## **כלי הבדיקות: מסמך עקרונות**

## רקע

לצורך אימות עמידה בדרישות תקן FHIR בכלל, ובדרישות הרגולציה בפרט, נדרשת היכולת להריץ באופן יזום ומבוקר סט רחב ומוגדר של בדיקות. הבדיקות האלו צריכות לכסות היבטים טכניים של מימוש התקן, עמידה בהוראות הרגולציה לניוד מידע וכן היבטים של איכות וכיסוי המידע. בנוסף, נדרשת היכולת להריץ מחדש את כל סט הבדיקות ככל שיידרש, וכן אחידות מירבית בבדיקות המופעלות בארגונים שונים. על כן נדרש כי הבדיקות יורצו באופן אוטומטי ככל שניתן, תוך התבססות מקסימלית על פעולות שניתן לבצע בקוד, ויספקו פלט שיאפשר לגורם אנושי להציף סוגיות שיש לבחון או לטפל בהן, וכן להסיק מסקנות לגבי מצב העמידה בתקן וברגולציה.

בכדי לתת מענה לצרכים אלו פותח כלי ה"סרטיפיקטור" המתואר במסמך זה.

הסרטיפיקטור יהא בנוי כך שיאפשר להריץ אותו ולקבל דוח תוצאות לשימוש הגורם שיוסמך על ידי משרד הבריאות כמבקר על עמידת הארגון בהוראות הרגולציה, וכן יוכל לשמש בידי הארגון ככלי לשימוש שגרתי לבדיקות תאימות לתקן ואף בדיקות איכות נתונים (DQA).

## סוגי הבדיקות

ניתן לחלק את הבדיקות לשלוש קבוצות – בדיקות CapabilityStatement, בדיקות API ובדיקות DQA.

### בדיקות CapabilityStatement

כל שרת FHIR מחויב לספק הצהרה על היכולות אשר הוטמעו בו בפועל ע"י חשיפת משאב מסוג CapabilityStatement בנתיב המוסכם [base]/metadata.

על ה-CapabilityStatement להצהיר לכל הפחות על מימוש היכולות הנדרשות מהארגון בהתאם לסעיפי הרגולציה החלים עליו, כפי שמפורט במדריך ההטמעה (IG) של IL Core ובמדריכי ההטמעה של סלי המידע אותם הוא מחויב לממש. הסרטיפיקטור יבדוק כי השרת מצהיר ב-CapabilityStatement על תמיכה בסלי המידע בהם הארגון מחויב, יבדוק את תקינות ההצהרות וכן יוודא בהמשך כי קיימת הלימה בין ההצהרות הללו לבין הפונקציונליות של ה-API בפועל. לא ניתן לעבור בהצלחה את סט הבדיקות מבלי שהשרת יספק CapabilityStatement תקין, המונגש בנתיב הנ"ל. יודגש, אספקה של CapabilityStatement באמצעים חלופיים איננה מקובלת.

### בדיקות API

בדיקות פונקציונליות המתמקדות בפן הטכני של מימוש FHIR Restful API בהתאם לתקן ובהתאם לדרישות הרגולציה. לרוב, הצלחה בבדיקות אלו אינה תלויה באיכות ובהיקף ה-data, אלא באופן הטמעת שרת ה-FHIR בארגון. הבדיקות יופעלו על סוגי משאבים שונים ויכללו הפעלת אינטראקציות שונות מול השרת. תגובתו של השרת תיבחן בהשוואה לתוצאה המצופה בכל תרחיש, עם דגש על תקינותם של HTTP Response Codes (כולל במצבי שגיאה), HTTP Headers, וכן התוכן עצמו.

#### בדיקות לדוגמה

* בעת שליפה של Resource קיים - האם מתקבל מידע בחזרה, בפורמט המבוקש, במבנה המוגדר לפי התקן ובליווי קוד הצלחה מהשרת וערכי Header נכונים
* בעת נסיון שליפה של Resource שאינו קיים - האם מתקבלת הודעת השגיאה הנכונה מהשרת
* בעת ביצוע חיפוש - האם חלוקת התוצאות ל Pages מתבצעת בהתאם לתקן.

להרחבה ראו נספח "פירוט הבדיקות".

### בדיקות DQA (Data Quality Assurance)

בדיקות אלו מתמקדות בהיבטים איכותיים של התוכן המונגש באמצעות שרת ה-FHIR והן עוסקות במידת המוכנות של הארגון לניוד המידע שברשותו בהיבטים הבאים:

* שלמות המידע - האם הארגון מייצג את מלוא המידע הנדרש
* איכות המידע - האם המידע מיוצג באופן תקין ובעל משמעות.
* ולידציה – האם המידע עומד בדרישות תקן FHIR, הפרופילים הישראלים כפי שנקבעו ב-IL Core והאם המידע עומד בדרישות סלי המידע.

