

מדריך הגדרות API בחשבשבת

תוכן עניינים

2.....	ממשק API
2.....	מה זה API?
2.....	הסבר לתהליך בחשבשבת
3.....	הגדרות - צד לקוח
4.....	הגדרות - צד ספק
4.....	הנחיות למתכנת -רישום וקבלת מזהה ספק להחתמת הנתונים
9.....	H-connect Errors
10.....	הגדרת "מסופון 2"
11.....	דוגמא לקוד קריאת API
11.....	API Post
13.....	API GET
16.....	דגשים לבניית מערכת ממשקים לאתרי סחר

מסמך זה מסביר את מערכת הממשקים לתוכנת החשבשבת **בגרסת חלונות** כולל הנחיות למתכנתים.

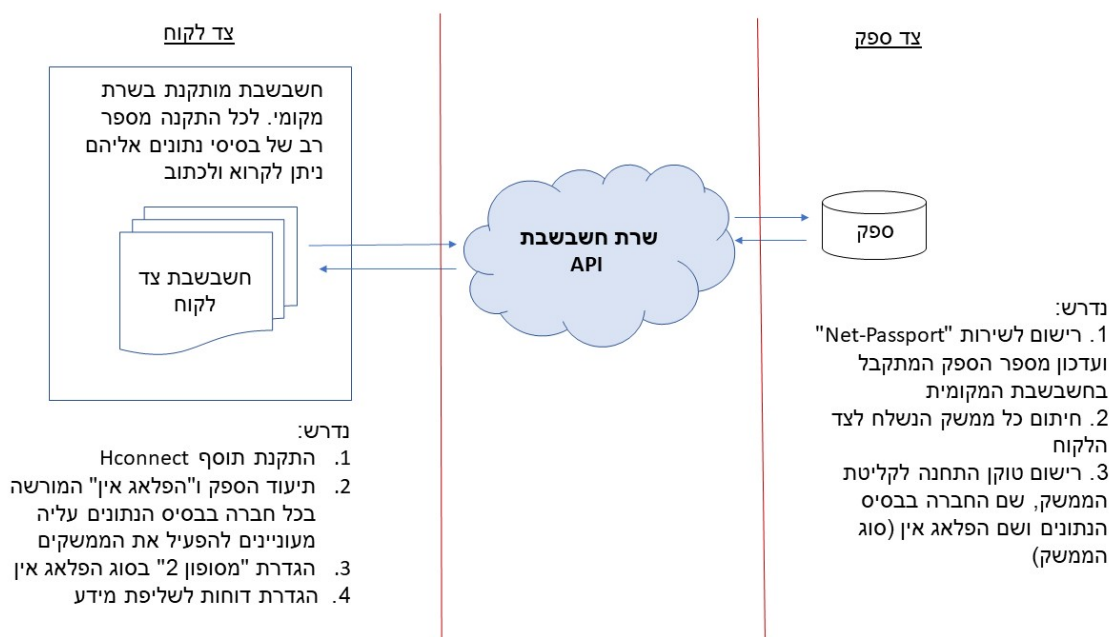
מה זה API?

API הוא ראשי תיבות של Application Programming Interface או בעברית **ממשק תכנות יישומים**. הוא כולל ספריית קוד פתוח, תקשורת נתונים ועוד, באמצעותן ניתן למשוך מידע או להתחבר למאגר מידע ונתונים של חברה אחרת. חשבשבת מאפשרת את החיבור אליה, על ידי קוד פתוח עבור הנתונים שלה ואף יצרה מאגרי מידע המסבירים את השימוש בקוד ה-API. באופן זה המתכנתים יכולים לחסוך זמן עבודה יקר. אחד היתרונות של ממשקי API הוא האפשרות להעביר מידע לחשבשבת המקומית בדחיפה דרך הדפדפן.

✓ הערה: לחילופין, ניתן להשתמש בממשקי הקבצים של חשבשבת ללא כל שינוי. הסברים על ממשקים אלה ניתן למצוא באתר חשבשבת בסעיף: מידע לבתי תוכנה.

הסבר לתהליך בחשבשבת

תוכנת חשבשבת עובדת עם שרת MS SQL server המותקן אצל הלקוח. ההתקנה מתבצעת במחשב/שרת מקומי, או בשרת בענן ה-API איננו מתקשר ישירות עם שרת הלקוח, אלא עם שרת ייעודי של חברת חשבשבת.



