הנחיות אבטחת מידע

לפיתוח ממשקי REST

בפרויקט שדרת המידע

גרסה 1.3

מעקב גרסאות

מס"ד	תאריך	עודכן על ידי	תיאור השינויים
1.0	2.7.2019	אלמוג חוזה	גרסא ראשונה
1.1	2.5.2020	אופיר יהב	תיקוף ועדכון המסמך. עדכון שמות מאשרי המסמך.
1.2	29.11.2020	אופיר יהב	תיקוף ועדכון המסמך. עדכון הלוגו עם המעבר למשרד הדיגיטל הלאומי.
1.3	7.11.2021	אופיר יהב	תיקוף ועדכון המסמך. עדכון הלוגו עם הקמת מערך הדיגיטל הלאומי.
1.4	7.11.2022	אופיר יהב	תיקוף ועדכון המסמך. עדכון הלוגו לאחר השינוי הארגוני.

נתוני גרסת המסמך

גורם	תפקיד	שם מלא	תאריך	חתימה
נערכה ע"י	ארכיטקט ראשי מערך ההגנה בסייבר	שרון משולם	7.11.2022	(חתימה)
נבדקה ע"י	ראש תחום מתודולוגיות הגנה בסייבר	אופיר יהב	7.11.2022	(חתימה)
אושרה ע"י	מנהל מערך ההגנה בסייבר	חגי פרלמוטר	7.11.2022	(חתימה)

תוכן עניינים

1. כללי 4

2. מונחים 5

3. דרישות אבטחת המידע 7

3.1 כללי 7

3.2 אפיון השירות 7

3.3 סכמת השירות(REST Schema) 8

3.4 URI 8

3.4.1 גרסאות 8

3.4.2 שמות המשאבים 8

3.4.3 סיומות קבצים 9

3.5 פעולות 9

3.6 הרשאות ואימות 10

3.7 פורמט שמות 10

3.8 הקשחת כותרים (Headers) 11

3.9 מבנה המידע העובר בין הלקוח לשרת 11

3.10 הודעות שגיאה 12

3.11 שימוש בפלטפורמות מוכרות 12

1. כללי

1.1. מערך הדיגיטל הלאומי מספק תשתית לשיתוף מידע בין משרדי הממשלה, בין משרדי הממשלה לעסקים במשק וכן בין משרדי הממשלה לאזרחים, בארכיטקטורת REST API. המידע משותף באמצעות מערכת רכיבי API-Gateway שמוטמעים במשרדים השונים בפרויקט "שדרת המידע".

1.2. מטרת מסמך זה הינה לפרט את הנחיות אבטחת המידע לפיתוח של צד השרת בממשקי REST API בפרויקט שדרת המידע.

1.3. מסמך זה הינו נספח למדיניות פיתוח מערכות מאובטחות של מערך הדיגיטל הלאומי. המסמך שם דגש על דרישות אבטחת המידע הייחודיות לממשקי API בשלב הפיתוח.

1.4. מסמך זה אינו כלל הנחיות לפיתוח צד הלקוח. בנוסף, אינו מפרט את הגנות התשתית הנדרשות בפרויקט שדרת המידע.

1.5. המסמך נועד למפתחי צד השרת של ממשקי API במערך הדיגיטל הלאומי ובמשרדים השונים.

1.6. המסמך נכתב, מעודכן ומתוקף על ידי ראש תחום פיתוח הגנה בסייבר, במערך ההגנה בסייבר של מערך הדיגיטל הלאומי.

2. מונחים

2.1 REST API

Representia**l** **S**tate **T**ransfer **A**pplication **P**rogramming **I**nterface
ארכיטקטורה המבוססת על מודל שרת-לקוח, כאשר השרת מחזיק את המשאבים והלקוח צורך את המשאבים ומבצע עליהם פעולות.

