפליקציה:
מצאנו את בוחרי התאריך הפשוטים של אנדרויד גדולים מדי בשביל צרכינו, ולכן יצרנו בוחר תאריך משלנו שמוגדר ב `control.CustomDatePicker.java`.
3. מעבר מספר אובייקטים דרך json.
נצרכנו עברו מספר פונ' לקבל שני פרמטרים עבור פונ' post, ולא מצאנו שם דרך לעשות זאת, אזי יצרנו אובייקט חדש שיכיל את הפרמטרים האלא כתכונות שלו, כך

שמועבר רק אובייקט אחד.

4. האם לבצע `setLists()`?

בכדי לדעת האם לבצע פעולת איתחול בסיס הנתונים `setLists()` יצרנו פונ' `isEmpty()` הבודקת אם טבלת הדוקטורים ריקה, ביציאה מנקודת ההנחה, כי היות וזו טבלה שצריך למלא בלפחות אובייקט אחד בכדי שאפליקציה תוכל לרוץ, אזי אם היא ריקה, כל הטבלאות עדין רקות. פונ' זו בעייתית ב `context` של `service` היות ובמהלך בדיקת בסיס הנתונים הוכנסו פרטים לבסיס הנתונים, ומנעו מפונ' `setLists()` לרוץ ולו פעם אחת.

5. מעבר טיפוסים מסוג `date` דרך `json`.

בכדי להעביר טיפוס מסוג `date` דרך `json`, אזי `gson` ממיר אותו ל `unix time stamp`. דבר זה אולי יעיל ביותר למעבר המידע ברשת, אך הקשה אלינו בהמרת הנתונים, היות וב `gson` אין אנו ממירים כל תכונה בנפרד אלא בונים את כל האובייקט בבת אחת, ולא ניתן להמיר תכונות נפרדות, אזי היה אלינו ליצור אובייקט מעבר של דוקטור שמיצג תאריך ע"י מספר – דבר מסובך ומגעיל. במקום זאת במקומות בהם היינו צריכים זאת, השתמשנו במחלקה החדשה יותר, והטובה יותר `ObjectMapper`, בה ניתן לשנות את הפורמט של התאריך בסיריאלזציה, לפורמט, שהמשתנה `date` יודע לתרגם חזרה לתאריך.

הצעות לשיפור

1. אנו מציאים לבדוק מדוע הפונ' `intent.putExtra(string, serializable)` לא מתפקדת כראוי, והאם כדי להציג לכלל התלמידים בשנה הבאה את `ObjectSerializer.java`.
2. אנו ממליצים לזרוק את `gson` ולעבור ל `objectMapper` באופן מוחלט לשנה הבאה.
3. אנו ממליצים, כי במקום שנצטרך להעתיק קטעי קוד שלמים בין שלושת התוכנות בפרויקט, נבנה ספריה שתכיל את כל הקוד המועתק הזה, ותשרת את כל 3 התוכנות: האפליקציה, השרת והאתר.