How to Contribute to SageMathCloud

William Stein

University of Washington

SageMath, Inc.

November 11, 2015

Screencast of this talk

https://youtu.be/GOuy07Kift4

Goal of talk

Goal: Explain how to work on SageMathCloud.

Motivation: I do around 95% of SMC development:

```
   > git log |grep Author |wc -l
   9763
   > git log |grep "Author: William Stein" |wc -l
   9340
   > git ls-files -z | stack overflow ...
278358 William Stein
  5582 Harald Schilly
  2622 Nicholas Ruhland
  1117 William Jon Nich
   909 Jon Nich ...
```

This must stop!!!

What is SMC?

- **Launch:** I launched SMC in April 2013.
- **Size:** SMC runs on dozens of VM's on Google compute engine. 250K projects made. Over 30K monthly active users. Over 850 simultaneous users.
- **Features:** realtime collaboration, LaTeX editing, Sage worksheets, R, Jupyter notebooks, color terminals, write Fortran/C/C++/Java/Haskell/etc. programs, todo lists, course management, chatrooms.
- **Devel:** SMC can also now be run in a new single-user developer-friendly mode.

The Stack

- **Node.js:** javascript on server side; amazing ecosystem
- **CoffeeScript:** Javascript preparser – no braces and semicolons
- **Engine.io/Primus:** websockets + fallback
- **React.js:** sane declarative user interfaces
- **RethinkDB:** scalable, replicated database that pushes changes (=nosql db + message queue)
- **SageMath:** mathematics
- **Jupyter Notebook:** interactive coding
- **webpack:** bundles all assets of webpage
- **LaTeX:** math typesetting: backend in Linux and mathjax on frontend

Contributing to SMC

- **Use Github:** `https://github.com/sagemathinc/smc`
- **SMC dev:** (1) in an SMC project, (2) on Linux/OS X
- **License:** GPLv3+ compatible.

Tutorial – part 1: setup SMC dev environment (15 min)

1. (2 min) In SMC, open or create a SMC project. On a laptop, install Node.js V5 and RethinkDB, and make sure python=python2.7.
2. (3 min) Open a terminal and clone the repo:

   ```
   git clone https://github.com/sagemathinc/smc
   ```
3. (1 min) Right now you must checkout the smc-in-smc branch (this will change): `cd smc/; git checkout smc-in-smc`
4. (8 min) Build/install everything (uses nearly 1GB disk):

   ```
   cd src; . smc-env; npm run make
   ```
5. (1 min) Start everything (needs more than 1GB RAM to work smoothly):

   ```
   cd dev/project; ./tmux-start-all
   ```

   This uses tmux, but you can instead directly start the three scripts in that directory.

BUGS!!!!!!!

There are several devel bugs right now. I will fix these this week.

- The first time after everything starts up, you will have to stop it all and start it again. (hub websocket server may not properly initialize the first time, when the database takes a long time to get configured.)
- On a laptop, you must do `pip install ~/smc/src/smc_pyutil/` to install various scripts systemwide.
- In an SMC project, *anybody* with a project on the same host could connect to and modify your RethinkDB instance! However, they can't login to your server.
- To test modifications to the sage server, you have to install `smc_sagews` into the project via `pip install --user /path/to/smc/src/smc_sagews/`
- Proxy server **not implemented**: IPython, Latex, project in project in project

Tutorial – part 2: exercises

1. **Change frontend UI:** Help page: change "Support" ⟶ "Mom's favorites".
2. **Change backend hub:** When creating account, change user's name to Super Mom.
3. **Change local hub:** When opening a file, put 'HI MOM!' at the top.
4. **Change console server:** Write 'HI MOM!' to terminal stream whenever opened.
5. **Change Sage server:** Define extra variable in Sage worksheets called MOM.

Change frontend UI

Exercise: Help page – change "Support" ⟶ "Mom's favorites".

1. Open `smc/src/smc-webapp/r_help.cjsx`
2. Change "Support" to "Random links"
3. When you save, webpack will update
4. Refresh help page and see change (react hot loader not yet implemented in SMC)

Someday: `http://gaearon.github.io/react-hot-loader/`

Change backend hub

Exercise: When creating account, change user's name to Super Mom.

1. Open `smc/src/smc-hub/hub.coffee`
2. Note that it is way too big and needs to be refactored...
3. Find the function `create_account`
4. Add this code: `mesg.first_name = "Super"; mesg.last_name = "Mom"`
5. Stop the hub and start it again: (1) use tmux to get to the hub log, hit control+c to stop it. (2) start it again in a separate terminal session, make new account, verify result.

Change local hub

Exercise: When opening a file, put 'HI MOM!' at the top.

1. Open `smc/src/smc-project/local_hub.coffee`
2. Look for lines involving `fs.readFile`, and find `@init(doc:data.toString()...`
3. Change it to `@init(doc:'Hi Mom!'  + data.toString()...`
4. Create a project and open it (NOTE: will get an error on first directory load!!)
5. Open a file

Change console server

Exercise: Write 'HI MOM!' to terminal stream whenever opened.

1. Open `smc/src/smc-project/console_server_child.coffee`
2. Put this line at the bottom: `socket.write("HI MOM!")`
3. Open a terminal (restart project if you had already opened a terminal!). Maybe hit return a few times. See "HI MOM!".

Change Sage server

Exercise: Define extra variable in Sage worksheets called MOM.

1. Open `smc/src/smc_sagews/smc_sagews/sage_server.py`
2. Search in the file for `__SAGEWS__`
3. Add this line after it: `namespace['MOM'] = "hi there!"`
4. (TEMPORARY-BUG) Install sage_server code *into a nested project*: `pip install --user /projects/[your project id]/smc/src/smc_sagews/`
5. Open a Sage worksheet in your nexted project and type MOM (shift+enter) and see if it worked. (If it didn't, restart your sage server.)