עבור בדיקות אלו יישלף מדגם של Resources מתוך שרת ה-FHIR, וייבחן אל מול קריטריונים שונים, ברובם סטטיסטיים. המדגם יישלף אך ורק באמצעות שילובים של אינטראקציות Search ו-Read מול שרת הFHIR הארגוני. לא תתבצע גישה ישירה לנתונים ממערכות מקור, וכן לא ניתן להריץ את הבדיקות על מדגם שיסופק באופן אחר כגון קבצים, שליפה ישירה מבסיסי נתונים וכו'.

להרחבה ראו נספח "פירוט הבדיקות"

## שלבים בתהליך

ריצת הסרטיפיקטור מתחלקת למספר שלבים. הפעלת השלבים השונים תיעשה באופן יזום ע"י הגורם המבקר מתוך הרשת הפנימית של הארגון.

### שלב 1 – אתחול

שלב האתחול כולל הפעלה ראשונית של הסרטיפיקטור, טעינת קונפיגורציות ובדיקת זמינות של שרת ה-FHIR. בסיום מוצלח של האתחול, ייחשף ה-API באמצעותו הגורם המבקר יתפעל את השלבים הבאים

### שלב 2 – בדיקת CapabilityStatement

כפי שמפורט בסעיף "בדיקות CapabilityStatement" במסמך זה. תוצאות הבדיקות יירשמו לקובץ. במצב של כישלון מוחלט בשלב זה לא ניתן להמשיך לשלבי הבדיקה הבאים. דוגמאות למצבי כישלון שלא מאפשרים המשך:

* השרת אינו מחזיר CapabilityStatement בכלל (מחזיר הודעת שגיאה)
* השרת מחזיר משהו שהוא איננו Json ולידי
* ה-CapabilityStatement מוחזר ואף ולידי, אך אינו מכיל הצהרה על אף יכולת רלוונטית לבדיקה (לדוגמא, אין תמיכה מוצהרת באף Resource Type)

### שלב 3 – בדיקות API

כפי שמפורט בסעיף בדיקות API במסמך זה. במהלך בדיקות ה-API יישמרו חלק מהתוצאות בקבצים לצורך שימוש חוזר בשלב בדיקות ה-DQA.

### שלב 4 – דגימת נתונים

הסרטיפיקטור יבצע דגימה אקראית מכל סוגי ה-Resources באמצעות שילוב של Search ו-Read. בסיום שלב זה, יהיו בתיקיית ה-I/O קבצי נתונים גולמיים המהווים מדגם מספק לצורך הרצת שלב 5.

בהתאם לכמות הנתונים הקיימים בשרת, למגבלות הביצועים שלו ואופן הטמעתו בפועל, שלב זה עלול לקחת זמן רב ובמהלכו תיתכן העמסה על משאבי שרת ה-FHIR. על הארגונים לקחת זאת בחשבון ולוודא התאמה של תשתית שרת ה-FHIR לעומס מסוג זה. הדגימות יתבצעו ברצף וללא הרצה מקבילית. ניתן לתאם את מועד הריצה עם הגורם הבודק כדי להימנע מריצה בשעות עומס ידועות.

### שלב 5 – Resource Validation

ביצוע ולידציה של ה-Resources שנשלפו בשלב הדגימה אל מול הגדרות התקן המלאות (Base) והיכן שרלוונטי – אל מול הפרופילים המתאימים מ-IL Core. ולידציה זו בודקת את תקינות המבנה, עמידה בכללים לוגיים (Invariants) ותקינות ערכים (טרמינולוגיה, RegEx, References וכו'), ולכן מהווה שלב ראשוני של בדיקת איכות הנתונים (DQA). הוולידציה תתבצע באמצעות רכיב צד שלישי (HL7 Java Validator) שיסופק ביחד עם הסרטיפיקטור. גם שלב זה עלול להימשך זמן רב, כתלות בכמות ה-Resources שנדגמו ובהיקף המידע המאוכלס בהם. בזמן ריצת השלב הזה, יידרשו משאבי עיבוד וזיכרון רבים מהתחנה עליה מורץ הוולידטור (אך לא משרת ה-FHIR).

### שלב 6 – DQA

בשלב זה יבוצעו בדיקות אגרגטיביות על המדגם כולו, והתוצאות יירשמו לקבצים. הגורם הבודק ישתמש בפלט של שלב זה כדי לזהות מגמות הדורשות התייחסות הן בהיבטים עסקיים (חתכי אוכלוסייה, חתכי זמן, התפלגויות נתונים וכו') והן בהיבטים של עמידה בתקן (זיהוי תבניות בעייתיות במזהים לוגיים, שימוש תקין במזהים עסקיים, התפלגויות שימוש בקודים וכדו'). בדיקות אלו ברובן אינן מחזירות תוצאת "הצלחה" או "כישלון" אלא נועדו לספק לגורם המבקר אינדיקציות מהותיות שיאפשרו לו לבחון ביחד עם הארגון אילו ממצאים הם תקינים ואילו ממצאים מעידים על בעיות נתונים.

