

Web Applications and Services

- Reading: Module 2 Introduction
10 min
- Reading: Web Applications and Services
10 min
- Reading: Data Serialization
10 min
- Reading: Data Serialization Formats
10 min
- Reading: More About JSON
10 min

Python Requests

Module 2 Project

Data Serialization Formats

There are lots and lots of ways to serialize data. In this course, we'll cover a couple of the most common ones and we'll look into how you can use them from Python. Once you get the hang of how this works, it's super easy to use a different format if needed.

JSON (JavaScript Object Notation) is the serialization format that we'll use the most in this course. We'll go into some details later but, for now, let's just use the **json** module to convert our **people** list of dictionaries into JSON format.

```
1 import json
2
3 with open('people.json', 'w') as people_json:
4     json.dump(people, people_json, indent=2)
```

This code uses the **json.dump()** function to serialize the **people** object into a JSON file. The contents of the file will look something like this:

```
1  [
2    {
3      "name": "Sabrina Green",
4      "username": "sgreen",
5      "phone": {
6        "office": "802-867-5309",
7        "cell": "802-867-5310"
8      },
9      "department": "IT Infrastructure",
10     "role": "Systems Administrator"
11   },
12   {
13     "name": "Eli Jones",
14     "username": "ejones",
15     "phone": {
16       "office": "684-348-1127"
17     },
18     "department": "IT Infrastructure",
19     "role": "IT Specialist"
20   },
21 ]
22
```

YAML (Yet Another Markup Language) has a lot in common with JSON. They're both formats that can be easily understood by a human when looking at the contents. In this example, we're using the **yaml.safe_dump()** method to serialize our object into YAML:

```
1 import yaml
2
3 with open('people.yaml', 'w') as people_yaml:
4     yaml.safe_dump(people, people_yaml)
```

That code will generate a **people.yaml** file that looks like this:

```
1 - department: IT Infrastructure
2   name: Sabrina Green
3   phone:
4     cell: 802-867-5310
5     office: 802-867-5309
6   role: Systems Administrator
7   username: sgreen
8 - department: IT Infrastructure
9   name: Eli Jones
10  phone:
11    office: 684-348-1127
12  role: IT Specialist
13  username: ejones
```

While this doesn't look exactly like the JSON example above, both formats list the names of the fields as part of the format, so that both the programs *parsing* the data and the humans looking at it can make sense out of it. The main difference is how these formats are used. JSON is used frequently for transmitting data between web services, while YAML is used the most for storing configuration values.

These are just a couple of the most common data serialization formats. We've left out some other pretty common ones like **Python pickle**, **Protocol Buffers**, or the **eXtensible Markup Language (XML)**. Each of them is useful in a specific context, although not the focus of this course. You can read more about them by following those links.

Mark as completed

