**XMLעבודה נכונה עם קבצי**

**נתי צדוק**

בעבודה עם פרויקטים של Maven, חשוב לארגן טסטים בצורה מובנית ויעילה. גישה אחת היא לחלק בדיקות למספר קבצי XML במקום לתחזק קובץ אחד גדול המכיל את כל הבדיקות,חלוקת בדיקות למספר קבצי XML מקלה על ניהולם. כאשר פרויקט גדל בגודלו ובמורכבותו, מספר הטסטים גדל גם הוא, ונרצה יכולת שליטה על כל הבדיקות שנעשות באוטומציה, נראה את היתרונות בכך.

**חלוקה לוגית נכונה**

ניהול מספר רב של בדיקות בקובץ XML בודד עלול להפוך למסורבל, כאשר בדיקות מפוזרות על פני חלקים שונים של הקובץ, מה שמקשה על איתור בדיקות ספציפיות. עם קבצי XML מרובים, ניתן לארגן בדיקות באופן לוגי תקין לקבצים נפרדים על סמך סוגם או הפונקציונליות שלהם, מה שמקל על הניהול והתחזוקה שלהם.

**בדיקה סלקטיבית וחסכון בזמני ריצה**

יתרון נוסף של חלוקת בדיקות למספר קבצי XML הוא שהיא מאפשרת בדיקה סלקטיבית. במקרים מסוימים, ייתכן שלא יהיה צורך להפעיל את כל הבדיקות, והפעלת כל הבדיקות עשויה להימשך זמן רב, ולהאט את תהליך הריצה. על ידי חלוקת בדיקות למספר קבצים, ניתן להפעיל באופן סלקטיבי בדיקות ספציפיות או קבוצות של בדיקות, ולחסוך זמן ומשאבים.

**שיתוף בין פרוייקטים**

חלוקת בדיקות למספר קבצי XML מקל גם על שיתוף ושימוש חוזר בבדיקות בין פרויקטים. בפרויקט גדול ומורכב, מקובל לקיים טסטים המשותפים בין חלקים שונים של הפרויקט. על ידי הצבת בדיקות משותפות בקבצי XML נפרדים, ניתן לפנות אליהם בקלות ולעשות בהם שימוש חוזר בחלקים שונים של הפרויקט. זה יכול לחסוך זמן ומאמץ, מכיוון שאין צורך לשכפל בדיקות על פני חלקים שונים של הפרויקט.

**הרצה במקביל**

יתרה מכך, על ידי חלוקת בדיקות למספר קובצי XML, ניתן להריץ בדיקות במקביל, מה שיכול להפחית משמעותית את הזמן שלוקח להפעיל בדיקות. בדיקה מקבילה כוללת הפעלת מספר בדיקות בו-זמנית על מכונות שונות, מה שמאפשר להריץ מספר רב של בדיקות בפרק זמן קצר יותר. זה יכול לעזור להאיץ את תהליך הריצה, להוביל למשוב מהיר יותר ותיקוני באגים מהירים.

**ניפוי באגים** **ובידוד כשלי בדיקות**

יתרון נוסף של חלוקת בדיקות למספר קבצי XML הוא בכך שהיא מספקת דרך לנפות באגים ולפתור תקלות בבדיקות בצורה יעילה יותר. כאשר כל הבדיקות כלולות בקובץ XML בודד, זה יכול להיות מאתגר לקבוע איזו בדיקה גרמה לכשל, במיוחד אם הקובץ מכיל מספר רב של בדיקות.

