

צוהר דיגיטלי

צוהר דיגיטלי הינה תשתית אפליקטיבית למעבר קבצים בתעבורה דו כיוונית Army – אינטרנט. הכיוון הנכנס – מהאינטרנט לארמי. הכיוון היוצא – מהארמי לאינטרנט. התשתית מחולקת לעולמות תוכן שכל אחד מכיל בתוכו פרויקטים.

כיוון נכנס:

כדי לעלות קובץ בכיוון הנכנס יש להשתמש Api של צוהר דיגיטלי. "כרטיס הכניסה" למערכת מתרחש עם קריאה ל func-upload ולאחר מכן קיימים עוד מספר פונקציות שיעזרו לכם לטפל בקובץ. כל קובץ שיעלה למערכת ישויד לעולם התוכן והפרויקט באופן ישיר. **כיצד?** המפתח שתקבלו יהיה ייחודי לפרויקט שלכם. בנוסף תקבלו תחילית. תחילית מוגדרת ככל התווים הראשונים ב nameToRed ששלחתם ב func-upload עד שמופיע אחד מהתווים " _ ". והיא תשייך את הקובץ לסכמה שהוגדרה לו.

הקובץ שהעליתם יעבור תהליך שונה לפי הסוג שלו :

קבצי מידע -

הגדרה: קובץ המכיל מידע טקסטואלי בצורת רשומות.

תהליך: יישלח ישירות לארמי.

הסוגים: xml ,txt ,json

קבצי צרופה -

הגדרה: קובץ המכיל מידע בצורת תמונה.

תהליך: יעבור הלבנה והמרה לTiff ורק לאחר מכן יישלח לארמי.

הסוגים: Pdf, pptx, docx, jpg, jpeg, png, tif, tiff, bmp

בסוף התהליך הקובץ יגיע לשרת ftp של צוהר דיגיטלי בארמי שמחולק לתיקיות לפי שמות עולמות התוכן, יש לגשת לתיקייה ולשלוח משם את הקבצים שהגיעו.

כיוון יוצא:

כדי לעלות קובץ בכיוון היוצא תצטרכו לגשת לתיקיית עולם התוכן שלכם ולשים שם את הקבצים שברצונכם לעלות. כדי לדעת שקובץ הגיע לצד השחור יש לעיין בדוקומנטציה של הכיוון היוצא.

סביבות:

תשתית Api מחולקת לשני גישות :

External – שימוש מסביבת חוץ – מחשבים אזרחיים, סביבת דב Azure וסביבות מחוץ לAzure .

Internal – שימוש פנים סביבתי – סביבת הפרוד Azure .

אם הפרויקט בפיתוח אזרחי יש להשתמש בExternal .

דרישות:

שהקבצים יעברו בתשתית נדרש להגדיר את הסכמה שלהם.

אם יש לכם סכמה מוכנה יכולים לעבור לסעיף 4.

אם אתם צריכים שנייצר לכם סכמה התהליך יבוצע בכמה שלבים :

1. יש לשלוח לצוות ממשקים את האלמנטים – שם, סוג, כמות.

2. אנחנו נשלח את הסכמה להטמעה, **התהליך לוקח בין שבוע וחצי לשבועיים**.
3. יש לדגום את צוות ממשקים לאחר שבוע לבדיקת סטטוס ההטמעה.
4. לאחר שההטמעה הסתיימה צוות ממשקים מעביר קובץ משני הצדדים לבדיקת תקינות ההטמעה.
5. אם ההטמעה כשלה צוות ממשקים יבצע תחקור עם הגורמים הרלוונטיים.
אם ההטמעה צלחה תקבלו מפתח Api של צוהר דיגיטלי preprod והתחילית של הקובץ.

יש לשלוח רק קובץ אחד לבדיקה ואז בודקים עם צוות ממשקים אם הקובץ הגיע והכל תקין לאחר שהתהליך הסתיים והכל תקין תקבלו את המפתח לפרוד.

- שימו לב – הסכמה מתאימה לשני הסביבות ואין הבדל בין prod לpreprod.

