

אבן דרך 2 – גרסה קלה

סיפור משתמש 1

יאיר פותח את הדפדפן האהוב עליו בכתובת localhost:8080. מופיע לו דף עם כתורת Anomaly Detection Server. יאיר בוחר אלגוריתם לגילוי חריגות מתוך רשימה נגללת שכוללת אלגוריתם מבוסס רגרסיה ואלגוריתם היברידי (אותם אלגוריתמים מסמסטר א'). יאיר בוחר קובץ CSV תקין ללא חריגות וקובץ נוסף שייתכנו בו חריגות. יאיר לוחץ על upload והקבצים מועלים לשרת, שם מתבצע גילוי חריגות והפלט עם דיווח החריגות מופיע באותו הדף בו צופה יאיר.

סיפור משתמש 2

באמצעות שפת תכנות כלשהי נפתלי שולח פקודת HTTP POST לכתובת localhost:8080. הפקודה כוללת בחירה של אלגוריתם לגילוי חריגות מבוסס רגרסיה או האלגוריתם היברידי (אותם אלגוריתמים מסמסטר א'), בחירה של קובץ CSV תקין ללא חריגות ובחירה של קובץ נוסף שייתכנו בו חריגות. נפתלי מקבל חזרה JSON שכולל דיווח היכן התרחשו החריגות.

תשימו לב ש

- בשני המקרים מדובר בפקודת HTTP אחת שכוללת את כל המידע ועל כן היא stateless. כך אנו שומרים על עקרונות ה REST.
- ניתן לממש את סיפור משתמש 2 ולעשות לו reuse עבור 1. ה action של טופס ה HTML יהיה פשוט שליחת פקודת ה HTTP POST וה JSON שיוחזר יעובד לתצוגה ב HTML.

תוכלו להשתמש באיזו טכנולוגיה שתמצאו נוחה כדי לממש את אפליקציית ה web לעיל. בפרט תוכלו להיצמד לדוגמאות בהרצאה ולהשתמש ב java | tomcat, או בטכנולוגיה חדישה יותר כמו node.js למשל.

ארכיטקטורה – MVC:

:View

מדובר בדף אינטרנט פשוט (HTML) עם form שכולל רשימה נגללת לבחירת האלגוריתם, שני שדות input לבחירת קבצים, ו frame פנימי לצורך התצוגה של הפלט.

:Controller

עבור פקודת http get ל "/" ה controller יחזיר לדפדפן את טופס ה HTML שהגדיר ה view. זה תלוי טכנולוגיה. אפשר גם לממש הפניה ישירה לקובץ HTML באמצעות השרת שעושים בו שימוש כמו בדוגמה עם tomcat.

עבור פקודת http post ל "/detect" ה controller מפענח את הבקשה ומורה למודל לבצע את החישוב. כשתוחזר התוצאה (כ JSON) ה controller מורה ל view ליצור את הפלט המבוקש. כל זה צריך להיעשות בצורה אסינכרונית. שימו לב שברוב הטכנולוגיות יש לכם את היכולת הזו מוטמעת בפנים.

:Model

המודל יכול לממש בעצמו את האלגוריתמים מחדש או פשוט להתממשק (כלקוח) לשרת שנכתב במסמסטר א' ולהפעיל אותו. הוא יחזיר את דיווח החריגות כ JSON

חזרה ל view:

ה controller מורה ל view מהיכן למשוך את ה json והוא בתורו מפענח ומציג אותו. כלומר יש לג'נרט את הקוד שנשלח ומוצג בצד הלקוח אצלו בדפדפן. התצוגה כוללת דוח שמורכב מרשימה קריאה וכרונולוגית של דיווח החריגות - לאיזו עמודה נתגלתה חריגה באיזה זמן.

כדי להרשים (אך לא חובה) תוכלו גם להציג ויזואליזציה כלשהי של החריגה בדומה למה שנעשה באבן דרך 1.

בהצלחה!