### שלב 7 – בחינת התוצאות

ניתוח תוצאות בדיקות הסרטיפיקטור יבוצע על ידי הגורם הבודק בדשבורד ייעודי שקורא את קבצי JSON שהסרטיפיקטור מפיק, הממוקמים על השרת בו הסרטיפיקטור רץ.

סוג התוצאות בהתאם לסוגי הבדיקות ומהותן:

* בדיקות Capability statement – בדיקות אלו מפיקות תוצאות מסוג עבר/נכשל ומתמקדות בבחינת העמידה בהגדרות התקן לגבי ה Capability statement והגדרות הרגולציה לגבי תמיכה בסלי המידע הנדרשים. בדיקה לדוגמה היא האם במסגרת ה Capability statement השרת מצהיר על תמיכה בסלי המידע המצופים מהארגון והאם הוא עושה זאת בהתאם להגדרות התקן.
* בדיקות מסוג FHIR API – בדיקות אלו מפיקות תוצאות מסוג עבר\נכשל ומתמקדות בבחינת העמידה בהגדרות התקן לגבי היכולות שהשרת מצהיר עליהן. בדיקה לדוגמה הינה בחינה האם פונקציונאליות החיפוש (במידה והשרת מצהיר על תמיכה בה) מחזירה תוצאות לפי המבנה המוגדר בתקן.
* בדיקות DQA – בדיקות אלו מפיקות תוצאות סטטיסטיות לגבי התפלגויות של פריטי מידע שונים והקשרים בינם. פרשנות התוצאות מבוססת על מאפייני הארגון, האוכלוסיות השונות שבו (מטופלים מטפלים), סוגי השירותים שהוא מספק ועוד. בדיקה לדוגמה הינה התפלגות ההתמחויות של הגורמים המטפלים בארגון ובחינה האם היא בהתאם למצופה.

לאחר סיום ניתוח התוצאות הגורם הבודק יערוך טיוטת דו"ח מסכם עם מדדי הצלחה (בהתאם לקריטריונים שיקבעו) ויציג את הטיוטה לדיון ביחד עם גורמים מטעם הארגון.

בהתאם להתייחסויות שיתקבלו מטעם הארגון הגורם הבודק יעדכן את המסקנות ואת מדדי ההצלחה ויפיק דוח מסכם לשימוש משרד הבריאות והארגון.

## התקנה ושימוש

התקנת הסרטיפיקטור תתבצע על תחנה (וירטואלית או פיזית) שתוקצה לשם כך ע"י הארגון ונמצאת ברשת הפנימית שלו. ההתקנה תתבצע ע"י הארגון בליווי גורמים מקצועיים מטעם המבקר. ארגונים שירצו לבצע את ההתקנה באופן עצמאי לחלוטין יוכלו לבקש מסמך הנחיות מפורט לשם כך.

דרישות סף: מערכת הפעלה Windows או Linux, NodeJS גרסה 20 ומעלה, Java Runtime, Postman, Microsoft power BI, גישה מאובטחת מהתחנה לשרת ה-FHIR הארגוני, אפשרות גישה לתחנה מרחוק (עבור הגורם הבודק).

מומלץ לספק תחנה חזקה (מבחינת מעבד וזיכרון) על מנת לצמצם זמני ריצה של שלבים 5+6.

## מרכיבי הסרטיפיקטור

הסרטיפיקטור נכתב בשפת Node.js ומכיל את הרכיבים הבאים:

1. קובץ קונפיגורציה – מוגדר בעת ההתקנה, ומכיל הגדרות ספציפיות לארגון כגון כתובת שרת ה-FHIR, timeout וכו'
2. HTTP Client – תוכנית צד לקוח שתפקידה לתשאל את שרת ה-FHIR הארגוני
3. API – אוסף שירותי Rest באמצעותם הגורם הבודק מתפעל את הסרטיפיקטור ויוזם הרצה של תסריטי בדיקות שונים
4. קבצי קוד המכילים את תרחישי הבדיקות עצמם
5. HL7 Java Validator – רכיב ייעודי לביצוע ולידציה של FHIR Resources בהתאם להגדרות הבסיס של התקן וכן הגדרות של פרופילי IL Core. לפירוט: <https://hl7.org/fhir/validator/>
6. תיקיית I/O – במהלך שלבי הריצה השונים הסרטיפיקטור נדרש ליצור ולקרוא קבצים. הכתיבה והקריאה של קבצים מתבצעת בתיקיית ה-I/O בלבד (ממוקמת בתיקיה בה הותקן הכלי) – אין צורך לאפשר לכלי גישה לכתיבה או קריאה של קבצים מנתיבים אחרים, רכיב UI להצגה ובחינה של תוצאות הבדיקות
7. חבילות FHIR נדרשות – אוסף קבצי JSON המכילים את הגדרות חבילת הבסיס של FHIR גרסה 4.0 וכן את כל הפרופילים הרלוונטיים לתהליך הבדיקה (IL Core, סלי מידע)
8. דשבורד לניתוח תוצאות הבדיקות, מבוסס Microsoft power BI