

Pydantic schema_extra

```
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()

class Item(BaseModel):
    name: str
    description: str | None = None
    price: float
    tax: float | None = None

class Config:
    schema_extra = {
        "example": {
            "name": "Foo",
            "description": "A very nice Item",
            "price": 35.4,
            "tax": 3.2,
        }
    }

@app.put("/items/{item_id}")
async def update_item(item_id: int, item: Item):
    results = {"item_id": item_id, "item": item}
    return results
```

Field additional arguments

```
from fastapi import FastAPI
from pydantic import BaseModel, Field

app = FastAPI()

class Item(BaseModel):
```

```

name: str = Field(example="Foo")
description: str | None = Field(default=None, example="A very nice Item")
price: float = Field(example=35.4)
tax: float | None = Field(default=None, example=3.2)

```

```

@app.put("/items/{item_id}")
async def update_item(item_id: int, item: Item):
    results = {"item_id": item_id, "item": item}
    return results

```

OpenAPI

```

from fastapi import Body, FastAPI
from pydantic import BaseModel

```

```

app = FastAPI()

```

```

class Item(BaseModel):
    name: str
    description: str | None = None
    price: float
    tax: float | None = None

```

```

@app.put("/items/{item_id}")
async def update_item(
    item_id: int,
    item: Item = Body(
        example={
            "name": "Foo",
            "description": "A very nice Item",
            "price": 35.4,
            "tax": 3.2,
        },
    ),
):
    results = {"item_id": item_id, "item": item}
    return results

```

Example in the docs UI

```
from fastapi import Body, FastAPI
from pydantic import BaseModel

app = FastAPI()

class Item(BaseModel):
    name: str
    description: str | None = None
    price: float
    tax: float | None = None

@app.put("/items/{item_id}")
async def update_item(
    *,
    item_id: int,
    item: Item = Body(
        examples={
            "normal": {
                "summary": "A normal example",
                "description": "A **normal** item works correctly.",
                "value": {
                    "name": "Foo",
                    "description": "A very nice Item",
                    "price": 35.4,
                    "tax": 3.2,
                },
            },
            "converted": {
                "summary": "An example with converted data",
                "description": "FastAPI can convert price `strings` to actual\n`numbers` automatically",
                "value": {
                    "name": "Bar",
                    "price": "35.4",
                },
            },
        },
    ),

```

```
        "invalid": {
            "summary": "Invalid data is rejected with an error",
            "value": {
                "name": "Baz",
                "price": "thirty five point four",
            },
        },
    },
),
):
    results = {"item_id": item_id, "item": item}
    return results
```