

```
from pydantic import BaseModel, Field
```

```
class TextToVideoRequest(BaseModel):
```

```
    model_name: str = Field(
```

```
        ...,
```

```
        example="default-text-to-video-model",
```

```
        description="Name of the text-to-video model to be used",
```

```
    )
```

```
    task: str = Field(
```

```
        ...,
```

```
        example="A girl smiling",
```

```
        description="Text input that will be converted to video",
```

```
    )
```

```
    resolution: str = Field(
```

```
        default="1080p", example="1080p", description="Resolution of the output video"
```

```
    )
```

```
    length: int = Field(
```

```
        default=60, example=60, description="Desired length of the video in seconds"
```

```
    )
```

```
    style: str = Field(
```

```
        default="realistic",
```

```
        example="realistic",
```

```
        description="Stylistic parameters or themes",
```

```
    )
```

```
    n: int = Field(default=1, example=1, description="Number of videos to generate")
```

```
output_type: str = Field(
    default="gif",
    example="gif",
    description="Type of output to be generated, e.g. video, gif, etc.",
)

output_path: str = Field(
    None,
    example="animate.gif",
    description="Path to save the output video",
)
```

```
class ErrorResponse(BaseModel):

    code: str = Field(..., example="400", description="Error code in case of failure")

    message: str = Field(
        ..., example="Invalid input text", description="Detailed error message"
    )
```

```
class TextToVideoResponse(BaseModel):

    status: str = Field(..., example="success", description="Status of the response")

    request_details: TextToVideoRequest = Field(
        ..., description="Details of the initial request"
    )

    output_path: str = Field(
        None,
```

```
example="animate.gif",  
description="URL where the generated video can be accessed",  
)  
  
# error: ErrorResponse = Field(  
#     None, description="Error details if the status is 'error'"  
# )
```