

ספריות PYPI - פרסום הספרייה

מזל טוב הספרייה סוף סוף מוכנה לפרסום!

בשביל לפרסם את הספרייה דבר ראשון צריך חשבון ב-PyPi, אם עדיין לא נרשמת לאתר זה הזמן [להירשם](#). ובאותה הזדמנות כדאי שנרשם גם ל-TestPyPi. האתר TestPyPi מאפשר לנו לנסות לבצע את כל תהליך פרסום החבילה מבלי שנצטרך להתמודד עם ההשלכות במקרה שההתקנה לא בוצעה כמו שצריך. כדי לעלות את הספרייה ל-PyPi נשתמש בכלי שנקרא twine. אפשר להוריד את המודול דרך pip :

```
pip install twine
```

השימוש ב-twine הוא מאוד פשוט, עוד מעט נראה כיצד להשתמש בו כדי לבדוק את החבילה ולפרסמה.

בניית הספרייה -

חבילות ב-PyPi לא מופצות כקוד מקור. במקום הן עטופות בחבילות הפצה (distribution packages). הפורמטים הנפוצים ביותר לחבילות הפצה הם source archives והמודול של פיתון wheels.

ה-source archive מכיל את כל קבצי הקוד והקבצים התומכים לו עטופים בקובץ tar. באופן דומה, wheel הוא קובץ zip שמכיל את הקוד והקבצים התומכים לו, בנוסף wheel כל ההרחבות המוכנות לשימוש.

כדי ליצור קובצי source archive ו-wheel נוכל להריץ את הפקודה הבאה משורת הפקודה:

```
python setup.py sdist bdist_wheel
```

זה יצור תיקייה חדשה שנקראת dist ובה שני קבצים חדשים עם שם jchkv והגירסה שלהם:

```
my_package-1.0.0-py3-none-any.whl
```

```
my_package-1.0.0.tar.gz
```

הערה: אם אתם משתמשים בווינדוס ה-source archive יהיה קובץ zip. כברירת מחדל, אבל אפשר לשנות את הפורמט אם נשתמש בדגל --formats. ואז נבחר את הפורמט שנרצה, במקרה זה gztar :

```
python setup.py sdist --formats=gztar
```

אתם בטח שואלים איך הפונקציה setup() מקבלת ארגומנטים מהטרמינל אם לא הגדרנו את זה בסקריפט. אז למעשה כל המשתנים של שורת הפקודה ממומשים במקום בספרייה distutils. ואפשר גם להוסיף לסקריפט עוד כמה ארגומנטים כשמפעילים את הפקודה בשורה הפקודה. בשביל לראות את רשימת הפקודות האפשריות ממולץ להסתכל על:

```
python setup.py --help
```

או אם להיות ספציפים עם הארגומנט שלנו:

```
python setup.py sdist --help
```

```
python setup.py bdist_wheel --help
```



לבחון את החבילה-

תחילה נבדוק אם חבילות ההפצה שיצרנו מכילות את הקבצים כפי שצריך. בלינוקס וב-macOS ניתן לראות את רשימת הקבצים בחבילה עם הפקודה `tar tzf` ואז שם חבילת ההפצה:

```
tar tzf my_package-1.0.0.tar.gz
```

בווינדוס אפשר להשתמש בתוכנות כמו `zip-7`.

אנחנו אמורים לראות רשימה של קבצי קוד המקור, ביחד עם כמה קבצים חדשים שנוצרו מפרמטרים שהכנסנו לקובץ `setup.py`. בפרט יש לבדוק שכל תתי הספריות והקבצים התומכים לקוד נמצאים בחבילת ההפצה.

אפשר גם לבדוק את קובץ ה-`zip` שנוצר מהמודול `wheels`, משום שזהו קובץ `zip` ניתן פשוט להיכנס אליו או לחלץ את הקבצים.

אבל אם ה-`source archive` (הקובץ `tar`) מכיל את כל הקבצים, אפשר לצאת מנקודת הנחה שגם בקבצים שנוצרו ב-`wheels` יש את כל הקבצים.

החל מגרסה 1.12.0 של `twine` ניתן לבדוק שתיאור החבילה ירונדר כמו שצריך ל-PyPi, וכל מה שצריך לעשות הוא להפעיל את הפקודה `twine check` על התיקיה שבה נמצאים חבילות ההפצה:

```
$ twine check dist/*
```

```
Checking distribution dist/my_package-1.0.0-py3-none-any.whl: Passed
```

```
Checking distribution dist/ my_package-1.0.0.tar.gz: Passed
```

העלאת הקבצים ל-PyPI :

עכשיו אנחנו למעשה מוכנים לעלות את החבילה ל-PyPI. כדי לעשות את זה נשתמש שוב בכלי `twine`, ונגיד לו לעלות את חבילות ההפצה שיצרנו.

אבל לפני נרצה לבחון שהחבילות יעלו כמו שצריך, לכן נעלה קודם את החבילות ל-TestPyPI:

```
$ twine upload --repository-url https://test.pypi.org/legacy/ dist/*
```

הוא יבקש את שם המשתמש והסיסמא.

אם העלנו כמו שצריך את החבילה נוכל לראות אותה באתר TestPyPI. היכנסו לחבילה וודאו שהכל עלה כמו שצריך.

לאחר שראינו שהכל עלה כפי שציפינו כל מה שנשאר לעשות הוא לעלות את החבילה לאתר של PyPI. זה יהיה פשוט אפילו יותר מלעלות אותו ל-TestPyPI, כל מה שצריך זה את הפקודה הבאה:

```
$ twine upload dist/*
```

אחר כך הוא יבקש את שם המשתמש וסיסמא.

מזל טוב החבילה שלכם ב-PyPI!

כנסו לאתר וודאו שהיא אכן שם. אחר כך בחנו שהיא עלתה כמו שצריך- צרו מרחב וירטואלי חדש והורידו את החבילה שלכם.

