



יצירת עוזר דרך ה-API של OpenAI, אתה יוצר עוזר וירטואלי שמותאם לצרכים ולפונקציונליות שאתה רוצה:

יצירת העוזר:

1. **מודל (Model):** זהו היסוד הבסיסי של העוזר. אתה בוחר מודל מהרשימה הזמינה (כגון GPT-4), שישמש כבסיס ליכולות של העוזר.
2. **שם (Name):** זהו שם העוזר, שיכול להיות כל טקסט שתבחר, עד 256 תווים.
3. **תיאור (Description):** תיאור מפורט של העוזר, מה הוא עושה, ואיך הוא יכול לעזור.
4. **הוראות (Instructions):** זהו טקסט המנחה את העוזר איך לפעול, לדוגמה, "השב בקצרה" או "השתמש בנתונים מסוימים".
5. **כלים (Tools):** כאן אתה יכול להוסיף כלים שונים, כמו מתרגם קוד, כלי לחיפוש מידע, או פונקציות מותאמות אישית. כל כלי מביא יכולת נוספת לעוזר.
6. **קבצים (File IDs):** אם יש לך קבצים שתרצה שהעוזר יוכל להשתמש בהם, אתה יכול לצרף אותם כאן.
7. **מטא-דאטה (Metadata):** זהו מקום לאחסון מידע נוסף בפורמט מפתח-ערך. לדוגמה, יכול להיות מידע על התחום שבו העוזר מתמקצע.

ביצירת העוזר, אתה יוצר ישות דיגיטלית שתוכל להשתמש בה למגוון פעולות, מתשובה על שאלות ועד לביצוע פונקציות מורכבות.

יצירת נושא והרצה:

יצירת נושא והרצתו מאפשר למשתמש לשלוח פקודה לעוזר וירטואלי שיבצע משימה במסגרת דיאלוג. בפקודה זו, המשתמש מגדיר את זהות העוזר, הודעות התחלתיות לנושא, ומידע נוסף אם נדרש. כל פרמטר בקריאה משמש להגדיר ולהורות לעוזר כיצד לבצע את המשימה.

יצירת נושא (Thread) והרצתו (Run) בבקשה אחת ב-OpenAI. המידע מתאר את גוף הבקשה עם הפרמטרים הבאים:

- **assistant_id**: מזהה חובה של העוזר שיבצע את הריצה.
- **thread**: מכיל הודעות ליצירת דיאלוג, כאשר כל הודעה כוללת:
 - **role**: תפקיד השולח (למשל 'user').
 - **content**: תוכן ההודעה.
- **file_ids**: מזהים של קבצים קשורים לנושא.
- **metadata**: מידע נוסף בפורמט מפתח-ערך שניתן לצרף לאובייקט.
- **model**: מזהה של מודל ספציפי לעיבוד הבקשה, אם מסופק משנה את המודול המוגדר.
- **instructions**: הוראות לעוזר, שיכולות לשנות את התנהגותו.
- **tools**: כלים ספציפיים שהעוזר יכול להשתמש בהם במהלך הריצה.

השימוש ב-API זה יוצר תקשורת ממוקדת ויעילה עם העוזר, מאפשר להתאים אותו לצרכים מסוימים ולהשיג תוצאות מהירות ומדויקות.

תגובת api:

- **id**: מזהה ייחודי של הריצה.
- **object**: סוג האובייקט, במקרה זה "thread.run".
- **created_at**: חותמת זמן של יצירת הריצה.
- **assistant_id**: מזהה העוזר המשתתף בריצה.
- **thread_id**: מזהה הנושא של הדיאלוג.
- **status**: מצב הריצה, למשל "queued" (בתור).
- **started_at**: זמן התחלת הריצה, אם החלה.
- **expires_at**: זמן פקיעת הריצה.
- **cancelled_at**: זמן ביטול הריצה, אם בוטלה.
- **failed_at**: זמן כישלון הריצה, אם נכשלה.
- **completed_at**: זמן סיום הריצה, אם הושלמה.
- **last_error**: מידע על השגיאה האחרונה, אם קיימת.
- **model**: המודל שנמצא בשימוש.
- **instructions**: הוראות לעוזר.
- **tools**: רשימת כלים שנמצאים בשימוש, כל אחד עם פרטים נוספים.
- **file_ids**: מזהי קבצים קשורים לריצה.
- **metadata**: מידע נוסף בפורמט מפתח-ערך.