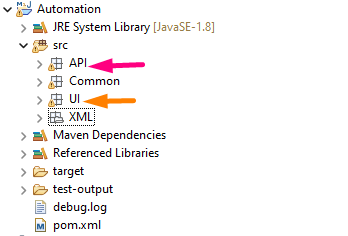
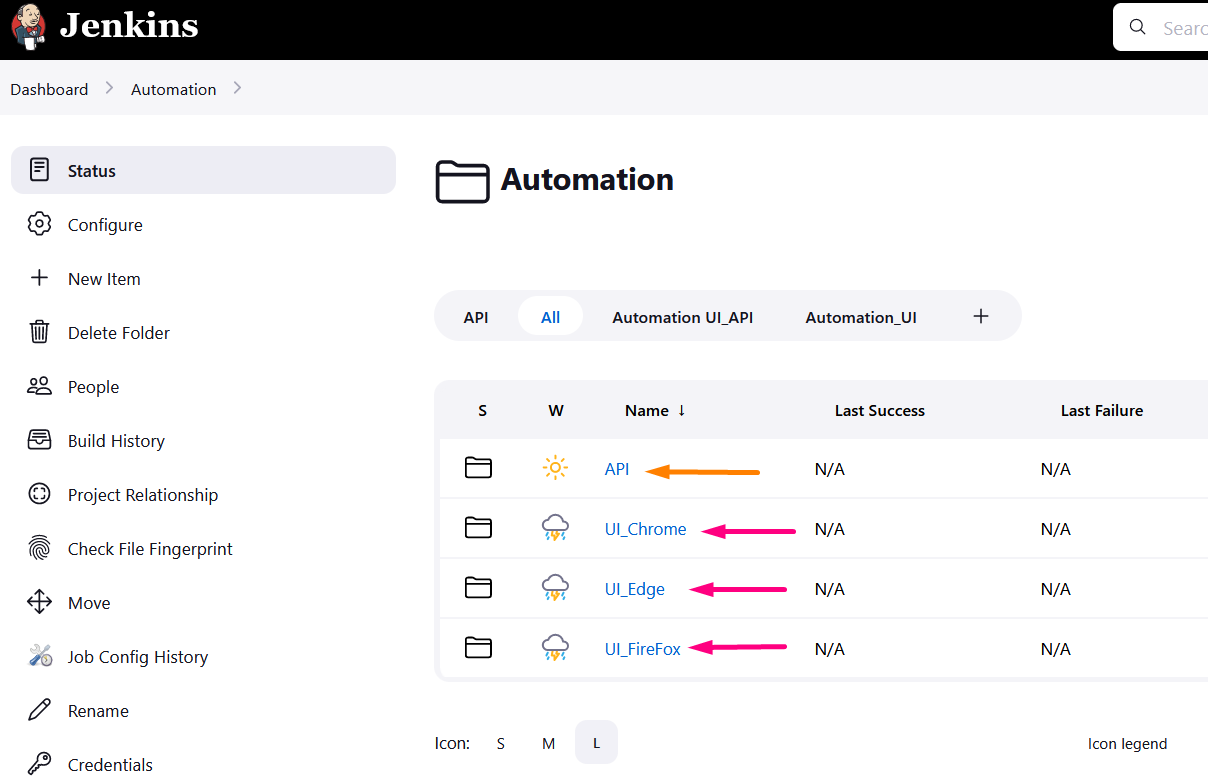
על ידי חלוקת בדיקות למספר קבצי XML, קל יותר לבודד ולזהות את המקור לכשל בבדיקה. אם בדיקה נכשלת,יכולים להסתכל על קובץ ה-XML המכיל את הבדיקה המסוימת ולמקד את מאמצי ניפוי הבאגים בקובץ הספציפי הזה. זה חוסך זמן ומאמץ על ידי מניעת הצורך לסרוק קובץ גדול כדי לזהות את הבדיקה הספציפית שנכשלה.   
בנוסף, אם ידוע כי בדיקה מסוימת נכשל או נוטה לכישלון, אנו יכולים להשבית את הבדיקה הספציפית הזו על ידי הוספת הערה בקובץ ה-XML הרלוונטי. זה מונע את הפעלת הבדיקה מבלי להשפיע על הבדיקות האחרות בפרויקט.

**עבודה מול Jenkins**

בעת שימוש בכלי אינטגרציה מתמשכת ופריסה רציפה (CI/CD) כגון Jenkins, חלוקת בדיקות למספר קבצי XML יכולה לספק אפשרויות ניפוי באגים נוספות. ניתן להגדיר את Jenkins להריץ תת-קבוצה של בדיקות בהתבסס על הקובץ הספציפי ששונה או נכשל בבנייה קודמת. המשמעות היא שאם בדיקה נכשלת, אנו יכולים להריץ מחדש במהירות רק את הבדיקות בקובץ המושפע, במקום להפעיל מחדש את כל הבדיקות.

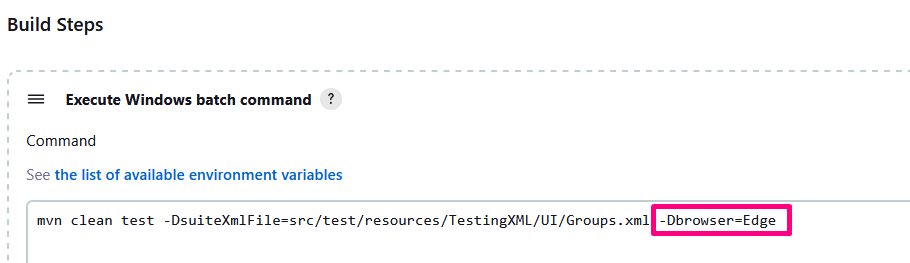
**חלוקה נכונה לתיקיות בג'נקינס**

ניצור מבניות זהה בין תשתית הקוד של האוטומציה לבין התיקיות בג'נקינס , נשכפל את בדיקות הUI לתיקיות לפי דפדפנים שאנחנו תומכים, במידה ויש לנו בדיקות API נחלק את הבדיקות לתיקיה נפרדת .



