EGU26-18563, updated on 18 Nov 2026
https://doi.org/10.5194/egusphere-egu26-18563
EGU General Assembly 2026
© Author(s) 2026. This work is distributed under
the Creative Commons Attribution 4.0 License.

Obspy + WebAssembly: Running AI-Assisted Seismic Analysis in the Browser

Matthias Meschede, Sylvain Corlay, and Thorsten Beier
QuantStack, Saint-Maur-Des-Fossés (Grand Paris), France (info@quantstack.net)

We succeeded to run Obspy—the popular seismic Python toolkit—entirely in the browser using WebAssembly and a novel, package distribution called
emscripten-forge. This breakthrough eliminates the need for local installations or complex backends for small to medium sized data.

Now, anyone can perform core seismic data analysis—including reading, processing, and visualizing seismological data—directly in their web browser,
on many devices, from desktops to mobile phones. For developers, Obspy's capacities can be embedded in ordinary static web sites that are highly
scalable, without any complex backend to maintain. Additionally, Large Language Models (LLMs) can command Obspy directly in the browser, allowing to build seismic applications driven by natural language.

The implications are profound. The lack of complex installations, backend maintenance, or specialized deployments, drastically simplifies using and
building seismic data analysis tools. Educators can incorporate hands-on seismology into curricula without technical overhead. Researchers can share
apps or notebooks, and explore data without fighting non-reproducible compute environments. Engineers in remote settings gain access to powerful
analytical tools on mobile devices. Users can interact with seismic data in the browser in natural language, lowering the barrier for non-experts and
professionals alike.

We will demonstrate this with JupyterLite, showcasing a fully functional, Jupyter environment with Obspy and scientific Python tools running completely in the browser, without any server and with LLM assistance. We will also highlight the limitations of in-browser execution, particularly regarding multi-threading, memory constraints and compute overhead. All of the demonstrated technology is open source, published under permissive licenses, and therefore available for anyone to use, modify, and build upon.