Data Transfer Objects

Advanced Python Programming

—

Valerio Velardo - The Sound of AI

What’s a Data Transfer Object?

What’s a Data Transfer Object?

- Object that encapsulates data

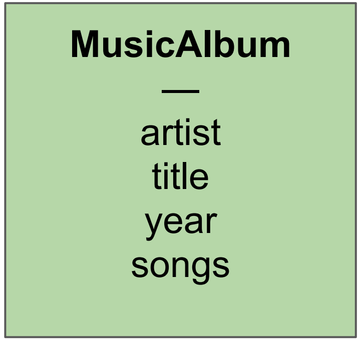
What’s a Data Transfer Object?

- Object that encapsulates data
- No behaviour

What’s a Data Transfer Object?

- Object that encapsulates data
- No behaviour
- Transfer data between different processes

Music Album DTO

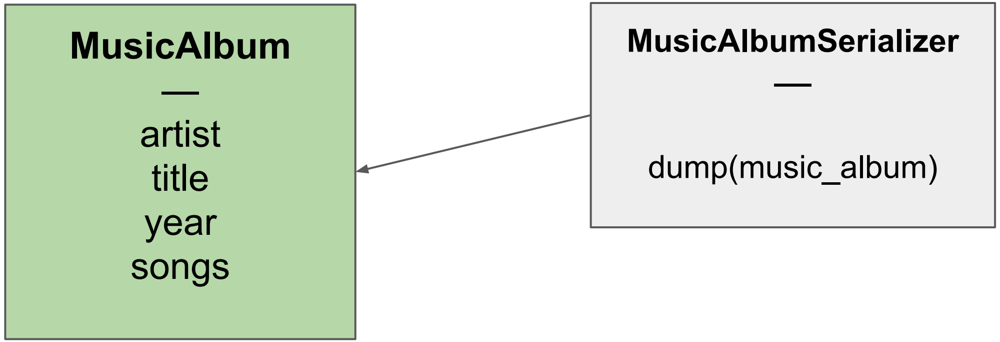


MusicAlbum

—

artist
title
year
songs

Music Album DTO



MusicAlbum

—

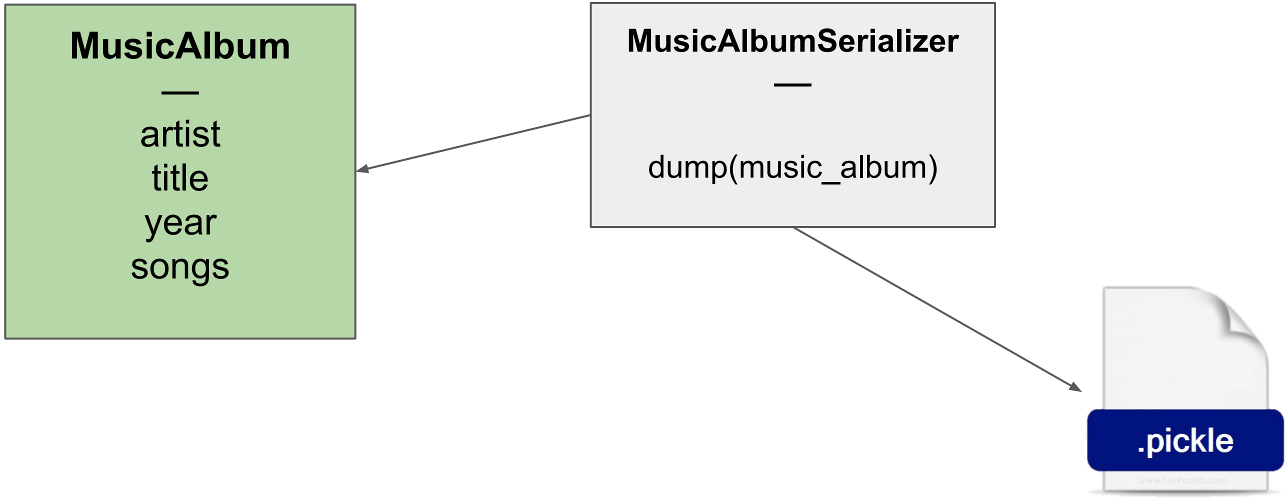
artist
title
year
songs

MusicAlbumSerializer

—

dump(music_album)

Music Album DTO



MusicAlbum

—

artist
title
year
songs

MusicAlbumSerializer

—

dump(music_album)

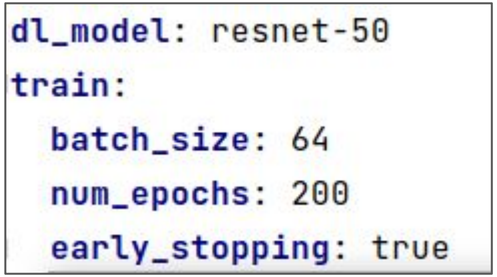
When should you use DTOs?

When should you use DTOs?

- Load configurations

When should you use DTOs?

