



Tablib Dataset Library v0.6.1 Released!

2010

2 min read • 376 words

I'm pleased to announce a new Python module: [Tablib](#). Tablib is a simple module for working with tabular datasets. It allows you create tables of data using standard Python datatypes, manipulate them, and easily export to Excel, JSON, YAML, and CSV.

Basic Usage

```
import tablib

headers = ('first_name', 'last_name', 'gpa')
data = [('John', 'Adams', 90), ('George', 'Washington', 67)]

data = tablib.Dataset(*data, headers=headers)
```

You can manipulate your data like a standard Python list:

```
>>> data.append(('Henry', 'Ford', 83))

>>> print data['first_name']
['John', 'George', 'Henry']

>>> del data[1]
```

Export Formats

You can easily export your data to JSON, YAML, XLS, and CSV:

```
>>> print data.json
[{"first_name": "John", "last_name": "Adams", "gpa": 90},
 {"first_name": "Henry", "last_name": "Ford", "gpa": 83}]

>>> print data.yaml
- {gpa: 90, first_name: John, last_name: Adams}
- {gpa: 83, first_name: Henry, last_name: Ford}

>>> print data.csv
first_name,last_name,gpa
John,Adams,90
Henry,Ford,83

>>> open('people.xls', 'w').write(data.xls)
```

Excel files with multiple sheets are also supported (via the `DataBook` object).

Design Philosophy

I built Tablib because I was frustrated with the lack of a simple, intuitive way to work with tabular data in Python. Most solutions were either too complex for simple tasks or too limited for real-world use. Tablib tries to strike a balance – it's simple enough for quick scripts but powerful enough for data processing workflows.

The key insight was treating datasets as first-class objects that know how to represent themselves in different formats. Instead of having separate functions for each export format, the data itself knows how to become JSON, CSV, Excel, etc. This feels much more natural and Pythonic.

Features and Future

Current features include:

- Intuitive API that works like Python lists and dictionaries
- Support for headers and column access
- Export to JSON, YAML, CSV, XLS, and more
- Excel workbooks with multiple sheets via DataBook
- Pure Python implementation, no external dependencies for core functionality

I'm already working on v0.7, which will include:

- Import capabilities (read existing CSV, Excel files)
- Additional export formats (LaTeX tables, HTML)
- Better handling of data types and validation
- Performance optimizations for large datasets

The goal is to make Tablib the go-to library for anyone who needs to work with structured data in Python, whether you're a data scientist, web developer, or just someone who occasionally needs to wrangle CSV files.

[\[Source on GitHub\]](#) [\[PyPi Listing\]](#)