אם אתם קפ"טים/רמ"דים/רש"צים פיתחנו במיוחד בשבילכם אתר שבו תוכלו לעקוב אחרי הקבצים שלכם! בקליק אחד ועם המון סוגים של סינונים אנחנו מעניקים לכם גישה לפרטים המלאים על הקבצים, הורדה שלהם וגם ניטור של הפרויקט שלכם. **אתר צוהר דיגיטלי**
אין לכם הרשאות לאתר? אל דאגה, תשלחו לנו **בטופס** את הפרטים שלכם ולמה אתם צריכים את השימוש ואנחנו נדאג לכם.

אנחנו בצוהר דיגיטלי ממליצים לכם לקרוא היטב ולעבור מסודר על כל השלבים בדוקומנטציה.

עדיין נתקעתם בבעיה?

אם אתם בתחילת פרויקט חדש וקבצים לא עוברים את התהליך מוזמנים לפנות לאביעד אלמוג. במידה והבעיה טכנית בקריאות לApi יכולים לשלוח **מייל** לצוות הפיתוח.

צוות ממשקים של אביעד אלמוג – 0525439933

בברכה,

**עידן לזר
ראש צוות צוהר דיגיטלי
054-8370965**

Api צוהר דיגיטלי:

<https://api.digital.idf.il/external/hatch> – External
<https://api-internal.digital.idf.il/internal/hatch> – Internal

כיוון נכנס:

func-upload

הדרך להעלאת קובץ למערכת הארכוב ו/או לשליחת קובץ לרשת האדומה מתחילה בפונקציית func-upload.

לאחר העלאה תקינה של קובץ למערכת הארכוב, התשובה שתקבל תכיל מזהה אותו הלקוח צריך לשמור על מנת שיוכל לבצע פעולות על הקובץ הספציפי בהמשך. פעולות שניתן לבצע על קובץ בעזרת המזהה שהתקבל כתשובה:

(1) קבלת פרטים על הקובץ (סטאטוס הקובץ, גודל, תאריך שמירה וכו') func-getFileMetadata

(2) קבלת תוכן הקובץ - func-getFile

(3) מחיקת הקובץ מהמערכת - func-delete

פעולות אלו יתוארו בהמשך.

שימו לב:

במקרה בו רוצים לעלות קובץ לשמירה בלבד במערכת הארכוב (ללא שליחת הקובץ לאדום), אין לשלוח בבקשה את השדה nameToRed.

Request

Request URL

(External/Internal)/func-upload

Request parameters

Field Name	Type		Explanation
filebase64	string	Required	תוכן הקובץ בהמרה ל-base64.
nameToRed	string	Optional	- במידה ורוצים לשלוח את הקובץ לאדום חובה להעביר שדה זה. - יש להעביר את השם <u>ללא</u> סיומת קובץ (למשל xml). - שם הקובץ צריך להיות חד חד ערכי, לא יכולים להיות 2 קבצים עם אותו שם. - חובה להעביר את השדה nameToRed שיתחיל בתחילית אשר הוגדרה לפרויקט. תחילית זה החלק היחסי משם הקובץ עד לאחד מבין התווים הבאים (לא כולל) - _ .

			<p>השם חייב להיות תואם ל-REGEX הבא: <code>"/^[a-zA-Z0-9]{1-15}[_.-]{0-14}[a-zA-Z0-9]{1-15}\$/"</code> לא יכול להגיע ריק.</p>
FileProperties	object	Required	JSON
FileProperties.FileName	string	Required	השם המקורי של הקובץ לא יכול להגיע ריק.
FileProperties.FileType	string	Required	סוג הקובץ
FileProperties.DynamicProperties	object	Optional	<p>שדה זה הינו אובייקט אשר מטרתו לשמש את מערכת הקצה אשר שולחת את הבקשה. נותן את האופציה לקשר נתונים אשר נמצאים במערכת הקצה ולייחד אותם עבור הקובץ הספציפי. (לדוג' – סוג קובץ, תאריך קליטה במערכת הקצה, מזהה מערכת וכו')</p>

Request headers

- Archive-API-Key (Type – string) – מפתח ייחודי לכל פרויקט שיתקבל בהצטרפות למערכת

Request methods

- Post

Request body

The request body.

