

Summer of Code

[Ethan Fremen](#) is working on mailman for his [Summer of Code Project](#). The project will use web standards, kid and mod_python to deliver a superior HTTP user experience with enhanced internationalization.

News

- [August 20th Summary](#)
- [June 7th News](#)

Technical Plan

- Build a fully [accessible](#) web interface using xhtml without CSS or JavaScript.
- Enhance that interface with CSS to make it visually intuitive.
- Add JavaScript that provides additional ease of use.
- Modules/Supporting logic modules, to be implemented or borrowed:
 - Converters
 - RFC-2822 into an ElementTree
 - .mbox files into an ElementTree
 - Elemental RFC-2822, mbox into xhtml
 - Elemental mbox into xhtml
 - Handlers (mod_python)
 - feed handler
 - authentication handler
 - fragment handler - provides fragments of the UI and accepts 'fragmented' form submissions for XMLHttp / REST access.
 - Modules
 - glue code to patch the 1-user-per-list situation
 - Filters (mod_python)
 - uniquer - take outgoing xml and render the ids within unique to mailman/the page. Would need helper JS to know how to decouple, but as only the JS really cares about the ids anyway, we can avoid problems. Primary need for this is auto-labeling for attributes on label and providing an ID namespace that doesn't collide with whatever page it's embedded in.
 - bugfr - "bug friend" - performs last minute dom/css/js munging to kludge for broken clients. this way all the inevitable client-specific workarounds don't make their way into the templates.

Notes

I've had to move away from the incremental approach and towards a reimplementation. I'm still working strictly from the existing implementation for feature-set, but because of the templating situation, I'll get there faster if I re-write the templates.

The decision to move to mod_python is based on the need for feed handlers that can maintain a semi-persistent cache and because I get access to Apache's internals. The existing CGI interface will be untouched, with the idea that eventually a wrapper will be found/written that provides backwards compatibility.

[Campaign monitor](#) is an interesting example of what people who run list-like things want from their webui

Existing templating situation

- There's a mishmash of MM- pseudo tags and python str/dict replacement for variable substitutions.
- Many pages are generated by a sort of proto-dom.
- All the pages are mid-90's tag soup, with upper case and lower case tags and invalid html attributes.
- Many pieces of code generate strings with html markup. This is bad form and makes i18n/multiple interfaces

harder.

Timeline

- May 23 - June 21: Rewrite mailman interface using xhtml templates with the following attributes:
 - Provide consistent navigation throughout the Mailman interface
 - Enhance the Administrator's user listing to allow for bulk modification and to highlight users that have been set nomail.
 - Collapse duplicate pages
 - Write pages in seperable, embeddable pieces.
- June 23- June 30: Establish the scope of the glue modules, and write interfaces/stub classes.
- July 3 - 14: Write handlers and RFC2822/ mbox -> ElementTree converter
- July 17 - 21: Fill in glue modules, with knowledge of how the (user)db-in-SQLAlchemy project is going.
- July 24 - 31: Add page-editing, easy support for stylesheet changes, site and list specific themes.
- August 1 - 4: Update documentation to reflect changes.
- August 7 - 11: Add ease-of-use JavaScript/CSS.
- August 14 - 21: Merge branch to trunk, and make a release of some sort.

"a plan is just a list of things that never happen" - Benecio del Torro in The Way of the Gun