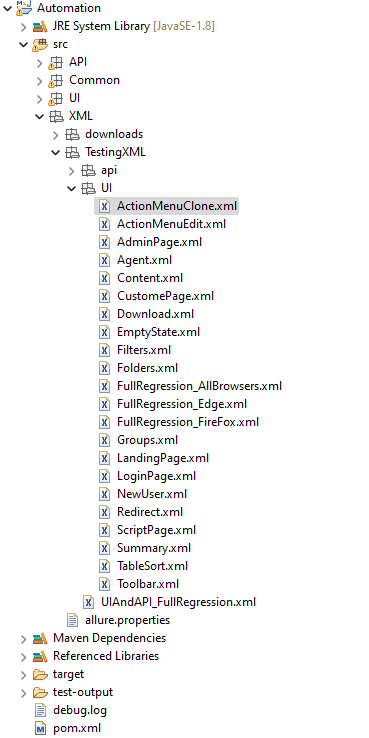
בכל תיקייה של UI ירוצו הטסטים הקיימים באוטומציה אך עם flag -Dbrowser של הדפדפן הספציפי , מספר התיקיות יהיה על פי מספר הדפדפנים שאנחנו תומכים.

שורת ההרצה של ג'וב מתוך תיקיית UI\_Edge לדוגמא תראה כך :

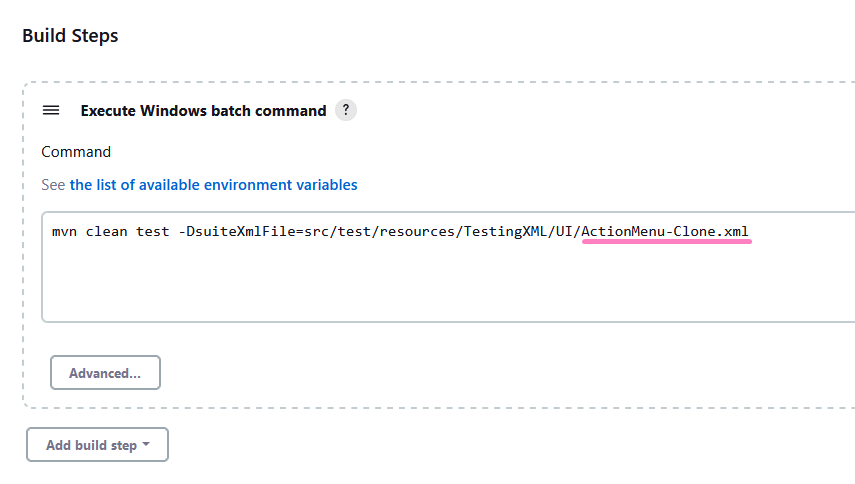


**מקביליות xml וג'ובים בג'נקינס**

עבור כל קובץ xml המכיל סוויטה של טסטים ניצור עבורו ג'וב בג'נקינס שיריץ רק אותו.



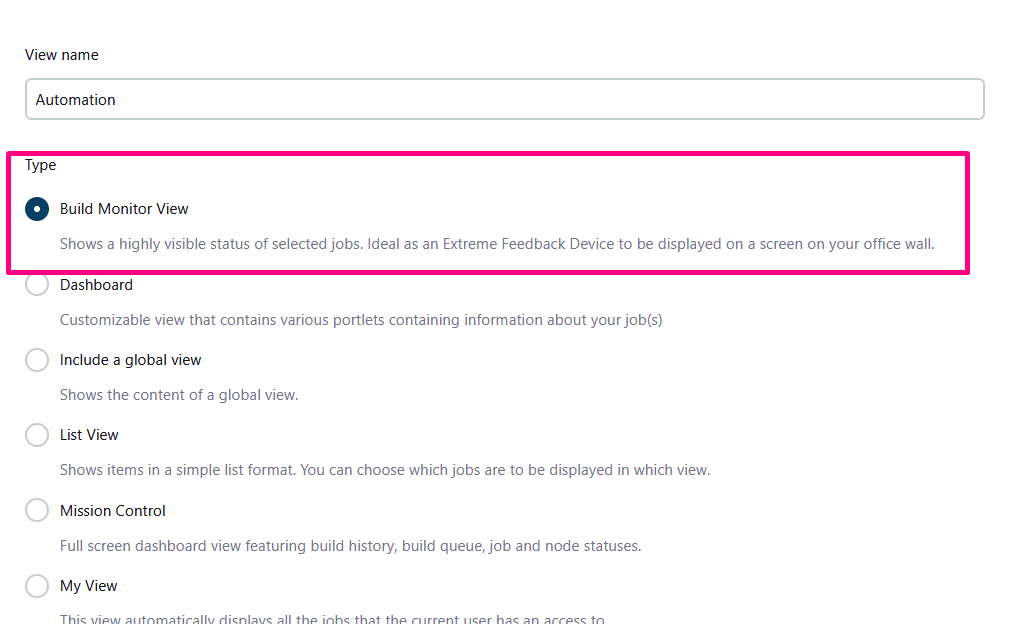
שורת הפקודה תהיה :