יש תחילה להירשם לשירות בכתובת: <https://forms.h-erp.co.il/HConnectSignContract.asp>

לאחר הרישום לשירות מתקין חשבשבת ייצור קשר לצורך התקנת התוסף שירות למתכנת במייל hatmaot@h-erp.co.il (ללא עלות)

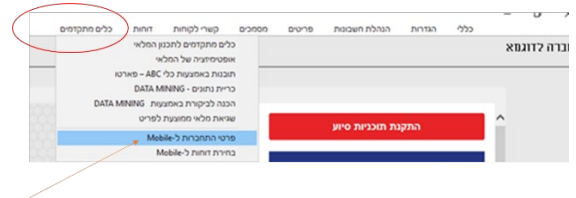
✓ תמיכה טלפונית/זום בתשלום עבור שעות הטמעה במחיר 400 ש"ח + מע"מ לשעה (או חלק ממנה)

מסמך זה מסכם את כל הנדרש למתכנתים על מנת להפעיל את ה-API ללא כל תמיכה או עזרה. יש לקרוא מסמך זה בעיון. ניתן לשלוח מייל ולציין את שם הלקוח, מספר המנוי, שם הספק (הספק הוא תוכנת צד ג' שמעוניינת להתקשר עם חשבשבת) ולצרף פירוט של בקשת התמיכה.

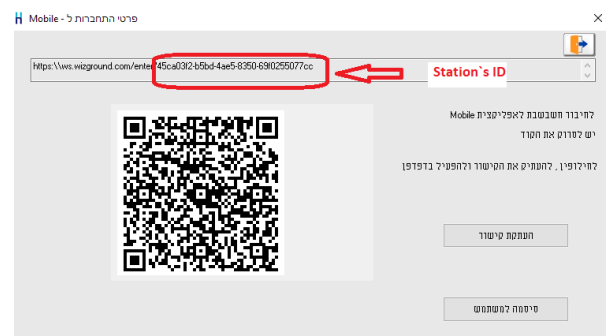
הגדרות - צד לקוח

1. Station ID

על מנת ששרת החשבשבת יזהה את התחנה שעבורה נשלחות קריאות ה-API יש להוסיף לכל קריאה פרמטר של Station, כך שניתן יהיה לזהות לאיזה לקוח אתם מפנים את קריאת ה-API. לאחר הרישום לשירות ולאחר התקנת התוסף ניתן להיכנס לתפריט - כלים מתקדמים << פרטי התחברות ל Mobile

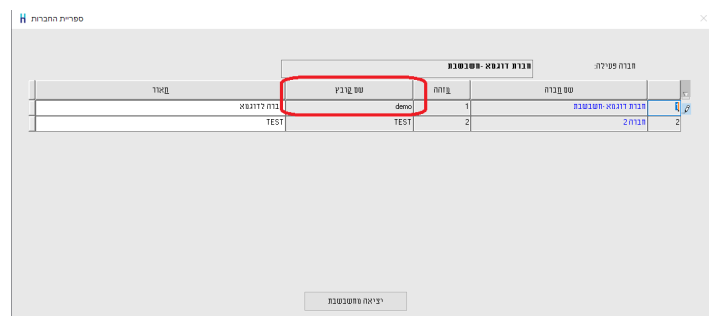


יפתח המסך הבא:



2. מזהה חברה - Company

בכל התקנת חשבשבת ניתן לפתוח מספר לא מוגבל של חברות. כדי לזהות את החברה בספריית החברות, אליה נדרשת התממשקות וקריאות ה API, יש להוסיף פרמטר הנקרא Company. בכניסה לתוכנת חשבשבת, מוצג מסך הכולל את רשימת כל החברות. יש להעתיק את שם הקובץ באנגלית המוצג בספריית החברות:



3. מזהה סוג הממשק – plugin

לפי פרמטר זה יודעים איזה סוג ממשק רוצים לקלוט. לדוגמה, כדי ליצור קבלה בתוכנת חשבשבת, יש לציין את שם הפלאגין kupain כדי לקרוא נתונים מחשבשבת יש להשתמש בפלאג אין בשם Reports. שמות הממשקים הקיימים Plugin - HESHIN חשבונות, IMOVEIN מסמכים, KUPAIN ממשק קבלות, Movein תנועות יומן, Itemin פריטים, Reports: משיכת נתונים משרת חשבשבת

✓ תוכן הממשק והשדות הנדרשים מפורטים במסמך **Plugin Data** - לחץ כאן

4. רישום סוג Plugin לקליטה לצורך אימות דו כיווני יש להעניק הרשאה לספק לבצע קריאת API מהתחנה המקומית: יש להיכנס לתפריט. **הגדרות** << **הגדרות H-Connect** ולרשום באחת מתחנות חשבשבת את הנתונים הבאים

- 4.1. **חברה:** לקליטת הממשק – חברה מספריית החברות המקומית
- 4.2. **מזהה הספק:** ה-ID של קובץ האבטחה, הנמצא בשרת הספק והתקבל מה NETPASSPORT (פירוט בהמשך).
- 4.3. **סוג ממשק לקליטה:** סוג Plugin אותו רוצים להפעיל.

מס	חברה	מזהה ספק	סוג ממשק לקליטה	מזהה חברה
1	חברה לדוגמא	623569617361764353	SignatureR	1
2	חברה לדוגמא	616231817616719873	heshin	1
3	חברה לדוגמא	616231817616719873	reports	1
4	חברה לדוגמא	616231817616719873	kupain	1
5				

הגדרות - צד ספק

לאבטחת המידע יש לבנות מנגנון דו כיווני. כלומר, לא די להגדיר את צד הלקוח ונתוני הלקוח, אלא יש להגדיר גם את צד הספק. לכן, יש להחתים את הנתונים בכל קריאת API, בין אם מעוניינים לקרוא נתונים מתוך מסד הנתונים של חשבשבת, ובין אם מעוניינים לשלוח נתונים למסד הנתונים של חשבשבת.