2.2 משאבים (Resources)

משאב מייצג אובייקט שמחזיק מידע. על המשאב ניתן לבצע פעולות כגון יצירת אובייקט חדש, מחיקה או עדכון המידע. דוגמא למשאבים בשרת המחזיק מידע ארגוני הן : "עובד", "מחלקה", "תחום" וכדומה. שמות המשאבים הינם שמות עצם בלבד, ולא שם פועל או פעולות.

2.3 אוספים (Collections)

קבוצה של משאבים. דוגמא למשאבים בשרת המחזיק מידע ארגוני: "מחלקות", "תחומים" המייצגים אוסף של משאב "מחלקה" או "תחום" (בהתאמה).

2.4 פעולות (Actions)

המבוצעות על משאב. הפעולות הן יצירה, שליפה, עדכון ומחיקה (CRUD - Create, Retrieve, Update, Delete), את הפעולות ממשים באמצעות הסוגים השונים של בקשות HTTP.

2.5 URI(Uniform Resource Identifier)

נתיב המאפשר גישה וביצוע פעולות על משאב ספציפי.

2.6 מפתח ממשק (API-Key)

מחרוזת ייחודית המתקבל לאחר הזדהות ומשמש את הלקוח על מנת לגשת למשאבים בממשקי API.

2.7 אימות (Authentication)

תהליך המאפשר לזהות באופן חד ערכי את הלקוח וההרשאות המתאימות עבורו.

2.8. **הרשאות (Authorizations)**

הפעולות שניתן לבצע והמשאבים אליהם יכול לגשת מפתח ממשק ספציפי.

2.9. **מרחב (Scope)**

אוסף הפעולות והמשאבים עליהם נותנים את ההרשאות למפתח ממשק ספציפי.

2.10. **HATEOAS (Hypermedia As The Engine Of Application State)**

רכיב ב-API שמאפשר ללקוח ללמוד את מבנה המשאבים אליהם הוא יכול לגשת בצורה דינאמית ללא צורך בשימוש במדריך. הלמידה מתבצעת באמצעות גישה ל- URI כלשהו ב-API, וה-API יחזיר כחלק מהתשובה גם את הנתיבים הנוספים תחת אותו URI שאליהם הלקוח יכול לגשת .

2.11. **רמת בשלות (Maturity Level)**

ישנן ארבע רמות בשלות של ממשק ה-API שבצד השרת:

- **Level 0** – שימוש ב- URI יחיד המייצג את כל המשאבים ושימוש בבקשות POST לביצוע כלל הפעולות בממשק.

- **Level 1** – שימוש ב- URI שונים על מנת לגשת למשאבים ולבצע עליהם פעולות ללא שימוש בסוגי בקשות HTTP לצורך ביצוע פעולות. דוגמא למימוש בשרת המחזיק מידע ארגוני:
- hxxp://organization.gov.il/departments/getAllDepartments

- **Level 2** – שימוש ב- URI שונים על מנת לגשת למשאבים וכן שימוש בבקשות HTTP שונות על מנת לבצע פעולות.

- **Level 3** – שימוש ב- URI שונים על מנת לגשת למשאבים וכן שימוש בבקשות HTTP שונות על מנת לבצע פעולות (בדיוק כמו Level 2) ובנוסף, שימוש ב- HATEOAS.

3.דרישות אבטחת המידע

3.1 כללי

דרישות אבטחת המידע במסמך זה מתייחסות למימוש ממשקי API ברמת בשלות 2 ומעלה. מימוש API ברמת בשלות נמוכה מ 2, בעייתית מבחינה תפעולית ועלולה לגרום לתקלות עקב חוסר התאמה לדרישות אבטחת המידע הנאכפים בתשתית הגישה לממשק ה- API. במידה וקיים ממשק API ברמת בשלות נמוכה מ -2 יש להתאימה לרמת בשלות 2 באמצעות רכיב ה- API-Gateway.

דרישות אבטחת המידע עבור מימוש צד השרת ה- API אינן תואמות באופן מלא לרמת בשלות 2. בכל מקרה של סתירה (קונפליקט) בין המימוש המקובל ברמת בשלות 2 לבין דרישות אבטחת המידע המצוינות במסמך זה יש להעדיף את דרישות אבטחת המידע.