Request example

[application/json](#)

1): Sync and archive the file ("**nameToRed**" is a required field)

```
{
  "nameToRed": "BlackDoh1In_123456",
  "filebase64": "GFDGDF",
  "FileProperties": {
    "DynamicProperties": {
      "Key1": "1",
      "Key2": "2"
    },
    "FileName": "origin-file",
    "FileType": "xml"
  }
}
```

2) Only archive the file



Responses:

200 OK – הקוד שחוזר כאשר הקובץ נשמר בהצלחה באחסון ובבסיס הנתונים במערכת הארכוב –

```
{ "FileId": "44b29af8-5555-4255-8fd3-acfdfdfdf3bf1" }
```

401 Access Denied – התשובה שחוזרת כאשר חלה בעיה בהזדהות באמצעות המפתח –

```
{ "statusCode": 401,"message": "Access denied due to missing subscription key. Make sure to include subscription key when making requests to an API." }
```

412 Precondition Failed - התשובה שחוזרת כאשר הבקשה לא מכילה את כל השדות הנדרשים -

```
{ "ErrorMessage" : "Accepted parameters are not valid",
  "ErrorCode" : 2 }
```

503/507 - Failed during upload.

```
{
  "ErrorMessage" : "Upload file failed",
  "ErrorCode" : 7
}
```

func-getFile

פונקציית func-getFile מאפשרת ללקוח לקבל קובץ אותו הוא העלה למערכת הארכוב (במידה והקובץ עדין נמצא שם).
הפונקציה תחזיר את הקובץ כחלק מ-JSON ב-Base64, בנוסף תחזיר שדות נוספים שנשמרו.

Request

Request URL

[\(External/Internal\)/func-getFile](#)

Request parameters

Field Name	Type		Explanation
fileId	string	Required	The fileId returned from the upload request.

Request headers

- מפתח ייחודי לכל פרויקט שיתקבל בהצטרפות למערכת – Archive-API-Key (Type – string)

Request methods

- post

Request body

The request body.

Request example

application/json



Responses:

200 OK - התשובה שחוזרת כאשר הקובץ נמצא בהצלחה באחסון ובבסיס הנתונים

401 Access Denied – התשובה שחוזרת כאשר חלה בעיה בהזדהות באמצעות המפתח

```
{ "statusCode": 401, "message": "Access denied due to missing subscription key. Make sure to include subscription key when making requests to an API." }
```

403 Precondition Failed - הקובץ נמחק/עדיין לא עבר סניטציה/קובץ נכשל בסניטציה

```
{ "ErrorMessage" : "Sanitation Failed", "ErrorCode" : 9 }  
{ "ErrorMessage" : "File wasn't sanitized yet", "ErrorCode" : 4 }  
{ "ErrorMessage" : "File was deleted ", "ErrorCode" : 3 }
```

404 - הקובץ לא נמצא במערכת

```
{ "ErrorMessage" : "File Wasn't Found", "ErrorCode" : 1 }
```

412 Precondition Failed - התשובה שחוזרת כאשר הבקשה לא מכילה את כל השדות הנדרשים

```
{  
  "ErrorMessage" : "Accepted parameters are not valid",  
  "ErrorCode" : 2  
}
```

func-getFileMetadata

פונקציית func-getFileMetadata מאפשרת ללקוח לקבל את ה-Data שנשמר אודות הקובץ אותו הוא העלה למערכת הארכוב.

Request

Request URL

(External/Internal)/func-getFileMetadata

Request parameters

Field Name	Type		Explanation
fileId	string	Required	The fileId returned from the upload request.

Request headers

- Archive-API-Key (Type – string) – מפתח ייחודי לכל פרויקט שיתקבל בהצטרפות למערכת

Request methods

- post

Request body

The request body.