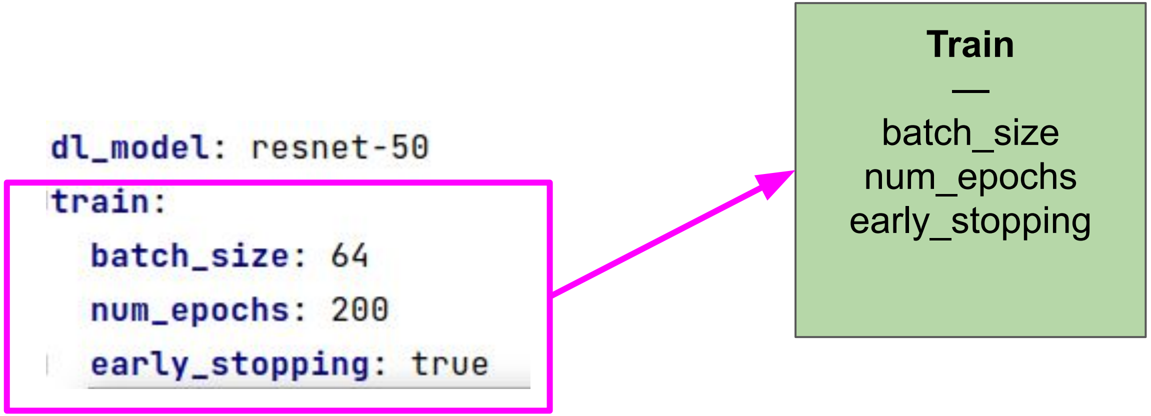
- Load configurations



```yaml
dl_model: resnet-50
train:
  batch_size: 64
  num_epochs: 200
  early_stopping: true
```

When should you use DTOs?

- Load configurations



```yaml
dl_model: resnet-50
train:
  batch_size: 64
  num_epochs: 200
  early_stopping: true
```

Train

—

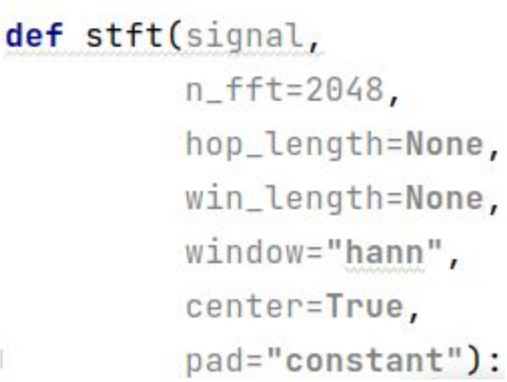
batch_size
num_epochs
early_stopping

When should you use DTOs?

- Load configurations
- Encapsulate parameters for method calls

When should you use DTOs?

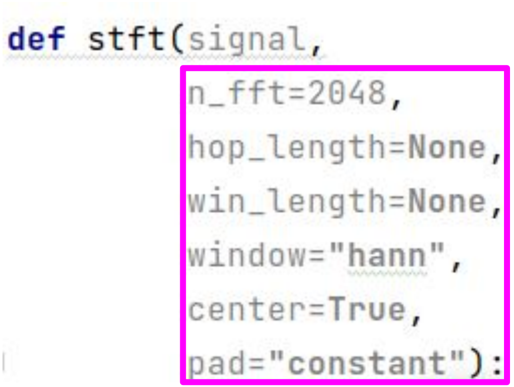
- Load configurations
- Encapsulate parameters for method calls



```
def stft(signal,
         n_fft=2048,
         hop_length=None,
         win_length=None,
         window="hann",
         center=True,
         pad="constant"):
```

When should you use DTOs?

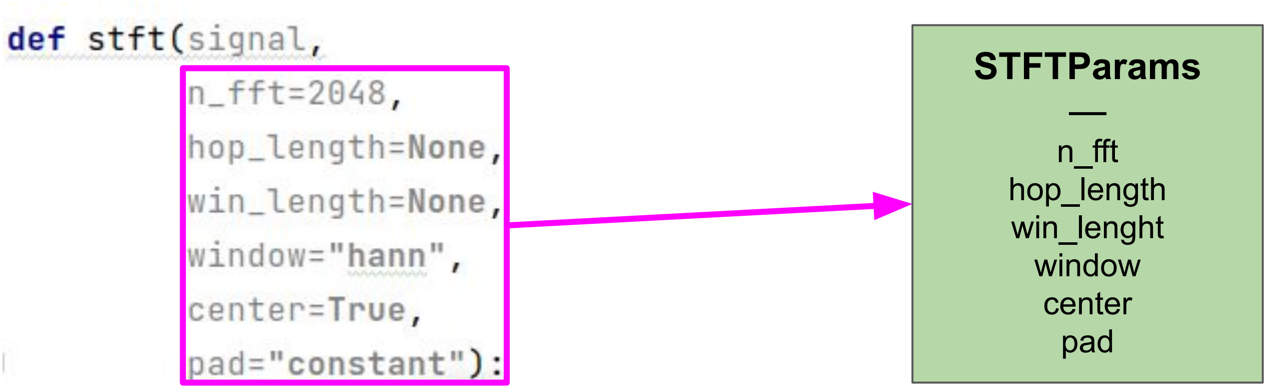
- Load configurations
- Encapsulate parameters for method calls



```
def stft(signal,
         n_fft=2048,
         hop_length=None,
         win_length=None,
         window="hann",
         center=True,
         pad="constant"):
```

When should you use DTOs?