3.2 אפיון השירות

יש לאפיין את השירות על ידי מסמך אפיון מוסדר במסמך האפיון יש לפרט ככל שניתן את הפרטים הבאים:

- הסבר כללי על השירות
- ייעוד המערכת
- פירוט קהל יעד וכמות משתמשים
- מטרות עיקריות
- תלויות במערכות צד שלישי
- פירוט תהליכים ופונקציות בשירות
- אפיון שדות קלט\פלט לרבות :שם השדה ,תיאור קצר ,סוג השדה וכו'
- תרשים ארכיטקטורה

3.3 סכמת השירות (REST Schema)

לכל שירות יש להגדיר סכמת מבנה נתונים שתכלול את ההגדרות הבאות:

- סוג הקלט.
- אורך
- רמת הקינון (עומק קינון) במקרה של Nested XML\JSON.
- ביטויים רגולריים.

3.4 URI

3.4.1 גרסאות

ה-URI יכיל את מספר הגרסה של ה- API מיד לאחר שם התחום (domain name) בפורמט v<major version number>.<subversion number>.

להלן שתי דוגמאות ל- URI המכיל מספר גרסה 2.0 ומספר גרסה 1.2 (בהתאמה):

- hxxp://organization.gov.il/**v2**/departments
- hxxp://organization.gov.il/**v1.2**/departments

3.4.2 שמות המשאבים

ה-URI יכיל את הנתיב למשאבים בלבד ללא תיאור הפעולה לביצוע. שמות המשאבים הינם שמות עצם בלבד ולא שם פועל. גישה למופע ספציפי של משאב תתבצע באמצעות ה-URI על ידי ציון המזהה של המופע.

להלן דוגמא ל- URI עבור ממשק המחזיק מידע ארגוני בו רוצים לגשת לנתוני עובד מסוים בצוות מסוים ובמחלקה מסוימת:

- hxxp://organization.gov.il/v1.5/departments/5/teams/3/employee/123456789

כאשר 5 זהו מספר מזהה המחלקה, 3 זהו מזהה הצוות ו- 123456789 זהו מזהה העובד.

3.4.3 סיומות קבצים

אין לציין את סיומת הקובץ באמצעות שימוש בנקודה בעת גישה למשאב המחזיר קובץ. לדוגמא:

- hxxp://organization.gov.il/v2/data

ולא:

- hxxp://organization.gov.il/v2/data.json

3.5 פעולות

פעולות על משאבים ימומשו אך ורק באמצעות שימוש בסוגי בקשות של HTTP באופן הבא:

- פעולת שליפת מידע תמומש באמצעות בקשת GET.
- פעולת מחיקה, עדכון או יצירת מופע חדש לאובייקט תמומש באמצעות בקשת POST כאשר הפעולה הרצויה והפרמטרים שלה יופיעו בפרמטרים של בקשת ה-POST.
- אין לבצע שימוש בסוגי בקשות PUT, DELETE, PATCH. בקשות אלו ממומשות ברמת ה-HTTP ולכן ניצול חולשה במימוש של אחת הבקשות יכול לאפשר לתוקף למחוק, לשנות או לחשוף מידע רגיש.
- בקשות OPTIONS מאופשרות ברמת התשתית לצרכי פעולה תקינה של API בדפדפנים, אך אין לממש אותם ברמת פיתוח ממשק ה-API.
- פעולת שליפה הדורשת שימוש בפרמטרים של HTTP תמומש באמצעות בקשת POST ולא באמצעות בקשת GET. בכל מקרה, אין לשלוח פרמטרים בבקשת GET.
- דוגמאות לפעולות שליפה הדורשות שימוש בפרמטרים הן: דפדוף (כאשר מחלקים מידע רב המוחזר משאילתה ל"דפים"), חיפוש על אוספים, מיון וסינון תוצאות החיפוש על אוספים.