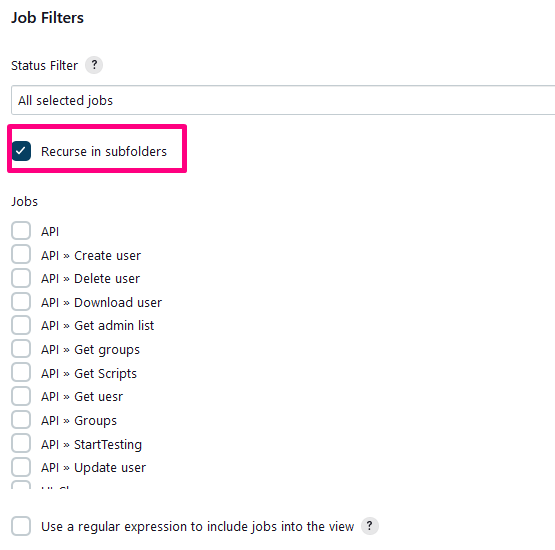
**Monitor view**

כחלק מהיתרון שאנחנו מקבלים עם חלוקת קבצי xml ללא שימוש בקובץ אחד שמכיל את כל הטסטים ולאחר התקנת monitor view plugin בJenkins אנחנו יכולים ליצור לוחות לקבלת סקירה ברמה גבוהה של התקינות והסטטוס של מספר ג'ובים, עם יכולת הרצה של טסטים בודדים ,יכולת דיבוג גבוהה יותר ויכולת אפיון בעיות בצורה רחבה ופרטנית כאחד.נוכל ליצור לוחות לפי הצורך –לוח שיכיל טסטים לפי דפדפן מסויים, לוח שיכיל רק בדיקות UI ואחד שיכיל הכל. תצוגה זו שימושית למעקב אחר ההתקדמות של מספר ריצות או ג'ובים במבט חטוף, מבלי צורך לנווט לכל אחד בנפרד.