- Load configurations
- Encapsulate parameters for method calls



```
def stft(signal,
         n_fft=2048,
         hop_length=None,
         win_length=None,
         window="hann",
         center=True,
         pad="constant"):
```

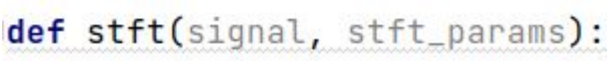
STFTParams

—

n_fft
hop_length
win_lenght
window
center
pad

When should you use DTOs?

- Load configurations
- Encapsulate parameters for method calls



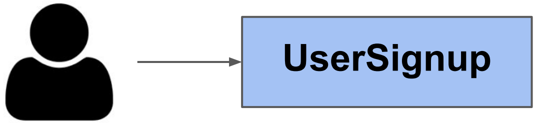
```
def stft(signal, stft_params):
```

When should you use DTOs?

- Load configurations
- Encapsulate parameters for method calls
- As an interface, to transfer objects across different components of an application

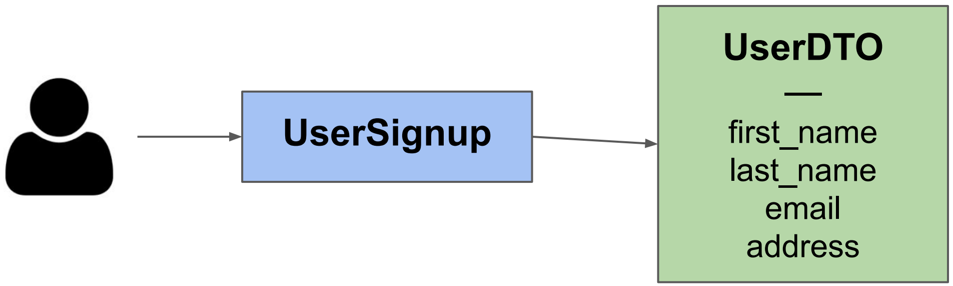
When should you use DTOs?

- Load configurations
- Encapsulate parameters for method calls
- As an interface, to transfer objects across different components of an application



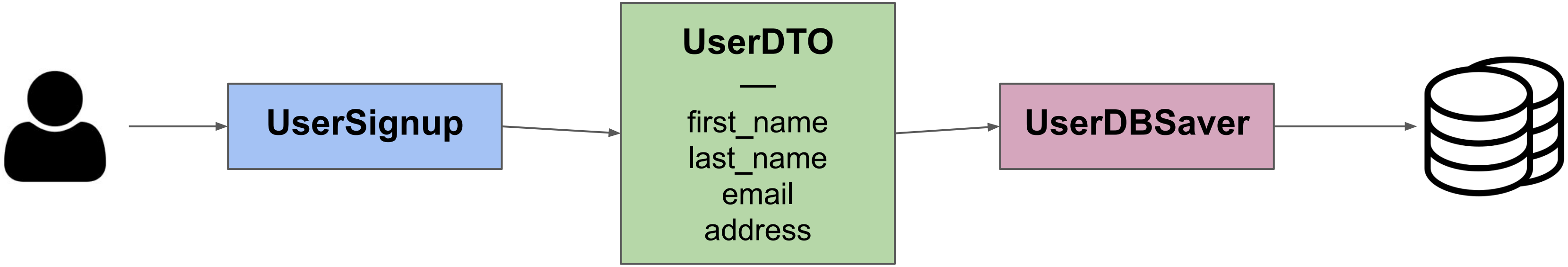
When should you use DTOs?

- Load configurations
- Encapsulate parameters for method calls
- As an interface, to transfer objects across different components of an application



When should you use DTOs?

- Load configurations
- Encapsulate parameters for method calls
- As an interface, to transfer objects across different components of an application



DTO in Python

DTO in Python

- Vanilla Python classes
    - Time intensive: implement custom __init__, __str__, __eq__

DTO in Python

- Vanilla Python classes
  - Time intensive: implement custom __init__, __str__, __eq__
- Python data classes
  - @dataclass decorator from *dataclasses*
  - Focus only on parameters
  - __init__, __str__, __eq__ come for free

DTO in Python

- Vanilla Python classes
  - Time intensive: implement custom __init__, __str__, __eq__
- Python *data classes*
  - @dataclass decorator from *dataclasses*
  - Focus only on parameters
  - __init__, __str__, __eq__ come for free
- *pydantic* objects
  - Python package for data validation
  - __init__, __str__, __eq__ come for free
  - Enforce type hints