

מהו API?

API ראשי תיבות של Application Programming Interface – ממשק תכנות יישומים.
אז נתחיל בלהבין מהו ממשק.

ממשקים

לכל מכשיר שאנו משתמשים בו יש ממשק כלשהו. למיקרוגל יש מספרים וכפתור הפעלה, בעוד שלמתג אור יש ממשק אפילו יותר פשוט.

אנו משתמשים בממשקים אלה כדי לגרום למכשיר לעשות את הדבר שאנו רוצים. אנחנו לא צריכים להבין את מעגלי החשמל ואת המדע כדי לחמם אורז ושעועית. אנחנו צריכים רק להשתמש בממשק שנחשף לנו.

דוגמא נוספת אם נשווה את פעולתו הפנימית של מנוע המכונית ביחס לדרך שאנו מתממשקים אליו.



החלק הפנימי - מורכב

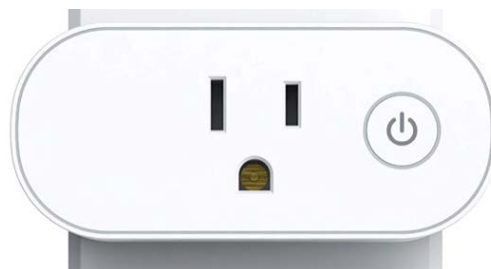


הממשק - פשוט

כל המורכבות הפנימית מופשטת כלפי חוץ ומשאירה את המשתמש עם הממשק האפשרי הפשוט ביותר.

הפשטה - Abstraction

ממשקי API מספקים שכבת הפשטה עבור המשתמש.
הפשטה מסתירה הכל מלבד מה שרלוונטי למשתמש, מה שהופך אותו לפשוט לשימוש.
בתקע של בית חכם כמו זה למשל, יש מורכבות רבה המתרחשת בפנים.



המכשיר צריך להתחבר לרשת ה-WiFi ולתקשר עם המכשירים החכמים האחרים שאומרים לו להידלק ולהיכבות.
עם זאת, המשתמש מקבל רק כפתור הפעלה.
כל המורכבות הפנימית מופשטת, מה שהופך את השימוש בה לפשוט ופשוט.
הפשטה - abstraction הוא מושג שנראה לעתים קרובות בתכנות, כך שזה מועיל להבין אותו היטב.

ה-API ב-API

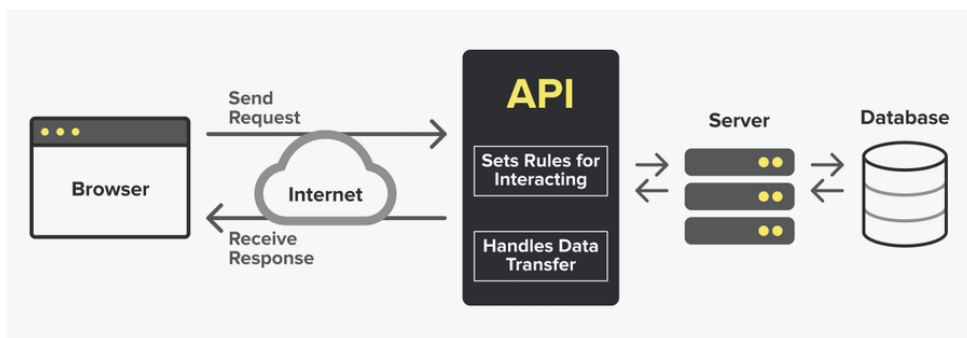
כעת, כשאנחנו יודעים את משמעות החלק - "ממשק" (Interface),
החלק של "תכנות יישומים" (Application Programming) קל יותר להבנה.

API הוא האופן שבו יישומים (אפליקציות) מדברים אחד עם השני.

לכל תוכנה שניתן ליצור איתה אינטראקציה באמצעות קוד יש צורה כלשהי של API, כך שניתקל במונח API במקומות רבים.
כאשר מפתחי אתרים מדברים על "להגיע ל-API", הם בדרך כלל מתכוונים לשירות אינטרנט המאפשר לשלוח בקשות ולקבל נתונים כתגובה.
בד"כ לשירות API יהיה תיעוד דרכו ניתן ללמוד ולהבין כיצד להשתמש ב-API.

HTTP Web APIs

אנו, מפתחי אתרים נעבוד לרוב עם ממשקי HTTP Web API. אלו הם ממשקי API שיושבים בין הקוד שלנו לבין נתונים או פונקציונליות מסוימת בשרת שאליו נרצה לגשת.



ה-API עושה בדרך כלל שני דברים:

ראשית, הוא **קובע כללים לאינטראקציה** איתו.

ה"כללים" הם שה-API אומר: "אם תבנה את הבקשה שלך בצורה כזו, אני אשלח לך בחזרה נתונים בצורה כזו." אם לא תבנה את הבקשה שלך בצורה שה-API מצפה, ה-API לא יידע מה אתה רוצה, ותקבל שגיאה בתגובה.