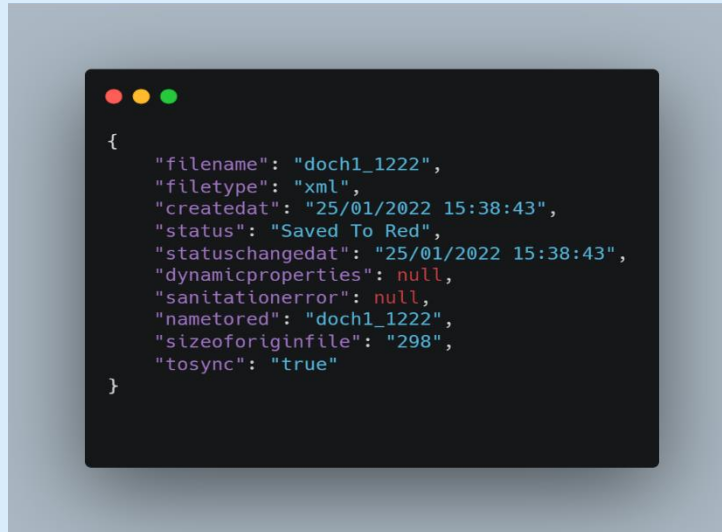
Request example

application/json



Responses:

200 OK - התשובה שחוזרת כאשר הקובץ נמצא בהצלחה באחסון ובבסיס הנתונים



401 Access Denied – התשובה שחוזרת כאשר חלה בעיה בהזדהות באמצעות המפתח

`{"statusCode": 401,"message": "Access denied due to missing subscription key. Make sure to include subscription key when making requests to an API."}`

404 - הקובץ לא נמצא במערכת



412 Precondition Failed - התשובה שחוזרת כאשר הבקשה לא מכילה את כל השדות הנדרשים

```
{
  "ErrorMessage" : "Accepted parameters are not valid",
  "ErrorCode" : 2
}
```

func-delete

פונקציית func-delete מאפשרת ללקוח למחוק קובץ אותו הוא העלאה למערכת הארכוב.

Request

Request URL

(External/Internal)/func-delete

Request parameters

Field Name	Type		Explanation
fileId	string	Required	The fileId returned from the upload request.

Request headers

- Archive-API-Key (Type – string) – מפתח ייחודי לכל פרויקט שיתקבל בהצטרפות למערכת

Request methods

- post

Request body

The request body.

Request example

application/json



Responses:

200 OK - התשובה שחוזרת כאשר הקובץ נמחק בהצלחה

```
{}
```

401 Access Denied – התשובה שחוזרת כאשר חלה בעיה בהזדהות באמצעות המפתח

```
{"statusCode": 401,"message": "Access denied due to missing subscription key. Make sure to include subscription key when making requests to an API."}
```

404 - הקובץ לא נמצא במערכת



412 Precondition Failed - התשובה שחוזרת כאשר הבקשה לא מכילה את כל השדות הנדרשים

```
{  
  "ErrorMessage" : "Accepted parameters are not valid",  
  "ErrorCode" : 2  
}
```

func-syncExistingFile

פונקציית func-syncExistingFile מאפשרת ללקוח לסמן אינדיקציית "שליחה לאדום" עבור קובץ שלא סומן כך בהעלאה הראשונה.

Request

Request URL

[\(External/Internal\)/func-syncExistingFile](#)

Request parameters

Field Name	Type		Explanation
fileId	string	Required	The fileId returned from the upload request.
nameToRed	string	Required	- במידה ורוצים לשלוח את הקובץ לאדום חובה להעביר שדה זה. יש להעביר את השם ללא סיומת. - השם חייב להיות תואם ל-REGEX הבא: "^[a-zA-Z0-9]{1-15}[_-]{0-14}[a-zA-Z0-9]{1-15}\$"

Request headers

- Archive-API-Key (Type – string) – מפתח ייחודי לכל פרויקט שיתקבל בהצטרפות למערכת

Request methods

- put

Request body

The request body.

