Templates

You can use any template engine you want with **FastAPI**.

A common choice is Jinja2, the same one used by Flask and other tools.

There are utilities to configure it easily that you can use directly in your **FastAPI** application (provided by Starlette).

Install dependencies

Install `jinja2`:

```
$ pip install jinja2

---> 100%
```

Using `Jinja2Templates`

- Import `Jinja2Templates`.
- Create a `templates` object that you can re-use later.
- Declare a `Request` parameter in the *path operation* that will return a template.
- Use the `templates` you created to render and return a `TemplateResponse`, passing the `request` as one of the key-value pairs in the Jinja2 "context".

```
{!../../../docs_src/templates/tutorial001.py!}
```

!!! note Notice that you have to pass the `request` as part of the key-value pairs in the context for Jinja2. So, you also have to declare it in your *path operation*.

!!! tip By declaring `response_class=HTMLResponse` the docs UI will be able to know that the response will be HTML.

!!! note "Technical Details" You could also use `from starlette.templating import Jinja2Templates`.

```
**FastAPI** provides the same `starlette.templating` as `fastapi.templating` just as a
convenience for you, the developer. But most of the available responses come directly
from Starlette. The same with `Request` and `StaticFiles`.
```

Writing templates

Then you can write a template at `templates/item.html` with:

```
{!../../../docs_src/templates/templates/item.html!}
```

It will show the `id` taken from the "context" `dict` you passed:

```
{"request": request, "id": id}
```

Templates and static files

And you can also use `url_for()` inside of the template, and use it, for example, with the `StaticFiles` you mounted.

```
{!../../../docs_src/templates/templates/item.html!}
```

In this example, it would link to a CSS file at `static/styles.css` with:

```
{!../../../docs_src/templates/static/styles.css!}
```

And because you are using `StaticFiles`, that CSS file would be served automatically by your **FastAPI** application at the URL `/static/styles.css`.

More details

For more details, including how to test templates, check Starlette's docs on templates.