NiceGUI

Hsv.py

April 22, 2025

Hamilton Woods

NiceGUI

Hey fellow developers! Ever wished you could whip up a snazzy web UI for your Python projects without the usual headaches?

Meet NiceGUI – a game-changer that's about to make your life a whole lot easier. Forget wrestling with complex JavaScript frameworks; NiceGUI lets you build interactive web apps using the familiar comfort of Python.

Think of it as a bridge between the elegance of Python and the visual power of the web. And, to top it all off, simply use `ui.run(native=True)` to run as a native desktop app instead of as a web app.

NiceGUI Resources

- Where to get these slides
  - https://github.com/hsvpy/talks/tree/master
- NiceGUI Documentation
  - https://nicegui.io/documentation
- Github repo
  - https://github.com/zauberzeug/nicegui
- Youtube (Turtle Code) Tutorial Channel
  - https://www.youtube.com/watch?v=8ThvEe9Ym5c&list=PLMi6KgK4_mk1xZc45zEBxlByLhpbJK2Uy

NiceGUI

- What is NiceGUI?
- Why NiceGUI? – Part I
- Why NiceGUI? – Part II
- NiceGUI Components
- NiceGUI in Action
- Use Cases for NiceGUI

What is NiceGUI?

- Python-based UI framework for building web-based GUIs
- Backed by three main technologies
  - FastAPI for backend operations
  - Quasar (Vue.js) for frontend interaction
  - Tailwind CSS for styling

Why NiceGUI? – Part I

We at Zauberzeug like Streamlit but find it does too much magic when it comes to state handling. In search for an alternative nice library to write simple graphical user interfaces in Python we discovered JustPy. Although we liked the approach, it is too "low-level HTML" for our daily usage. But it inspired us to use Vue and Quasar for the frontend.

We have built on top of FastAPI, which itself is based on the ASGI framework Starlette and the ASGI webserver Uvicorn because of their great performance and ease of use.

Why NiceGUI? – Part II

- User-Friendly UI Development Tool
- Real-Time Interactivity
- Visual Elements and Layouts
- Integration with External Libraries
- Works with Docker
- No HTML/CSS/JavaScript Required!

NiceGUI Components

- Text Elements – label, link, chat, markdown, etc.
- Controls – button, dropdown, checkbox, toggle, etc.
- Audiovisual Elements – image, overlays, audio, etc.
- Data Elements – table, aggrid, interactive charts, etc.
- Binding Properties – dictionary, variable, storage, etc.
- Page Layout – auto-context, card, grid, splitter, tab, etc.
- Styling and Appearance – Quasar props, Tailwind CSS, mode
- Events – UI, time, keyboard, async, generic, etc.
- Tasks – CPU-bound, I/O-bound, JavaScript, etc.
- Pages & Routing – page, private, layout, navigation, etc.
- Configuration & Deployment – ui.run(), native mode, etc.
- Testing – pytest plugin: simple or modular approach

NiceGUI in Action

- VSCode demos
  - UI Controls (with binding, light and dark modes)
  - UI Controls with Events
  - Page Layout
  - Native Mode
  - Shutdown
  - MasterMind

Use Cases for NiceGUI

- Web Apps (NiceGUI website is written in …)
- Dashboards
- Kiosk Apps
- Interactive Data Presentation and Charting
- UI for Apps on Small Devices