בדיקת סטטוס ריצה:

להשיג מידע על הריצה שנעשתה על ידי עוזר. היא מציינת את זהות הנושא (thread) ואת זהות הריצה (run), ומצפה לקבל פרטים על מה שקרה במהלך אותה ריצה.

- **thread_id (חובה):** המזהה של הנושא (thread) שאתה רוצה לאחזר ממנו הודעות.
- **run_id (חובה):** המזהה של הריצה.

מספקת תכונה שמאפשרת לך לעקוב אחרי סטטוס הריצה של עוזר דיגיטלי. סטטוס הריצה מספק מידע חשוב על מצב הבקשה בזמן אמת, כולל אם היא נמצאת בביצוע, הושלמה, בוטלה, נכשלה או פג תוקפה. כאשר ריצה מחכה לפעולה ("requires_action"), זה אומר שיש צורך בקלט נוסף מהמשתמש, כמו לשלוח תוצאות מכלי מסוים שנקראו במהלך השיחה. זה חיוני להבין כי התהליך מחכה לפעולות משתמש כדי להמשיך את הביצועים, ולא ימשיך באופן אוטומטי ללא זה.

- **In Progress, Cancelled, Failed, Completed, Expired:** מצבים שונים של הריצה, מהתהליך עד לסיומו.
- **Submit Tool Outputs- Requires Action:** התהליך דורש פעולה כגון שליחת תוצאות מכלים.

המאפיין "status" באובייקט "run step" מייצג את מצב הביצוע של השלב הנתון בתהליך הריצה. הוא יכול לקבל אחד מהערכים הבאים:

in_progress: השלב נמצא בתהליך ביצוע. זה אומר שהפעולה שמבוצעת בשלב זה עדיין לא הושלמה.

cancelled: השלב בוטל. זה יכול להיות בעקבות החלטה מכוונת לבטל את הביצוע או כאשר משתמש ביטל את הפעולה.

failed: השלב נכשל. זה אומר שהייתה בעיה בביצוע השלב והוא לא הצליח להשלים את משימתו כראוי.

completed: השלב הושלם בהצלחה. זה אומר שכל הפעולות הנדרשות בשלב זה בוצעו כמתוכנן והשלב נגמר בהצלחה.

expired: השלב הסתיים עקב פקיעת הזמן. זה יכול לקרות אם יש הגבלת זמן לביצוע השלב והזמן עבר לפני שהשלב הושלם.

כאשר תהליך (run) נמצא במצב "requires_action" והפעולה הנדרשת ("required_action.type") היא "submit_tool_outputs", פירוש הדבר שהתהליך מחכה לקלט מצד המשתמש או המערכת לפני שהוא יכול להמשיך. במקרה זה, הפעולה הנדרשת היא לשלוח את התוצאות של קריאות הכלים (tool calls) שבוצעו.

מצב "Requires Action": התהליך מחכה לפעולה מסוימת לפני שיוכל להמשיך. זה יכול לקרות כאשר יש צורך בקלט אנושי, או בתהליך שצריך להיות מושלם ידנית.

פעולה נדרשת - "Submit Tool Outputs": הפעולה המסוימת שנדרשת במצב זה היא לשלוח את התוצאות שנוצרו מהכלים (כגון תוצאות חישוב, תמונות, טקסטים וכו') שהתהליך קרא להם.

