Instructions for Contributors

(How to contribute to the relay repository)

# 

If you’re viewing this page, hopefully it’s because you want to help contribute to the Relay project (anyone can!). This is accomplished by ‘cloning’ the relay repository to your local computer, making edits, and then committing the changes back to the repo. Changes are then peer-reviewed, authorized and pushed live.

Contributions include

1. .js files (programming) - Javascript files provide all Relay functionality for both Servers and Clients and can be found throughout the source code.
2. .html files (content) - HTML5 is the Relay content format. All UI and local content is defined in single .html files throughout the source code.
3. .css files (design) - CSS is how one makes HTML5 content look (and act) like anything you can think up. CSS files also define Themes used in Relay allowing new themes to be contributed

# 

# Repository Instructions

# **Install git (the repository manager):**

1. If you already have git, skip this step.
2. [Windows git] Download cygwin setup.exe from <https://www.cygwin.com/> [[x86](http://cygwin.com/setup-x86.exe) or [x64](http://cygwin.com/setup-x86_64.exe)]
3. Install with packages **ssh, git, python, wget, nano**Search for each package and select **install** on each library in the group

**Clone the Relay git repository:**

1. After Cygwin is installed (or using your own console/git), open your command console,   
   browse (command **cd**) to your dev directory,   
   and execute git clone [repo url]:  
   $ *cd [your dev folder]*$ *git clone* [*https://github.com/Protricity/relay*](https://github.com/Protricity/relay)

# Server Instructions

(If you want to run a Relay server locally)

**Install node.js and mongodb (if you want to run the server locally):**

1. Skip this step if you prefer to run the server remotely. You will need your own server to access the html file i.e. http://[your dev server]/relay/index.html
2. Install Node.JS   
   <https://docs.npmjs.com/getting-started/installing-node>
3. Download and Install MongoDB  
   <https://www.mongodb.org/downloads?_ga=1.56315051.1067887463.1444014081#production>  
   Instructions:  
   <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>
4. Init node libraries:  
   $ *cd [your dev folder]*$ *sh init-node.sh*
5. Browse to the relay directory and execute:  
   $ *cd [your dev folder, if you’re still there]*$ *node server.js*
6. Try to access your localhost server:  
   <http://localhost/>

# Testing

**Test in Chrome (edit files via chrome debugger):**

1. Open the index page in chrome on your local server i.e. [http://localhost/](http://localhost:8080/relay-server/) (or your own dev server link)
2. Experiment with the chrome debugger (F12 or Ctrl-Shift-I) which lets you step through client code. This debugger is literally all you need to develop for this project client-side, but feel free to use PHPStorm (way better) or NetBeans, or any javascript IDE  
   Additional instructions on editing project files from your chrome browser debugger: <https://developer.chrome.com/devtools/docs/workspaces>
3. In the debugger, select the ‘sources’ tab, and right click. Select ‘add folder to workspace’  
   Select the relay folder you cloned earlier and chrome should be able to sync up the local and remote asset files. Any file edited in real-time is saved locally and can be committed back to the repo.

**Directory structure:**

1. ***/client*** - Contains all client files (Javascript/HTML5)
   1. ***/app*** - Applications and Non-Core features
   2. ***/channel*** - Chat and Channel features
   3. ***/client*** - Client features and UI
   4. ***/client/themes*** - Client Themes and theme resources
   5. ***/keyspace*** - KeySpace features and database
   6. ***/node\_modules*** - Node.js support files
   7. ***/pgp*** - PGP Encryption features
   8. ***/server/http*** - HTTP Server
   9. ***/server/socket*** - Socket Server

**For Designers:**

1. Open the test page corresponding to a design template in the browser to view  
   ex. <http://localhost:8080/relay-server/command/template/base/pages/home/index.html>
2. Edit the template by modifying associated css/image files and refreshing the page in the browser
3. Use test.html files to test individual template components. Add tests as necessary
4. Commit your changes back to the repo

Please update this file with any questions, improvements on the instructions, or anything else you feel like changing

# **Clone commands:**

$ cd [your dev folder]

$ git clone <https://github.com/Protricity/relay>

## Commit commands:

$ cd [your dev folder]/relay/   
$ git status (shows the status if your commit)  
$ git add [your file path] (add your file to the repo)  
$ git commit -m “fixed the thing, added stuff” (commit to repo with message)  
$ git push origin master (push commit to server)

cygwin is a unix driver for windows that lets you do many of the same things linux/unix can do on the command line. The command line is called the CygWin Terminal