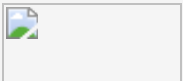


Pydantic v2

Technical hour

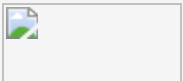
Sunniva Indrehus



Pydantic v2

Technical hour

Sunniva Indrehus



```
$ whoami
```



NGI scientist's aim

Understand a simplified version
of the real world



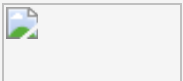
Figure credit: [Ali Bati](#)

Engineering example

Define a valid **data model**

Perform operation X

Interpret results



Pydantic

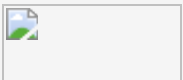
💥 v2.0 released 30.06.2023 💥

What's the fuzz?

v2 [promise](#) 5-50x speed up compared to Pydantic v1

★ GitHub [stars](#) 15.4k (22.08.23)

📦 [pr week](#) \approx 22M (week 34 2023)



Pydantic

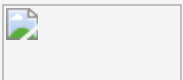
Why?

🐍 Python is dynamically typed, and [PEP484 Type Hints](#) is not enforced during run time

What?

Pydantic is the most widely used data validation library for Python.

From the official docs

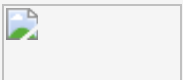


Pydantic v2

A new core

Validations outside of Python

Recursive function calls with Rust
and small overhead



Pydantic v2

Some new features

Functionality for discriminated unions

`pydantic.functional_validators` let you do validation without `base_model` (e.g a function)

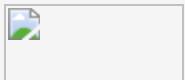
Possibility to use `TypeAdapter` instead of `BaseModel`

You can define `field_serializer` you can do *custom serialization*

Examples and inspiration

[Migration guide](#)

[v1 to v2 video example](#)



Speed-up for *free*

Language changes

`root_validator`  `model_validator`

```
validated_data = [Wine(**item).model_dump(exclude_none=True,  
by_alias=True) for item in data]
```

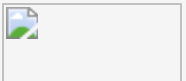


```
validated_data = [Wine(**item).dict(exclude_none=exclude_none) for  
item in data]
```

Speed-up with *investements* 💰

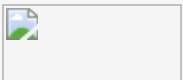


Demo some benchmarking



Example run times

Fruit	Colour	Amount	Cost
Banana	Yellow	4	£1.00
Apple	Red	2	£0.60
Orange	Orange	10	£2.50
Coconut	Brown	1	£1.50



Credit

[Pydantic v2 Plan](#) (accessed 22.08.2023)

Terrence Dorsey and Samuel Colvin [Pydantic v2 Pre Release](#) (accessed 22.08.2023)

Prashanth Rao, The Data Quarry, [Obtain a 5x speedup for free by upgrading to Pydantic v2](#) (accessed 22.08.2023)

Yaakov Bressler in [Medium](#) (accessed 22.08.2023)

Samuel Colvin at [PyCon US 2023](#) (accessed 22.08.2023)

Configuring [mypy with Pydantic](#) (accessed 22.08.2023)