Request example

application/json

```
{
  "fileId" : "867f5f27-491e-4ec4-9f32-026634c49a94",
  "nameToRed" : "api-test1"
}
```

Responses:

200 OK - התשובה שחוזרת כאשר הקובץ סומן בהצלחה

```
{}
```

401 Access Denied – התשובה שחוזרת כאשר חלה בעיה בהזדהות באמצעות המפתח

```
{"statusCode": 401,"message": "Access denied due to missing subscription key. Make sure to include subscription key when making requests to an API."}
```

404 - הקובץ לא נמצא במערכת



412 Precondition Failed - התשובה שחוזרת כאשר הבקשה לא מכילה את כל השדות הנדרשים

```
{ "ErrorMessage" : "Accepted parameters are not valid",  
  "ErrorCode" : 2 }
```

כיוון יוצא:

שלבים למשיכת קבצים שהגיעו מהצד אדום

הכיוון השני של המערכת הינו קבלת/משיכת קבצים שהגיעו מהצד האדום.
הקבצים נשמרים במערכת הארכוב באחסון.

התהליך מתחלק לכמה שלבים –

1. רשימת הקבצים הממתינים לשליחה
2. משיכת הקבצים
3. סימון הקבצים שנמשכו

פירוט השלבים:

1. **func-listWaitingFiles**
תהליך העברת הקבצים מצד אחד לצד השני קורה פעם בשעה.
בכדי לדעת מהם הקבצים אותם המערכת צריכה למשוך, המערכת נדרשת לקרוא לפונקציה func-listWaitingFiles. הפונקציה תחזיר את כל הקבצים שממתינים למשיכה תחת אותו פרויקט. בגלל שתחת כל פרויקט ישנן מספר תחיליות, אפשר להפריד את בבקשה לפי השדה prefixList ולמשוך רק את הקבצים של אותם תחילית.
2. **func-getFileFromRed**
הפונקציה func-getFileFromRed מקבלת FILE ID בודד בכל בקשה ויכולה להחזיר קובץ אחד בכל פעם.
הפעולה מחזירה את תוכן הקובץ בפורמט בו הוא נשמר כלומר JSON/XML.
3. **func-mark-processed**
אחרי שהמערכת קיבלה את הקבצים, היא נדרשת לסמן כי הקבצים נשמרו ביעד בהצלחה.
הפונקציה func-mark-processed אחראית על הפעולה הזו.
הפונקציה מקבלת קובץ בודד או רשימה של קבצים ומסמנת אותם כ status = processed .
אחרי שקובץ סומן כ-processed מערכת הארכוב תדע שהתהליך קבלה שלו הסתיים.

func-listWaitingFiles

Request

Request URL

[\(External/Internal\)/func-listWaitingFiles](#)

Request parameters

Field Name	Type		Explanation
days	Integer	Required	The number of days backwards to search waiting files from. Between 0 to 14 . 0 days will list all of the files that were uploaded today, 1 day will list all of the files that were uploaded today or yesterday, And so on.
prefixList	Array<string>	Optional	The prefixes of the files to search. The prefixes should not include separating char at the end. For example: ["a", "b", "c"] – valid. ["a_", "b.", "c_"] – invalid.

Request headers

- Archive-API-Key (Type – string) – מפתח ייחודי לכל פרויקט שיתקבל בהצטרפות למערכת

Request methods

- put

Request body

The request body.

Request example

application/json

```
{
  "days" : 2,
  "prefixList": ["prefix"]
}
```

Responses:

200 OK - התשובה שחוזרת נשלף המידע עבור הקבצים בהצלחה



401 Access Denied – התשובה שחוזרת כאשר חלה בעיה בהזדהות באמצעות המפתח

`{ "statusCode": 401, "message": "Access denied due to missing subscription key. Make sure to include subscription key when making requests to an API." }`

404 - הקובץ לא נמצא במערכת



412 Precondition Failed - התשובה שחוזרת כאשר הבקשה לא מכילה את כל השדות הנדרשים

`{ "ErrorMessage" : "Accepted parameters are not valid",
 "ErrorCode" : 2 }`

func-getFileFromRed

Request

Request URL

(External/Internal)/func-getFileFromRed

Request parameters

Field Name	Type		Explanation
fileId	string	Required	The unique id of the file (guid from listWaitingFiles).