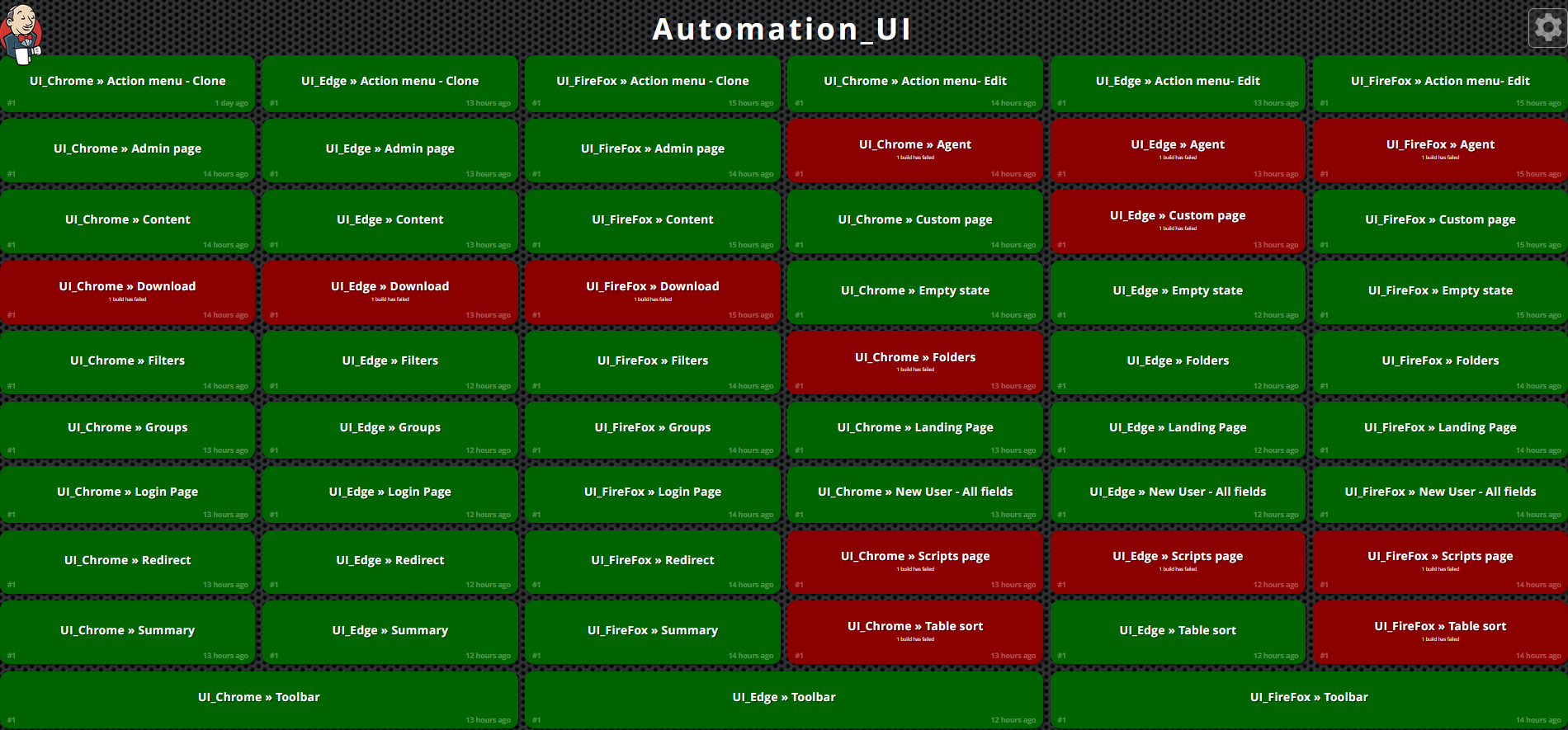
אחד היתרונות המרכזיים של השימוש ב-Monitor View הוא שהיא מספקת נראות בזמן אמת למצב של מספר ריצות או ג'ובים, ומאפשרת לצוותים לזהות במהירות כל בעיה או כשל שדורשים התייחסות. על ידי מעקב בזמן אמת, צוותים יכולים לנקוט בפעולה מיידית כדי לפתור בעיות ולמנוע מהן להסלים לבעיות משמעותיות יותר.

