



Integrating Gradio with FastAPI (instead of Django)

You *can* mount a Gradio interface on a FastAPI app and still pass along your chat/user/session context. Gradio provides a helper `gr.mount_gradio_app()` for FastAPI. In practice you would do something like:

```
from fastapi import FastAPI
import gradio as gr

app = FastAPI()

# Define your Gradio interface or Blocks
io = gr.Interface(lambda x: "Hello, " + x + "!", "textbox", "textbox")

# Mount the Gradio app under a sub-path (e.g. "/gradio")
app = gr.mount_gradio_app(app, io, path="/gradio")
```

This serves the Gradio UI at `http://<host>:8000/gradio`¹. (By contrast, Django has no direct “mount” support, so people typically run Gradio as a separate service and embed it via an iframe or link.) Mounting under FastAPI lets you handle all routing and sessions in one place.

Passing chat/user/session context

Because FastAPI controls the session and user state, you can inject that into Gradio callbacks. For example, add Starlette’s `SessionMiddleware` to FastAPI:

```
from starlette.middleware.sessions import SessionMiddleware
app.add_middleware(SessionMiddleware, secret_key=SECRET_KEY)
```

Then any Gradio callback function can take a parameter `request: gr.Request`. Gradio will pass in the FastAPI request object, so inside your function you can read `request.request.session`, cookies, headers, etc. For instance:

```
def get_user_info(request: gr.Request):
    username = request.request.session.get("username", "Guest")
    return f"Welcome, {username}!"
```

You can use `demo.load` (for Blocks) or other event handlers to call this function when the UI loads. Skowalak's example shows exactly this pattern: defining `get_user_info(request: gr.Request)` and then doing `demo.load(get_user_info, outputs=user_info)` to display a greeting from the session ² ³. In short, any "chat context" or user data you store on the FastAPI side (e.g. in the session or a database keyed by session/user) can be fetched inside your Gradio logic via the `gr.Request` object ⁴ ².

Controlling navigation / redirects

Gradio's Python API does *not* natively support returning an HTTP redirect from a callback. Instead, you must trigger navigation on the client side. Common workarounds are:

- **Button with `link=`.** A `gr.Button` can have a `link="/some-path"` parameter, which makes a click go to that URL. For example, you can add a Logout button that points to your FastAPI `/logout` route ⁵. Clicking it will navigate the browser there.
- **Custom JavaScript on events.** You can attach JS callbacks to buttons or other components. For instance, Poilet66 shows using `btn.click(None, None, None, js="() => { window.location.href = '/newurl' }")` to redirect when a button is clicked ⁶. This runs in the browser and changes `window.location.href`.

In practice, to "govern the URL redirection by receiving events", you typically have the Gradio UI trigger a FastAPI route or a direct link. For example, your Gradio interface could set a hidden link or run custom JS on certain events to jump to another path. But Gradio won't call back into FastAPI to issue a redirect itself. Instead, use the above methods (button links or JS callbacks) to navigate the user to new URLs.

Summary: With FastAPI you can mount Gradio (`gr.mount_gradio_app`) and still use FastAPI's session/user management inside your Gradio handlers (via the `gr.Request` parameter) ¹ ². To redirect or navigate, put the logic in the UI (e.g. a Button `link` or JS) because Gradio doesn't yet support server-side redirects from event handlers ⁵ ⁶. These techniques give you tight integration of Gradio into your FastAPI app while passing along chat/user/session context as needed.

Sources: Gradio docs on FastAPI mounting and request access ¹ ⁴; StackOverflow example using `gr.Request` and `Blocks.load` ² ³; Gradio GitHub discussion on using JS in events ⁶.

¹ ⁴ ⁵ Sharing Your App

<https://www.gradio.app/guides/sharing-your-app>

² ³ python - Gradio HTML component display mounted on FAST API - Stack Overflow

<https://stackoverflow.com/questions/77195870/gradio-html-component-display-mounted-on-fast-api>

⁶ Allow event listeners to redirect to another page · Issue #7838 · gradio-app/gradio · GitHub

<https://github.com/gradio-app/gradio/issues/7838>