החתמת הנתונים מתבצע באמצעות מערכת **Net-Passport** כפי שיוסבר בהמשך התיעוד.

הנחיות למתכנת-רישום וקבלת מזהה ספק להחתמת הנתונים:

ניתן להוריד חינם מאתר <https://netpassport.io/register> קובץ אבטחה שחותם כל קריאה ובתיבה לשרת המקומי. קובץ זה חותם את הספק המורשה והנו ייחודי לספק.

הקמת ספק חדש:

ספק חדש הוא התוכנה צד ג' המבקשת לבצע קריאות API לחשבשבת המקומית. הרישום הינו ברמת הספק ולא ברמת לקוח קצה או ממשק, כלומר, נדרש רישום חד פעמי. ניתן להשתמש בהגדרות אלו ללקוחות קצה שונים ולממשקים שונים.

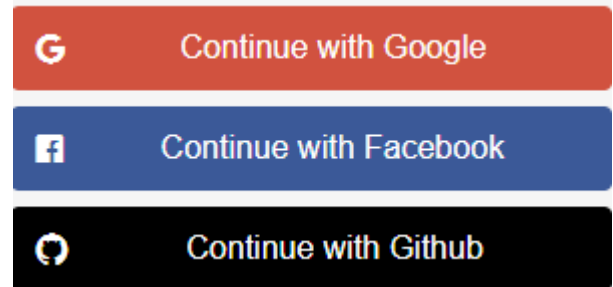
1. יש להירשם לשרות בכתובת - <https://netpassport.io>

בכניסה לאתר מוצג המסך הבא:

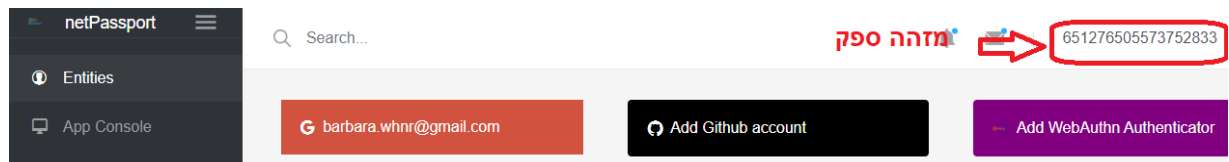


1.1 לוחצים על SIGN IN:

ואז ניתן להתחבר לשרות באמצעות שלוש האפשרויות הבאות:



בסיום הקמת החשבון מוצג מספר הספק Netpassport id בצד ימין של המסך בחלקו העליון:



מזהה ספק זה, ישמש אתכם לכל קריאות ה-API שתבצעו לתכנת חשבשבת. מספר זה יש לרשום בשדה netPassport id כפי שפורט לעיל.

1.2 – יש להוריד את שני הקבצים הבאים מ-netPassport

publickey.pem

privatekey.pem

App console → Generate key pair → Save keys & download public key

יש להוריד את קובץ KEYS.ZIP ולחלץ אותו לתיקיה.

בעת החתמת הנתונים, יש להפעיל את החתימה עם ה-netPassport id ועם ה-privatekey, ניתן להפנות את הקוד לקובץ (ע"פ הדוגמא המצורפת), או להזין את ה-privatekey בקוד עצמו

2. החתמת הנתונים:

כפי שפורט לעיל, על מנת להבטיח את אמינות הנתונים, וכדי שרק מי שמורשה לכך, יוכל לבצע קריאות API משרת הנתונים של חשבשבת, נדרש בכל קריאת ה-API להחתיים את הנתונים.

בכל קריאת ה-API החתימה משתנה בהתאם לנתונים בקובץ JSON.

החתמת הנתונים מתבצעת לאחר הגדרת מזהה ספק, (netpassport id) כפי שפורט לעיל, ע"י השוואת ה-private key ומזהה הספק, מול הנתונים הקיימים בהגדרות הספק בשרת חשבשבת.

שימו לב! יש להחתיים אך ורק את הנתונים שנמצאים תחת ה-message כפי שמופיע בדוגמא לקוד בהמשך התיעוד

להלן הסבר ודוגמא בקוד כיצד לבצע את החתמת הנתונים בכל קריאה.

בהמשך יפורט איך ניתן להחתים את הנתונים גם למי שלא עובד עם node.js או לא מעוניין להתקין ספריות של node.js בשרת.

Sign data

There are two way to sign data , use netPassport in node.js or by EXE file

Node.js

- download your private key from netpassport.io .
- \$ npm install net-passport

```
const { signer } = require("net-passport");
const fs = require("fs");
// Pass in the .pem file or a path to the file
const pk = fs.readFileSync(
  path.join(__dirname, "lib", "keys", "privatekey.pem"),
  "utf-8"
);
// OR
const pk = path.join(__dirname, "lib", "keys", "privatekey.pem")
// Initiate your message object
const message = {
  netPassportID: 112233,
  myData: "Hi there"
};
// Pass in two parameters that includes your object message (as mentioned above) and
  a private key or path to your private key
const signature = signer.sign(message, pk);
```

send the signature to the API.