3.6 הרשאות ואימות

אימות הלקוח בממשק API ממומש באמצעות פרוטוקול OAuth ע"י רכיב הAPI-Gateway באמצעות גישת הלקוח לממשק Token. לאחר אימות מול ממשק Token הלקוח מקבל חזרה שני סוגי מפתחות ממשק (API-KEY), אחד, קצר טווח, וישמש עבור הגישה לממשק והשני, ארוך טווח, שישמש עבור חידוש מפתח הממשק קצר הטווח.

כיוון שמנגנון ההרשאות והאימות ממומש באמצעות מוצר ה-API-Gateway אין צורך לממש אותו ברמת הקוד בעת פיתוח ממשקים חדשים אך יש לנהל את ההרשאות ע"י הגדרת Scope ברכיב API-Gateway.

עם זאת, בעת שימוש ב- HATEOAS, יש לוודא בצד השרת כי המידע המוחזר באמצעות HATEOAS מכיל רק את המשאבים המתאימים להרשאות של מפתח הממשק. יכולת זאת איננה ממומשת ע"י ה-API-Gateway באופן אוטומטי אלא יש להגדיר אותה ידנית.

3.7 פורמט שמות

- שמות המשאבים יהיו מורכבים מתווים אלפנומריים (אותיות וספרות) והאותיות בשמות המשאבים יהיו אותיות באנגלית בלבד.
- הסעיפים הבאים לא מכילים דרישות אבטחת מידע אך מומלצים למימוש:

1. שמות המשאבים ב- URI יכילו אותיות "קטנות" בלבד (a ולא A וכו').
2. שמות המשאבים יופיעו תמיד ברבים.
3. שמות המשאבים יהיו באנגלית ולא ב"עברית-אנגלית" (כתיבה אנגלית של מילים בעברית).
4. ההפרדה בין מילים תתבצע באמצעות מקף ("-") בלבד ללא שימוש באותיות גדולות או בקו תחתון ("_"). לדוגמא יש לרשום:

- hxxp://organization.gov.il/v2/current-user
- hxxp://organization.gov.il/v2/companies

ולא:

- hxxp://organization.gov.il/v2/current_user

- hxxp://organization.gov.il/v2/currentUser
- hxxp://organization.gov.il/v2/company
- hxxp://organization.gov.il/v2/havarot

5. בכל מקרה יש לשמור על אחידות בקונבנציה, גם במקרה של בחירת קונבנציה שונה מהמומלץ לעיל.

3.8 הקשחת כותרים (Headers)

יש להקשיח את תוכן המידע אשר עובר בהודעות ה-HTTP באמצעות הוספת הכותר "Content-Type" שיכיל את התוכן הבא עבור JSON ו- XML בהתאמה:

- Application/JSON
- Application/XML

הקשחת שאר הכותרים תתבצע ברמת התשתית ואין צורך לממש אותם ברמת הפיתוח.

3.9 מבנה המידע העובר בין הלקוח לשרת

בעת ביצוע פעולות לשליפת מידע רב, כך שהמידע המוחזר מכיל כמות גדולה של מופעים, יש לחלק את הבקשות למקטעים (עמודים) . כאשר הלקוח יפנה לממשק על מנת לשלוף את המידע הוא יפנה עם פרמטר המכיל את מספר המקטע של המידע אותו הוא מעוניין לשלוף.

הגדרת כמות המופעים המקסימאלית עד לחלוקה למקטעים תוגדר באופן דינאמי בעת אפיון השירות.

3.10 הודעות שגיאה

על השירות להחזיר הודעות שגיאה גנריות אשר אינן מסגירות מידע אודות תשתיות או מבנה המערכת או על מהות השגיאה.

3.11 שימוש בפלטפורמות מוכרות

בעת פיתוח ממשק API מומלץ להשתמש בפלטפורמות פיתוח מוכרות ולא לפתח את הממשק באופן עצמאי.