Custom Request and APIRoute class

In some cases, you may want to override the logic used by the `Request` and `APIRoute` classes.

In particular, this may be a good alternative to logic in a middleware.

For example, if you want to read or manipulate the request body before it is processed by your application.

!!! danger This is an "advanced" feature.

`If you are just starting with **FastAPI** you might want to skip this section.`

Use cases

Some use cases include:

- Converting non-JSON request bodies to JSON (e.g. `msgpack`).
- Decompressing gzip-compressed request bodies.
- Automatically logging all request bodies.

Handling custom request body encodings

Let's see how to make use of a custom `Request` subclass to decompress gzip requests.

And an `APIRoute` subclass to use that custom request class.

Create a custom `GzipRequest` class

!!! tip This is a toy example to demonstrate how it works, if you need Gzip support, you can use the provided `GzipMiddleware`.

First, we create a `GzipRequest` class, which will overwrite the `Request.body()` method to decompress the body in the presence of an appropriate header.

If there's no `gzip` in the header, it will not try to decompress the body.

That way, the same route class can handle gzip compressed or uncompressed requests.

`Python hl_lines="8-15" {!../../../docs_src/custom_request_and_route/tutorial001.py!}`

Create a custom `GzipRoute` class

Next, we create a custom subclass of `fastapi.routing.APIRoute` that will make use of the `GzipRequest`.

This time, it will overwrite the method `APIRoute.get_route_handler()`.

This method returns a function. And that function is what will receive a request and return a response.

Here we use it to create a `GzipRequest` from the original request.

```
Python hl_lines="18-26" {!../../../docs_src/custom_request_and_route/tutorial001.py!}
```

!!! note "Technical Details" A `Request` has a `request.scope` attribute, that's just a Python `dict` containing the metadata related to the request.

```
A `Request` also has a `request.receive`, that's a function to "receive" the body of the rec

The `scope` `dict` and `receive` function are both part of the ASGI specification.

And those two things, `scope` and `receive`, are what is needed to create a new `Request` in

To learn more about the `Request` check <a href="https://www.starlette.io/requests/" class="
```

The only thing the function returned by `GzipRequest.get_route_handler` does differently is convert the `Request` to a `GzipRequest`.

Doing this, our `GzipRequest` will take care of decompressing the data (if necessary) before passing it to our *path operations*.

After that, all of the processing logic is the same.

But because of our changes in `GzipRequest.body`, the request body will be automatically decompressed when it is loaded by **FastAPI** when needed.

Accessing the request body in an exception handler

!!! tip To solve this same problem, it's probably a lot easier to use the `body` in a custom handler for `RequestValidationError` (Handling Errors).

```
But this example is still valid and it shows how to interact with the internal components.
```

We can also use this same approach to access the request body in an exception handler.

All we need to do is handle the request inside a `try`/`except` block:

```
Python hl_lines="13  15" {!../../../docs_src/custom_request_and_route/tutorial002.py!}
```

If an exception occurs, the`Request` instance will still be in scope, so we can read and make use of the request body when handling the error:

```
Python hl_lines="16-18" {!../../../docs_src/custom_request_and_route/tutorial002.py!}
```

Custom `APIRoute` class in a router

You can also set the `route_class` parameter of an `APIRouter`:

```
Python hl_lines="26" {!../../../docs_src/custom_request_and_route/tutorial003.py!}
```

In this example, the *path operations* under the `router` will use the custom `TimedRoute` class, and will have an extra `X-Response-Time` header in the response with the time it took to generate the response:

```
Python hl_lines="13-20" {!../../../docs_src/custom_request_and_route/tutorial003.py!}
```