Learn more about net-passport in <https://www.npmjs.com/package/net-passport>.

By exe file

download your private key from netpassport.io , and EXE file from:
for windows: <https://sws.wizground.com/files/netPassport-win.exe>
for mac: <https://sws.wizground.com/files/netPassport-macos>
for linux: <https://sws.wizground.com/files/netPassport-linux>

```
// send port as parameter to netPasport EXE and execute it

let port = 8040;
const exec = require("child_process").exec;
function createServer() {
  return new Promise((resolve, reject) => {
    var child = exec(`netPassport-win.exe ${port}`);

    child.stdout.on("data", data => {
      console.log(data);
      resolve("ok");
    });

    child.on("error", err => {
      console.log(`exe failed! , ${err}`);
      reject("err");
    });
  });
}

// send post message to http://localhost:yourPort/sign with the
// following parametrs for signing your data:
// privatekey - .pem file or a pth to the file.
// netPassportID - your net passport ID (from
// https://netpassport.io/home# on the top right).
// data - your data.
let signature = await axios.post(`http://localhost:${port}/sign`, {
  privatekey: "./privateKey.pem",
  netPassportID: "588406346880516097",
  data: JSON.stringify(modulData),
});

// to check if the signature is correct send post message to
// http://localhost:yourPort/verify with the following parametrs:
// privatekey - .pem file or a pth to the file.
// netPassportID - your net passport ID (from
// https://netpassport.io/home# on the top right).
// data - your data.
//signature - the signature you got in sing request

let verify = await axios.post(`http://localhost:${port}/verify`, {
```

```
privatekey: "./privateKey.pem",  
netPassportID: "588406346880516097",  
data: JSON.stringify(modulData),  
signature: signature.data,  
});  
console.log(verify.data);
```

 **WORDPRESS** [להסבר לחץ כאן](#) - אפשרות חתימה לאתרי WORDPRESS בתוסף נלווה -

H-connect Errors

	Error in API parameters	
<u>1</u>	Body. Station parameter is missing	חסר שדה station בגוף הבקשה
<u>2</u>	Body. Plugin parameter is missing	חסר שדה station בגוף הבקשה
<u>3</u>	Body. Company parameter is missing	חסר שדה station בגוף הבקשה
<u>4</u>	Body. Message parameter is missing	חסר שדה station בגוף הבקשה
<u>5</u>	body.message.netPassportID parameter is missing	שדה netPassportID בשדה message שבגוף הבקשה
<u>6</u>	body.message.pluginData parameter is missing	חסר שדה pluginData בשדה message שבגוף הבקשה
<u>7</u>	body.signature parameter is missing	חסר שדה signature בגוף הבקשה
<u>8</u>	your net passport is not correct	חסר אימות / נכשל Net Passport
<u>9</u>	Not in service	התחנה המבוקשת לא בשירות - Hconnect
<u>10</u>	you are not allowed to use [moduleType] module	אין הרשאה בחשבשבט למודול המבוקש
<u>11</u>	Failed to load [moduleType] module	שגיאה בטעינת המודול
<u>12</u>	No [plugin] parameters file	אין קובץ פרמטרים למודול הנבחר
<u>13</u>	Failed to validate [plugin] module	שגיאה באימות הנתונים
<u>14</u>	Batch file '[batchPath]' doesn't exist	קובץ אצווה לא קיים
<u>15</u>	Invalid JSON	JSON לא תקין
	parameters file – קובץ הפרמטרים (בתחנה)	
<u>1</u>	no field key	אין שדה field
<u>2</u>	'field' value is empty	ערך השדה field ריק
<u>3</u>	no required key	אין שדה required
<u>4</u>	'required' value is not Boolean	ערך השדה required לא בולאני
<u>5</u>	no type key	אין שדה type
<u>6</u>	invalid 'type'	ערך השדה type לא חוקי
<u>7</u>	no length key	אין שדה length
<u>8</u>	'length' value is not numeric	ערך השדה length לא תואם לסוג בשדה type
<u>9</u>	pluginData object	JSON הנתונים לא חוקי
<u>10</u>	no data key in pluginData	אין שדה data ב pluginData
<u>11</u>	your data is not an array	סוג השדה data אינו מערך
	invalid data- נתונים לא תקינים	
<u>1</u>	required field	שדה נדרש
<u>2</u>	type error , [field] is not a [field type]	סוג הערך שנשלח שגוי
<u>3</u>	length err ([field]) length ([length]) is not small equal to [type] length ([field length])	אורך הערך שנשלח שגוי

הגדרת "מסופון 2"

עד כאן הסבריים לגבי העברת מידע מהספק לשרת המקומי בצד הלקוח. נתוני הממשק נשמרים בתיקייה בשרת החשבשבת תחת `hash>>rep>>hconnect`

בשלב זה הפלאג-אין מבצע המרה לקובץ שאותו ניתן לקלוט בחשבשבת ודוחף אותו דרך "מסופון 2" לחשבשבת.