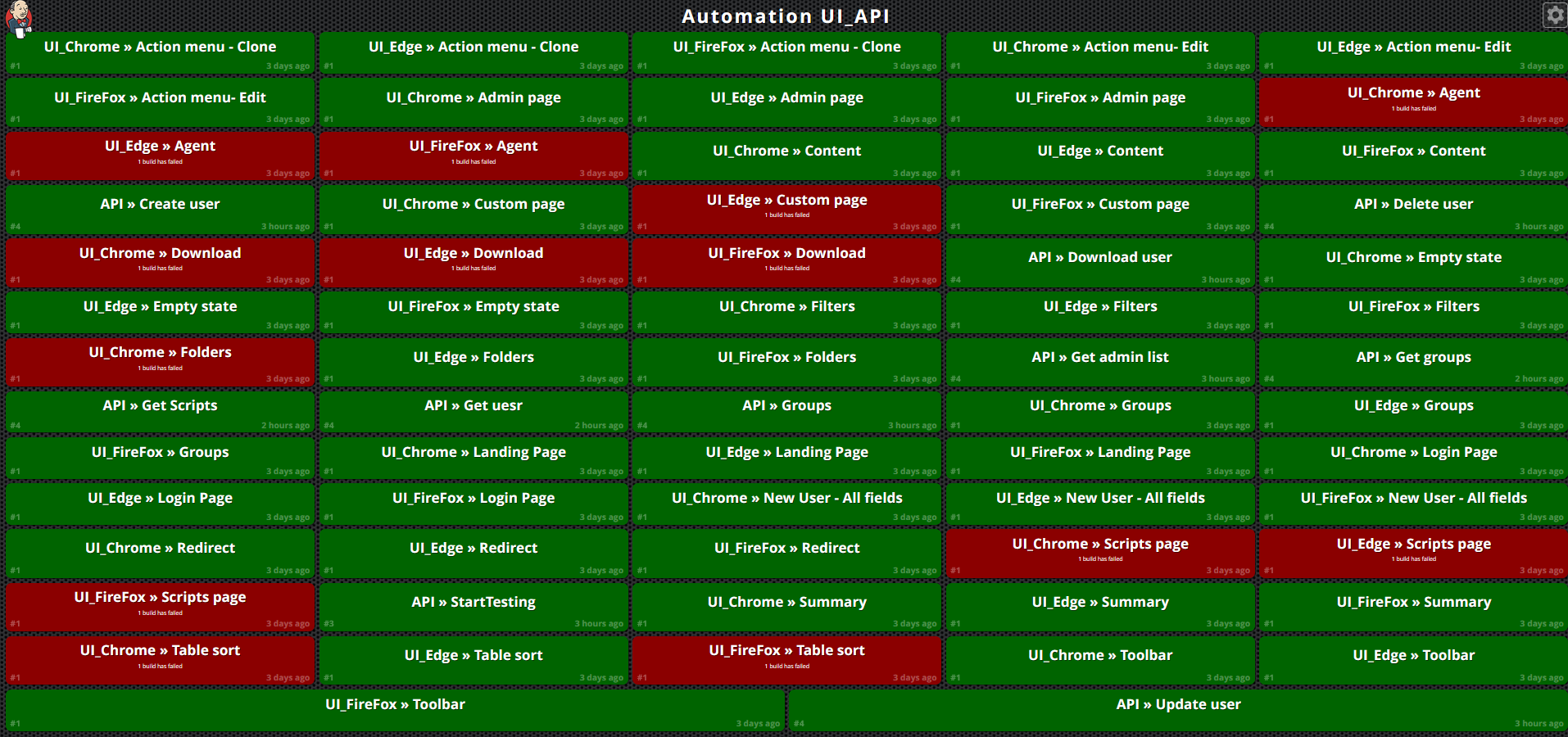


1

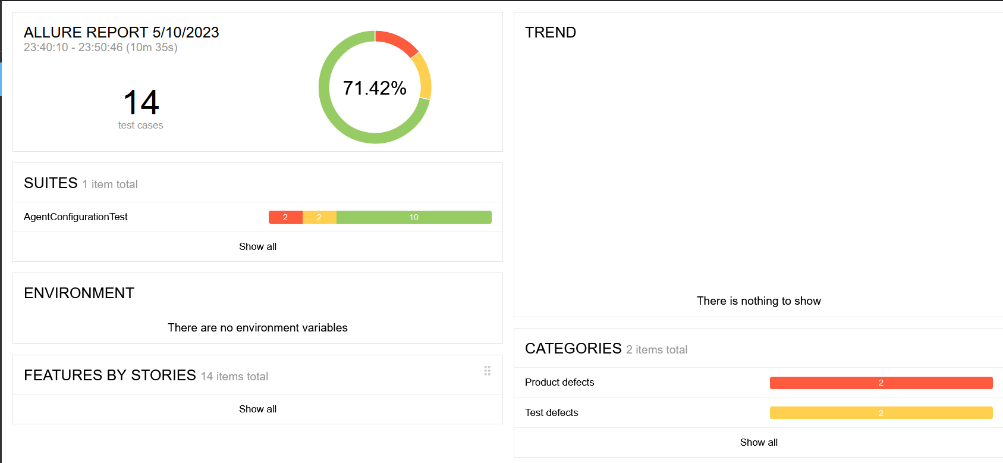


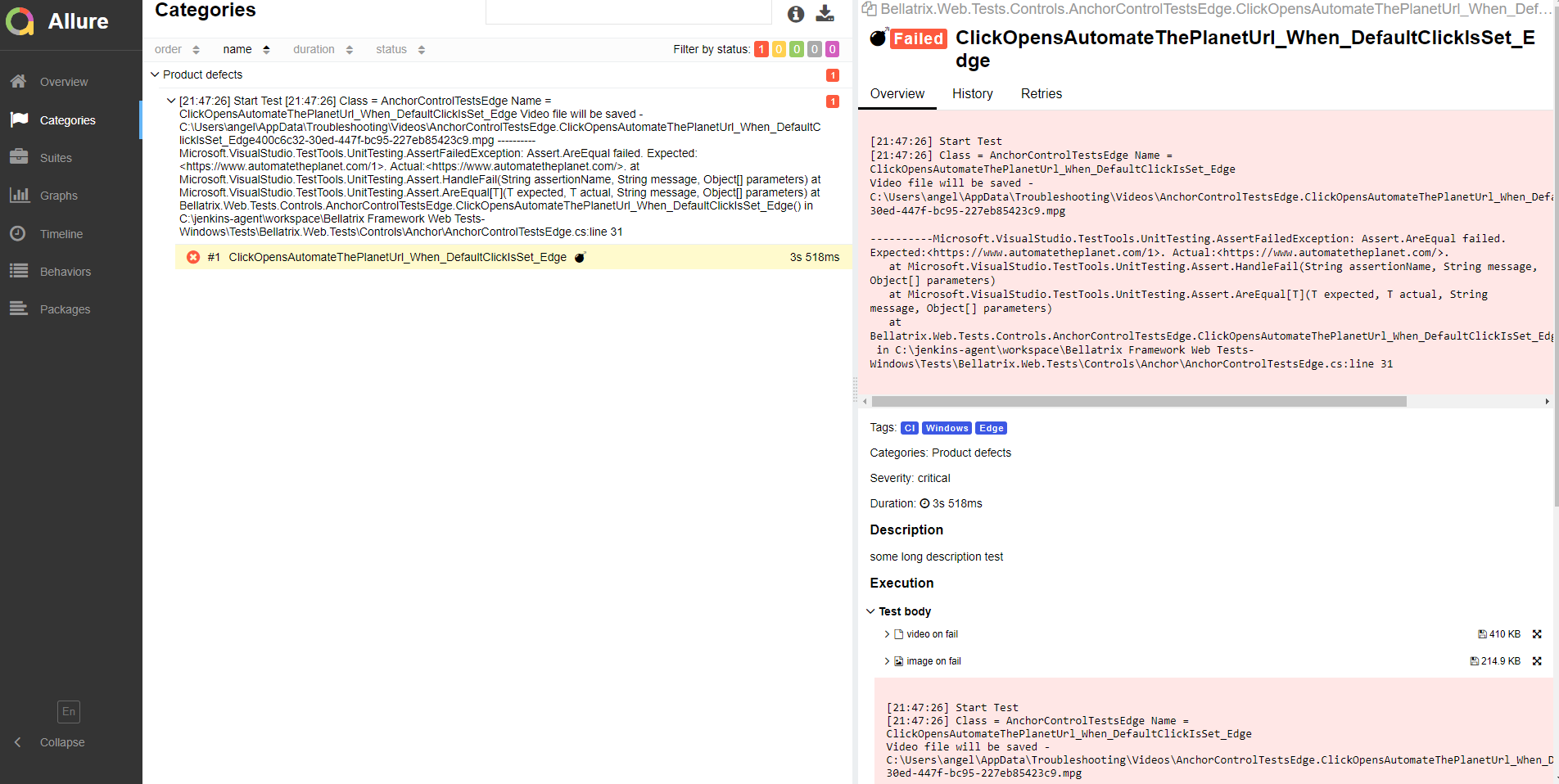
2

ניצור בורד אחד עבור כל האוטומציה של הUI

ונשתמש ביכולת הזאת ליצור גם בורד עבור אוטומציה של UI ו- API.

כל לחיצה על ריבוע מתוך הבורד תוביל אותנו אל הג'וב הספציפי , אם צבעו אדום(משמעות שנכשל) ונוכל לחקור. ניגש אל דוח הallure שלו , ונחקור בקלות את הנפילה של הג'וב הספציפי הזה.

לדוגמא עבור הטסט Agent נקליק עליו ונקבל את הדוח , ששם נוכל להגיע לנפילה שגרמה לטסט להכשל כולל אפשרות לצילום מסך מתוך הריצה בזמן אמת.



**קבצי FullRegression**

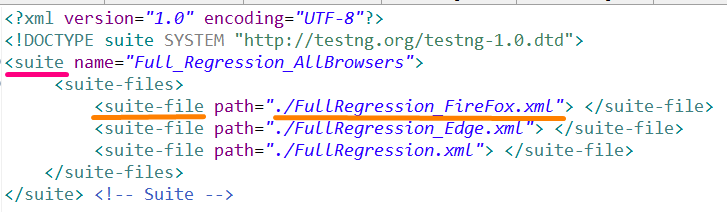
למרות שהעבודה השוטפת תהיה עם קבצי xml קטנים נרצה לשמור על קובץ מקיף וריכוזי שכולל את כל הטסטים , כדי לקבל תמונה כוללת של האוטומציה , נבנה 3 קבצים שיתנו לנו מבט בhigh-level לפי סדר גודלם.

1. **Full\_regression.xml**

יחזיק את הסוויטה שמכילה את כל הטסטים הקיימים והרצה שלו מול ג'וב מותאם בג'נקינס ייתן לנו בעצם אפשרות למבט בhigh-level על כל הטסטים של ה UIעבור דפדפן ספציפי.