שליחה בבקשה אחת: חשוב לשלוח את כל התוצאות בבקשה אחת. זה אומר שאתה צריך לאגור את כל התוצאות מכל קריאות הכלים שבוצעו ולשלוח אותם יחד בתגובה אחת למערכת.

בפרקטיקה, זה אומר שאם יש לך תהליך שביצע מספר קריאות לכלים שונים, וכעת הוא מחכה לפעולה נוספת, עליך לאסוף את כל התוצאות מאותן קריאות ולשלוח אותן חזרה למערכת בבקשה אחת כדי שהתהליך יוכל להמשיך.

בדיקת תגובה של העוזר:

מחזירה רשימת ההודעות שנשלחו בתוך נושא ספציפי. היא מאפשרת למשתמש לקבל את כל התקשורת שהתרחשה בתוך נושא נתון, וניתן להתאים את הבקשה לפי מספר פרמטרים:

- **thread_id (חובה):** המזהה של הנושא (thread) שאתה רוצה לאחזר ממנו הודעות.
- **limit (אופציונלי):** מגביל את מספר ההודעות שיחזרו, וניתן לבקש בין 1 ל-100 הודעות.
- **order (אופציונלי):** מגדיר את סדר ההודעות לפי זמן יצירתן (asc לסדר עולה, desc לסדר יורד)

הבקשה מחזירה את רשימת ההודעות עם כל הפרטים הרלוונטיים כמו תוכן ההודעה, המשתמש ששלח אותה, וזמן השליחה.

תגובת ה-API מכיל רשימת הודעות שנשלחו במהלך דיאלוג בין משתמש לעוזר דיגיטלי. כל אלמנט ברשימה מייצג הודעה בודדת עם הפרמטרים הבאים:

- **id:** מזהה ייחודי של ההודעה.
 - **object:** סוג האובייקט, במקרה זה הודעה בנושא.
 - **created_at:** חותמת זמן של יצירת ההודעה.
 - **thread_id:** מזהה של הנושא שאליו שייכת ההודעה.
 - **role:** תפקיד השולח (למשל, משתמש או עוזר).
 - **content:** מערך שמכיל את תוכן ההודעה, כולל טקסט ואנוטציות.
 - **file_ids:** מערך של מזהי קבצים קשורים להודעה.
 - **assistant_id ו-run_id:** מזהים קשורים לעוזר ולריצה של ההודעה, אם קיימים.
 - **metadata:** מידע נוסף בפורמט מפתח-ערך.
-

הפעלת פונקציות במהלך שיחה:

כאשר ריצה מחכה לפעולה ("requires_action"), זה אומר שיש צורך בקלט נוסף מהמשתמש, כמו לשלוח תוצאות מכלי מסוים שנקראו במהלך השיחה. זה חיוני להבין כי התהליך מחכה לפעולת משתמש כדי להמשיך את הביצועים, ולא ימשיך באופן אוטומטי ללא זה. מטרתה להגיש פלטים מכלים שנקראו במהלך ריצה של עוזר דיגיטלי שנתקל בסטטוס "requires_action". כאשר עוזר דיגיטלי דורש פעולה נוספת כדי להמשיך, אתה צריך לשלוח את הפלטים שיצרו הכלים שהוא השתמש בהם. כאן פירוט הפרמטרים:

- **thread_id**: המזהה של הנושא (thread) שאליו שייכת הריצה.
- **run_id**: המזהה של הריצה שדורשת הגשת פלטי כלים.
- **tool_outputs**: מערך המכיל את הכלים והפלטים שלהם שיש להגיש.
 - **tool_call_id**: מזהה של קריאה ספציפית לכלי במהלך הריצה.
 - **output**: הפלט של הכלי שיש להגיש כדי להמשיך את הריצה.

קריאת ה-API תחזיר את האובייקט של הריצה עם השינויים המתאימים לאחר ההגשה.

ומכאן יש להמשיך לבדיקת סטטוס ריצה וכו"ל.

כמובן שיש עוד הרבה אפשרויות, אבל זה על קצה המזלג.

בהצלחה.