Declare Request Example Data

You can declare examples of the data your app can receive.

Here are several ways to do it.

Pydantic `schema_extra`

You can declare an `example` for a Pydantic model using `Config` and `schema_extra`, as described in [Pydantic's docs: Schema customization](#):

=== "Python 3.6 and above"

~~~~
```Python hl_lines="15-23"
{!> ../../../docs_src/schema_extra_example/tutorial001.py!}
```
~~~~

=== "Python 3.10 and above"

~~~~
```Python hl_lines="13-21"
{!> ../../../docs_src/schema_extra_example/tutorial001_py310.py!}
```
~~~~

That extra info will be added as-is to the output **JSON Schema** for that model, and it will be used in the API docs.

!!! tip You could use the same technique to extend the JSON Schema and add your own custom extra info.

```
For example you could use it to add metadata for a frontend user interface, etc.
```

`Field` additional arguments

When using `Field()` with Pydantic models, you can also declare extra info for the **JSON Schema** by passing any other arbitrary arguments to the function.

You can use this to add `example` for each field:

=== "Python 3.6 and above"

~~~~
```Python hl_lines="4  10-13"
{!> ../../../docs_src/schema_extra_example/tutorial002.py!}
```
~~~~

=== "Python 3.10 and above"

~~~~
```Python hl_lines="2  8-11"
{!> ../../../docs_src/schema_extra_example/tutorial002_py310.py!}
```
~~~~

!!! warning Keep in mind that those extra arguments passed won't add any validation, only extra information, for documentation purposes.

`example` and `examples` in OpenAPI

When using any of:

- `Path()`
- `Query()`
- `Header()`
- `Cookie()`
- `Body()`
- `Form()`
- `File()`

you can also declare a data `example` or a group of `examples` with additional information that will be added to **OpenAPI**.

`Body` with `example`

Here we pass an `example` of the data expected in `Body()`:

=== "Python 3.6 and above"

~~~
```Python hl_lines="21-26"
{!> ../../../docs_src/schema_extra_example/tutorial003.py!}
```
~~~

=== "Python 3.10 and above"

~~~
```Python hl_lines="19-24"
{!> ../../../docs_src/schema_extra_example/tutorial003_py310.py!}
```
~~~

Example in the docs UI

With any of the methods above it would look like this in the `/docs`:

`Body` with multiple `examples`

Alternatively to the single `example`, you can pass `examples` using a `dict` with **multiple examples**, each with extra information that will be added to **OpenAPI** too.

The keys of the `dict` identify each example, and each value is another `dict`.

Each specific example `dict` in the `examples` can contain:

- `summary`: Short description for the example.
- `description`: A long description that can contain Markdown text.
- `value`: This is the actual example shown, e.g. a `dict`.
- `externalValue`: alternative to `value`, a URL pointing to the example. Although this might not be supported by as many tools as `value`.

=== "Python 3.6 and above"

~~~
```Python hl_lines="22-48"
{!> ../../../docs_src/schema_extra_example/tutorial004.py!}
```
~~~

=== "Python 3.10 and above"

~~~
```Python hl_lines="20-46"
{!> ../../../docs_src/schema_extra_example/tutorial004_py310.py!}
```
~~~

Examples in the docs UI

With `examples` added to `Body()` the `/docs` would look like:

Technical Details

!!! warning These are very technical details about the standards **JSON Schema** and **OpenAPI**.

```
If the ideas above already work for you, that might be enough, and you probably don't
need these details, feel free to skip them.
```

When you add an example inside of a Pydantic model, using `schema_extra` or `Field(example="something")` that example is added to the **JSON Schema** for that Pydantic model.

And that **JSON Schema** of the Pydantic model is included in the **OpenAPI** of your API, and then it's used in the docs UI.

JSON Schema doesn't really have a field `example` in the standards. Recent versions of JSON Schema define a field `examples`, but OpenAPI 3.0.3 is based on an older version of JSON Schema that didn't have `examples`.

So, OpenAPI 3.0.3 defined its own `example` for the modified version of **JSON Schema** it uses, for the same purpose (but it's a single `example`, not `examples`), and that's what is used by the API docs UI (using Swagger UI).

So, although `example` is not part of JSON Schema, it is part of OpenAPI's custom version of JSON Schema, and that's what will be used by the docs UI.

But when you use `example` or `examples` with any of the other utilities (`Query()`, `Body()`, etc.) those examples are not added to the JSON Schema that describes that data (not even to OpenAPI's own version of JSON Schema), they are added directly to the *path operation* declaration in OpenAPI (outside the parts of OpenAPI that use JSON Schema).

For `Path()`, `Query()`, `Header()`, and `Cookie()`, the `example` or `examples` are added to the OpenAPI definition, to the `Parameter Object` (in the specification).

And for `Body()`, `File()`, and `Form()`, the `example` or `examples` are equivalently added to the OpenAPI definition, to the `Request Body Object`, in the field `content`, on the `Media Type Object` (in the specification).

On the other hand, there's a newer version of OpenAPI: **3.1.0**, recently released. It is based on the latest JSON Schema and most of the modifications from OpenAPI's custom version of JSON Schema are removed, in exchange of the features from the recent versions of JSON Schema, so all these small differences are reduced. Nevertheless, Swagger UI currently doesn't support OpenAPI 3.1.0, so, for now, it's better to continue using the ideas above.