Request headers

- Archive-API-Key (Type – string)

Request methods

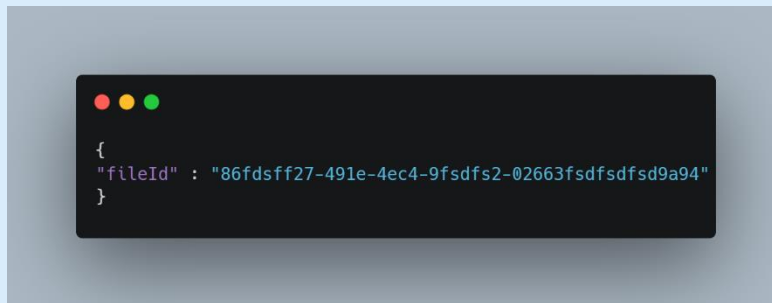
- put

Request body

The request body.

Request example

application/json



Responses:

200 OK - התשובה שחוזרת נשלף המידע עבור הקבצים בהצלחה

<? "xml version="1.0" encoding="WINDOWS-1255" standalone="no? >

....

</ns0:..>

401 Access Denied – התשובה שחוזרת כאשר חלה בעיה בהזדהות באמצעות המפתח

`{"statusCode": 401,"message": "Access denied due to missing subscription key. Make sure to include subscription key when making requests to an API."}`

404 - הקובץ לא נמצא במערכת



412 Precondition Failed - התשובה שחוזרת כאשר הבקשה לא מכילה את כל השדות הנדרשים

**`{ "ErrorMessage" : "Accepted parameters are not valid",
"ErrorCode" : 2 }`**

func-mark-processed

Request

Request URL

(External/Internal)/func-mark-processed

Request parameters

| Field Name | Type | | Explanation |
|----------------|---------------|----------|--|
| fileIds | Array<string> | Required | The unique id's of the file (guids from listWaitingFiles). |

Request headers

- **Archive-Api-Key** (Type – string)

Request methods

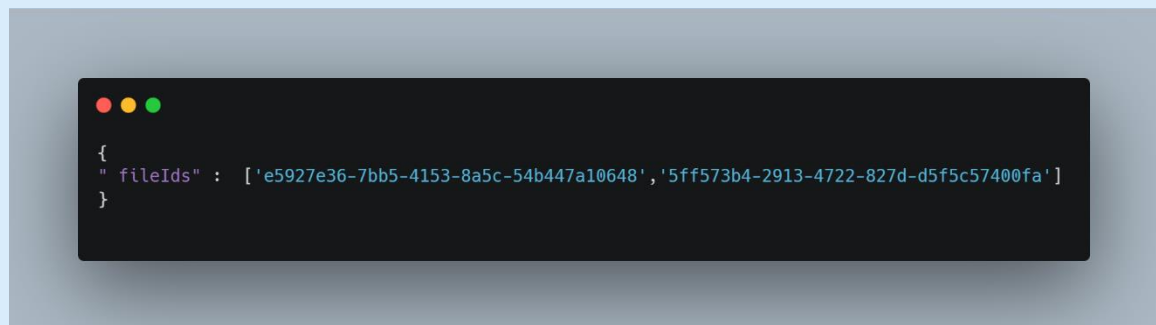
- **post**

Request body

The request body.

Request example

application/json



Responses:

200 OK - התשובה שחוזרת נשלף המידע עבור הקבצים בהצלחה

{}

401 Access Denied – התשובה שחוזרת כאשר חלה בעיה בהזדהות באמצעות המפתח –

`{ "statusCode": 401, "message": "Access denied due to missing subscription key. Make sure to include subscription key when making requests to an API." }`

404 – שגיאה בעדכון המידע במערכת



404 – סימון קובץ כהסתיים לפני שנמשך



412 Precondition Failed - התשובה שחוזרת כאשר הבקשה לא מכילה את כל השדות הנדרשים -

`{ "ErrorMessage" : "Accepted parameters are not valid",
"ErrorCode" : 2 }`