מנגנון זה הינו הדרך היחידה לקליטת ממשקים לחשבשבת והוא שומר הסף לאימות הנתונים, כך שירשמו בצורה תקינה בתוכנה.

יצירת קובץ אצווה "מסופון 2" במבנה הבא:

batch file name: `compld_interface_netPassportI`

קובץ מיובא : `c:\hash7\Hconnect\compld_interface_netPassportId_*.dat`

מבנה הקובץ: Delimited

תו מפריד: (,)

קובץ פרמטרים: *.prm

המרה לעברית: Ansi2Ansi

דוגמה:

תו מפריד	מבנה הקובץ	קוד קולט	קובץ פרמטרים (*)	קובץ מיובא (*)	אופן יבוא	סוג היבוא	חברה מייבאת
(,)	Delimit...	1	c:\hash7\hconn...	c:\hash7\hconnect\1_signaturer_*.dat	טקסט	תנוע...	demo

לפירוט ולהרחבה על תוכנית מסופון 2 לחץ [כאן](#)

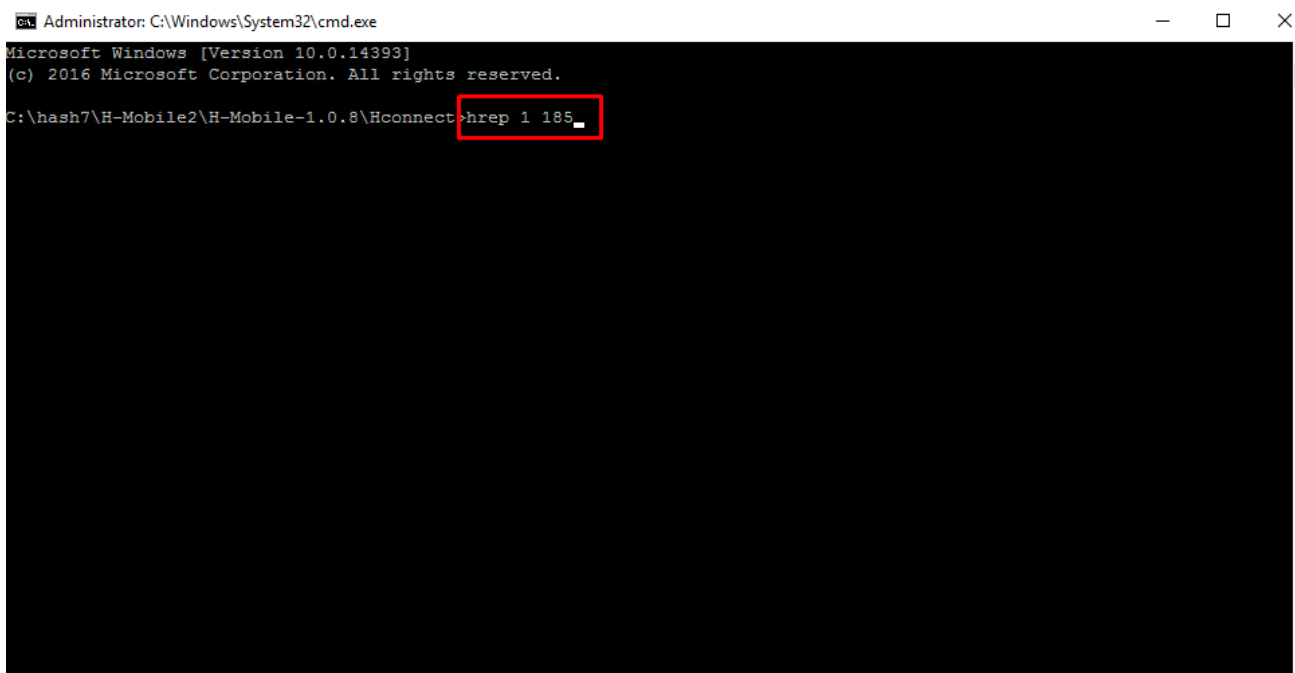
JSON KUPAIN	Imovein json:
<pre> "accountKey":"2"} ,"docNumber":"699" ,"transType":"33" ,"payDate":"13\01\2021" ,"suF":"100.00" ,"payments":"1" ,"יזיה":"creditFirm" ,"cheqsCustomerID":"123456789" ,"stockPhone":"0509876543" ,"Digits":"18804" ,"FirstPayment":"0" "stockExtraText5":"bob@gmail.com" { </pre>	<pre> }} : ,"DocumentID": "34" ,"AccountKey": "20003" ,"לייוט": "AccountName" ,"ValueDate": "16/02/2021" ,"ItemKey": "4203" ,"ענבר": "ItemName" ,"Quantity": "20.00" ,"Price": "45.00" ,"DiscountPrc": "10.00" ,"API בדיקה": "Remarks" "reference": "90009122" ,{ } ,"DocumentID": "34" ,"AccountKey": "20003" ,"לייוט": "AccountName" ,"ValueDate": "16/02/2021" ,"ItemKey": "4202" ,"מקלדת אינטרנט": "ItemName" ,"Quantity": "10.00" ,"Price": "50.00" ,"DiscountPrc": "10.00" ,"API בדיקה": "Remarks" "reference": "90009122" [{ </pre>