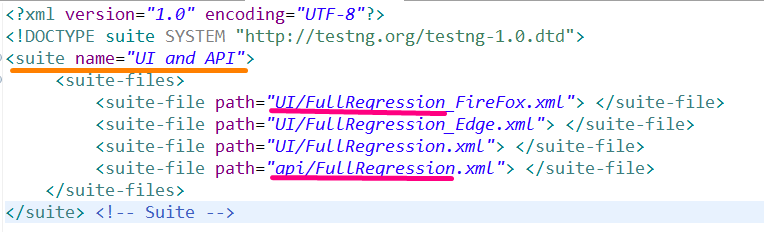
1. **Full\_regression\_AllBrowsers.xml**

קובץ שלא מכיל טסטים אלא הוא מכיל סוויטות של טסטים. ויתן לנו מבט גדול יותר על **כל** האוטומציה שלנו עבור כל הדפדפנים שאנו תומכים .

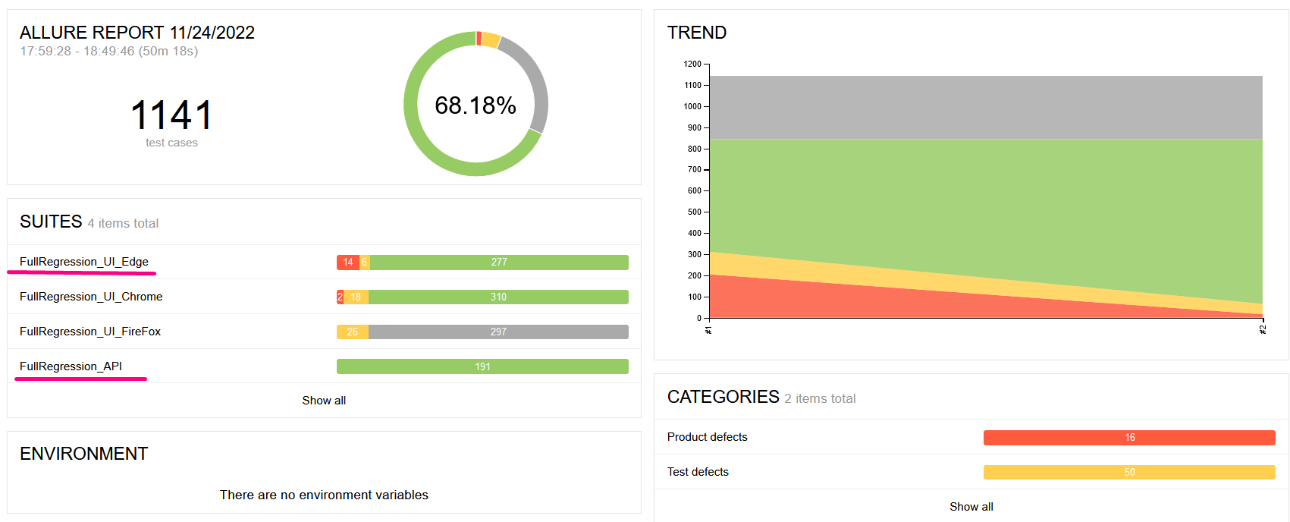


1. **UIAndAPI\_FullRegression.xml**

למעשה זה קובץ שנותן תמונה כללית ורלוונטי לפרוייקטים שמכילים בדיקות UI וגם בדיקות API



דוח allure של הרצה מסוג קובץ כזה יראה כך



לסיכום, חלוקת בדיקות למספר קבצי XML בפרויקט של Maven מביאה יתרונות רבים. זה מקל על ניהול בדיקות, מאפשר חלוקה לוגית נכונה של הרצת טסטים,מאפשר בדיקות סלקטיביות וחסכון בזמני ריצה , מקדם שימוש חוזר בבדיקות בין פרויקטים ומאפשר בדיקות מקבילות, מאפשר הרצה במקביל, אפשרות לניפוי באגים מהיר יותר, ויכולת ניצל האפשרויות הקיימות בJenkins בצורה עמוקה יותר. כל היתרונות הללו יכולים לעזור לשפר את היעילות של תהליך פיתוח האוטומציה, תחזוקה באופן שוטף, וכתוצאה מכך לאוטומציה איכותית יותר ולשחרור מהיר יותר של קוד.

לכל שאלה – ניתן לפנות אלי בלינקדאין <https://www.linkedin.com/in/nati-zadok/>