ה-API **מטפל גם בהעברת נתונים** בין השרת לקוד המבצע את הבקשה. ה-API הוא תוכנית הפועלת כמתווך בין אפליקציית web ובין השרת ומסד הנתונים.

ברגע שהוא יקבל בקשה חוקית, הוא יפעיל פונקציה (או מספר פונקציות).

זו המורכבות שה-API מפשט עבור המשתמש.

בהתאם למה שנבקש, ה-API עשוי להחזיר תמונה, נתונים מסוימים או פשוט להודיע שהוא קיבל בהצלחה את הבקשה.

כעת ניגע בכמה מושגים שכדאי להבין כשעובדים עם ממשקי API של HTTP.

Endpoints – נקודת קצה

ממשקי API מספקים נקודת קצה או כתובת URL ספציפית שבה נחשפים הנתונים או הפונקציות הרצויות. עבור ממשק ה-API של Unsplash (אתר של תמונות), ניתן לגשת לתמונות דרך נקודת הקצה שלהן בכתובת <https://source.unsplash.com/> ולהוסיף פרמטרים של השאילתה שלנו לאחר הסלאש האחרון.

Authentication

ממשקי API מסוימים דורשים להירשם לחשבון או לקבל מפתח ייחודי כדי לגשת למידע שלהם. זה יכול להיות כדי לאבטח נתונים, למנוע שימוש לרעה בשירות, או בגלל שהם רוצים לגבות תשלום עבור הנתונים.

אם נרצה לשנות את הנתונים שלנו במסד הנתונים באמצעות API, נדרוש לבצע אימות. אנו לא נרצה לתת לאף אחד אחר את היכולת לערוך או למחוק את הקבצים והנתונים.

עם אימות, נעביר ל-API מפתח סודי אשר יזהה משתמש ספציפי או בקשה של האפליקציה. לאחר מכן השרת יכול לקבוע אם ניתן לגשת לנתונים או לא.

אם API דורש אימות, בטוח שיהיה קיים תיעוד של ה-API אשר יסביר איך זה עובד ואת אופן האימות.

HTTP Verbs

עם כל בקשת HTTP שנוצרת, תמיד יש 'HTTP Verb' שמתחבר עם הבקשה. הנפוצים ביותר הם `GET`, `POST`, `PUT`, `DELETE`.

כאשר אתרים מבקשים נתונים מהשרת, זו בדרך כלל בקשת `GET`. בקשות `POST` ו-`PUT` משמשות לשינוי או הוספת נתונים, ו-`DELETE` מוחק משאב (resource) ספציפי.

תפקידי המפתחים

מפתחי Front-End, במסגרת עבודתם יקבלו endpoint של API על ידי מפתחי ה-Back-End, יחד עם הנחיות כיצד להגיש בקשות ולמה לצפות בתגובה.

מפתחי Back-End, תפקידם לעצב ולממש את ממשקי ה-API שמפעילים פונקציות ושאלות מול מסד הנתונים. הם יצטרכו לספק למפתחי ה-Front-End תיעוד ברור על אופן הפעולה של ה-API.

וכמובן שמפתחי Full Stack יידרשו לטפל בשני החלקים.

עבודה עם JSON

רוב ממשקי ה-API של HTTP שבהם נשתמש יקבלו ויחזירו נתונים בפורמט JSON.

JSON שומר את הנתונים שלו בצמדי key : value.

בדוגמא הבאה ישנו JSON שאנו מקבלים בחזרה כאשר אנו מבקשים נתונים מ-API Star Wars.

אם נבקש כתובת אתר זו: <https://swapi.dev/api/people/5/>

נקבל response עם ה-JSON הבא:

```
{
  "name": "Leia Organa",
  "height": "150",
  "mass": "49",
  "hair_color": "brown",
  "skin_color": "light",
  "eye_color": "brown",
  "birth_year": "19BBY",
  "gender": "female",
  "homeworld": "http://swapi.dev/api/planets/2/",
  "films": [
    "http://swapi.dev/api/films/1/",
    "http://swapi.dev/api/films/2/",
    "http://swapi.dev/api/films/3/",
    "http://swapi.dev/api/films/6/"
  ],
  "species": [],
  "vehicles": [
    "http://swapi.dev/api/vehicles/30/"
  ],
  "starships": [],
  "created": "2014-12-10T15:20:09.791000Z",
  "edited": "2014-12-20T21:17:50.315000Z",
  "url": "http://swapi.dev/api/people/5/"
}
```

ניתן לראות את קשרי הגומלין בין המפתח לערך.

למפתח "שם" יש ערך של "Leia Organa".

אנו יכולים להשתמש באובייקט זה בקוד ה-JavaScript שלנו כדי להציג את המידע שאנו בוחרים או אפילו לבצע בקשות API למעקב.

אם היינו מבקשים את <https://swapi.dev/api/people/6/>, המפתחות (שם, גובה, מסה) היו נשארים

זהים, אבל הערכים (Leia Organa, 150, 49) ישתנו.

JSON הוא פורמט קל משקל (מפני שהוא string) וניתן להשתמש בו ב-PHP, Python, JavaScript וכל שפה אחרת שבה ניתן להשתמש באינטרנט.