API GET

לצורך משיכת נתונים מהחשבשבת המקומית, ניתן ליצור דוח במחולל הדוחות ולקרוא לדוח דרך מערכת ה-API, ע"י הפעלת ה-Command prompt

בתיקייה C:\hash7\H-Mobile\H-Mobile-1.0.8\Hconnect יש לבצע את הפעולות הבאות:

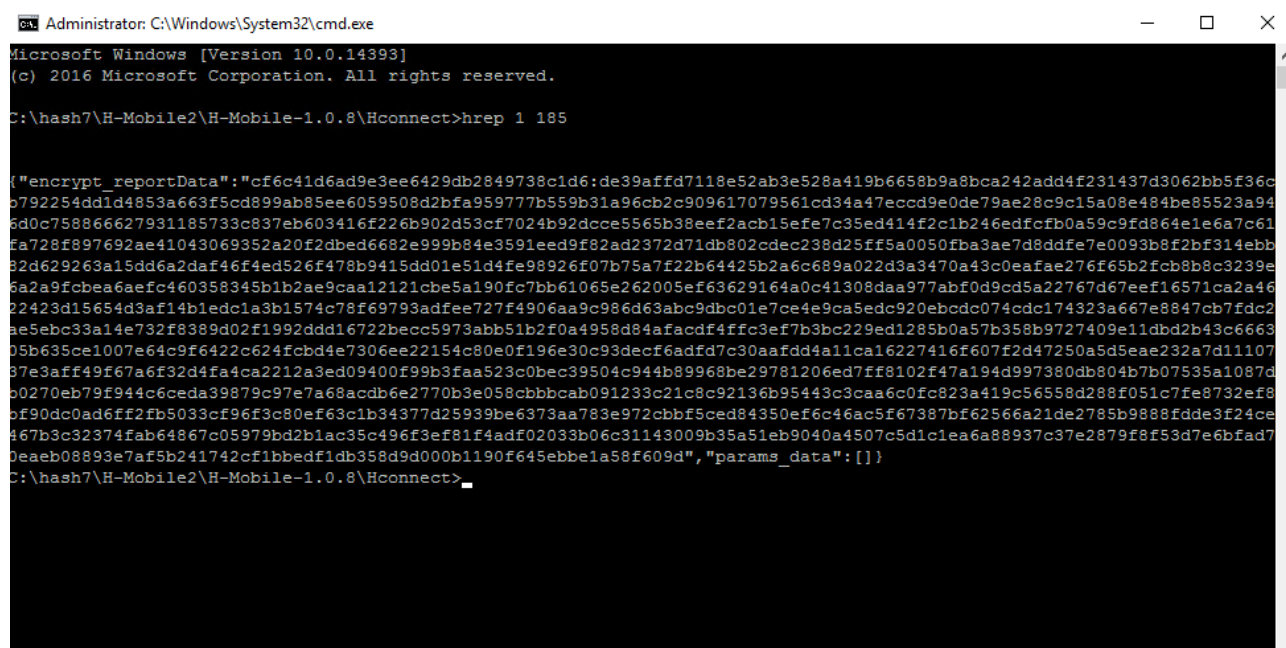
1. להפעיל את התוכנית Hrep
2. להגדיר את ה-ID של החברה ממנה רוצים למשוך את הנתונים (קוד חברה). לעזרה בנושא לחץ [בא](#)
3. להגדיר את ה-ID של הדוח שממנו רוצים לשלוף את הנתונים. לעזרה בנושא לחץ [בא](#)



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\hash7\H-Mobile2\H-Mobile-1.0.8\Hconnect>hrep 1 185_
```

לאחר מכן, יש להעתיק את השורות המוצפנות של השאילתה שנוצרה, ולהדביק אותה בקוד ה-encrypt_reportData



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\hash7\H-Mobile2\H-Mobile-1.0.8\Hconnect>hrep 1 185

{"encrypt_reportData": "cf6c41d6ad9e3ee6429db2849738c1d6:de39affd7118e52ab3e528a419b6658b9a8bca242add4f231437d3062bb5f36c
b792254dd1d4853a663f5cd899ab85ee6059508d2bfa959777b559b31a96cb2c909617079561cd34a47eccd9e0de79ae28c9c15a08e484be85523a94
6d0c758866627931185733c837eb603416f226b902d53cf7024b92dccc5655b38eef2acb15efe7c35ed414f2c1b246edfcfb0a59c9fd864e1e6a7c61
fa728f897692ae41043069352a20f2dbed6682e999b84e3591eed9f82ad2372d71db802cdec238d25ff5a0050fba3ae7d8ddfe7e0093b8f2bf314ebb
32d629263a15dd6a2daf46f4ed526f478b9415dd01e51d4fe98926f07b75a7f22b64425b2a6c689a022d3a3470a43c0eafae276f65b2fcb8b8c3239e
6a2a9fcb6a6aefc460358345b1b2ae9caa12121cbe5a190fc7bb61065e262005ef63629164a0c41308daa977abf0d9cd5a22767d67eef16571ca2a46
22423d15654d3af14b1edc1a3b1574c78f69793adfee727f4906aa9c986d63abc9d9bc01e7ce4e9ca5edc920ebcd074cdc174323a667e8847cb7fdc2
ae5ebc33a14e732f8389d02f1992ddd16722becc5973abb51b2f0a4958d84afacdf4ffc3ef7b3bc229ed1285b0a57b358b9727409e11dbd2b43c6663
05b635ce1007e64c9f6422c624fcbd4e7306ee22154c80e0f196e30c93decf6adfd7c30aafdd4a11ca16227416f607f2d47250a5d5eae232a7d11107
87e3aff49f67a6f32d4fa4ca2212a3ed09400f99b3faa523c0bec39504c944b899688be29781206ed7ff8102f47a194d997380db804b7b07535a1087d
b0270eb79f944c6ceda39879c97e7a68acdb6e2770b3e058cbbcab091233c21c8c92136b95443c3caa6c0fc823a419c56558d288f051c7fe8732ef8
bf90dc0ad6ff2fb5033cf96f3c80ef63c1b34377d25939be6373aa783e972cbbf5ced84350ef6c46ac5f67387bf62566a21de2785b9888fdd3f24ce
467b3c32374fab64867c05979bd2b1ac35c496f3ef81f4adf02033b06c31143009b35a51eb9040a4507c5d1c1ea6a88937c37e2879f8f53d7e6bfad7
0eae08893e7af5b241742cf1bbef1db358d9d000b1190f645ebbe1a58f609d", "params_data": []}
```

```

const { signer } = require("net-passport");
const fs = require("fs");
const axios = require("axios");
var exec = require("child_process").exec;

//let modulData = [{-accountKey: "12345", -docNumber: 80, -transType: 33, -suF: 34.54, -creditFirm: "ויזח"-}];
//modulData = [{-accountKey: "12345", -exFile: "\\file\\1.png", -EXPIC: "\\file\\2.pdf", -VatExampt: 0-}];

const modulData = {"encrypt_reportData": "5260c06a6ade889dc227cbef7708d82e:62aa9393ad3dd7cc2f0985145b1e411c62b2016d9c5218d01e24f656f86c"};

let port = 8040;

function createServer() {
  return new Promise((resolve, reject) => {
    var child = exec(`netPassport-win.exe ${port}`);

    child.stdout.on("data", data => {
      console.log(data);
      resolve("ok");
    });

    child.on("error", err => {
      console.log(`\n\t\tERROR: spawn failed! (${err})`);
      reject("err");
    });
  });
}

```

Reports JSON:

field	remarks	required	
encrypt_reportData		✓	מפתח לקוח
params_data		✓	מספר קבלה

- The data in this file include two parts:
 - "**encrypt_reportData**" – a long encrypted string that refers to the report structure.
 - "**params_data**" - if the report was designed to be issued according to parameters, this part lists the parameters that you have to enter for a proper export of the data.
- If you do use parameters you should update one property value only on each property object. The property "defVal" includes a default value for this property and this is the place to change it to the value that you want. All other properties should be modified according to their type. For example for report (below) having sort codes 100 - 399 you should change to "100" instead "-999999999" and 399 instead "999999999".
 - Date type parameter is in mm/dd/yy format.

Jason Example with "client" changes:

```
encrypt_reportData =
'Oc61e292d66a71326e011b089534ee1.....4b4312a84bac14e0ab520a0361d9bcd090';

params_data=[
  { "p_name": "__MUSTACH_PO__",
    "id": "0",
    "type": "long",
    "name": "חתך/שדה מרשומת החשבון/1. פרטי חשבון/קוד מיון",
    "defVal": "100",
    "opName": "מ.מ.עד",
    "opOrigin": "from"
  },
  { "p_name": "__MUSTACH_P1__",
    "id": "500",
    "type": "long",
    "name": "חתך/שדה מרשומת החשבון/1. פרטי חשבון/קוד מיון",
    "defVal": "399",
    "opName": "מ.מ.עד",
    "opOrigin": "to"
  }
];
```

✓ חשוב: על מנת לקרוא נתונים על פי פרמטרים לפי התיעוד לעיל, יש ליצור דו"ח בחשבשבת, עם הפרמטרים הנדרשים. אם הדוח לא נבנה בחשבשבת באופן שיאפשר שינוי פרמטרים, לא ניתן להזין זאת בקוד.

דגשים לבניית מערכת ממשקים לאתרי סחר סנכרון אתר האינטרנט הוא בעל חשיבות משתי סיבות:

1. שיקוף הנתונים הקיימים בתוכנת חשבשבת ERP אל מול הלקוחות (מלאי זמין למכירה, מחיר המכירה ללקוח קיים וכו').
2. חיסכון בכוח אדם ויצירת מנגנון אוטומטי בין האתר לבין מערכת המידע.

הפונקציות אותן ניתן להטמיע בין המערכות באמצעות מערכת הממשקים:

1. **סנכרון כרטיסי חשבון:** אתרי B2B ידרשו סנכרון חשבון בין חשבשבת לבין האתר. מכיוון שניהול הלקוח מתבצע בחשבשבת (ניהול האובליגו והאשראי), יש חשיבות להקמת כרטיסי הלקוחות בחשבשבת ובסנכרון חד כיווני לאתר.
 2. **סנכרון כרטיסי פריטים:** הפריטים מוקמים בחשבשבת. חשוב מאד לנצל את האפשרויות בתוכנה לניהול מלאי מוקפד, וליצירת הזמנות רכש, תוך שימוש בכלים המתקדמים לתכנון הרכש והאופטימיזציה של המלאי. סנכרון הפריטים מתבצע בתדירות הרצויה, פעם בלילה או מספר פעמים ביום. הסנכרון הוא חד כיווני לאתר האינטרנט. דוח הפריטים יכיל את הפריטים הקיימים באתר וכן את המלאי הזמין למכירה באתר.
- ### מהו מלאי זמין?
1. מלאי זמין מתייחס ליתרת הפריט ממחסן האתר, אחרי גריעת צבר ההזמנות. ניתן לרשום בשדה **נתון נוסף** בפריט כמות מינימום, ולהתחשב בנתון זה בנוסחה לחישוב המלאי הזמין.
 3. **סנכרון מחירוניהם:** חשבשבת מאפשרת לקבוע מחירים ללקוחות בדרכים מגוונות. לדוגמא, ניתן לקבוע מחיר לפי כמות, או להחיל על המחיר מבצע כלשהו. יש להעביר את המחירון הרלוונטי מהמשרד האחורי לאתר, ולוודא שבקבוצת הפריטים הנמכרים באתר, אין משתמשים במשרד האחורי במחירים תחת תנאים מורכבים (שאם לא כן, המחיר באתר עלול להיות שונה מהמחיר במשרד האחורי) יש לוודא שמחיר המכירה העובר לאתר אינו כולל מע"מ, וגם הממשק החוזר אינו כולל מע"מ במחיר. על אף זאת, חובה להציג באתר קמעונאי את המחיר הכולל מע"מ.
 4. **ממשק הזמנות:** הזמנות מלקוחות מתקבלות בערוצים השונים: בהקלדה משיחת טלפון, באמצעות מסופוני סוכנים, מאתרי סחר וגם מאתר האינטרנט. כדאי לסמן את ההזמנות מהאתר כדי שניתן יהיה לפלח את ההזמנות לפי ערוצים השונים. קיימות שתי אפשרויות לקליטת הזמנות: למאגר זמני או למאגר קבוע. אם ההזמנות המקוריות הופקו באמצעות תוכנה אחרת (או בפנקס ידני) יש לקלוט את ההזמנות באמצעות המסמך **הזמנת סוכן**. ההזמנה צריכה לכלול את נתוני המוצרים (פריטים) כמויות, מחירים (ללא מע"מ), סוג משלוח, פרטי הלקוח, הערות וכו'.
 5. **ממשק קבלות:** אחד הנושאים הקריטיים בניהול אתר אינטרנט המבצע סליקה של כרטיס אשראי הוא הפקת קבלות. לפי הוראות רשות המסים **יש להפיק קבלה מיד עם סליקת כרטיס האשראי**. חשבשבת מאפשרת קליטה של הקבלה ממערכת הסליקה החיצונית ושליחת הקבלה ללקוח במייל. נתוני התשלום ייקלטו במאגר זמני של הקבלות, ועל המשתמש להפיק את הקבלה (או מספר קבלות ברצף) **באופן יזום**. הקבלות יופקו וישלחו למייל אותו רשם הלקוח כאשר ביצע את ההזמנה באתר.
 6. **חשבונית אוטומטית:** הזמנת סוכן הנכנסת מממשק יכולה להפוך, באופן אוטומטי, לחשבונית מס. פעולה זו מתאימה ללקוחות שהמוצר שהם רוכשים אינו פריט מלאי ואין תהליך ליקוט, אריזה ומשלוח, כך שהמוצר סופק במלואו בזמן הרכישה באתר. אם רוצים לממש אופציה זו נדרשת תמיכה ממחלקת הטמעות.
 7. **פקודות יומן:** באתרים מסויימים מפיקים חשבוניות מס וקבלות בתוכנות אחרות, ומעוניינים לממשק לחשבשבת אך ורק את נתוני פקודות היומן (הנהח"ש בלבד). אם רוצים לנהל את המלאי בחשבשבת, יש להעביר ממשק **נוסף** - של תנועות המלאי.