For Relay Contributors

(How to contribute to the relay repository)

# 

Welcome to Relay! Hopefully you’re here because you want to help contribute to *The Relay Project* (thank you). This is accomplished by ‘cloning’ the Relay source repository to your local computer, making edits, and then publish the changes back to the repository. Changes are then peer-reviewed, authorized and published live.

# FEATURE STATUS AND BUG REPORTS

Use this link to check feature status and submit new bug reports:

<https://github.com/clevertree/relay/issues>

CONTRIBUTIONS INCLUDE

* .js files (programming) - Javascript files provide all Relay functionality for both Servers and Clients and can be found throughout the source code.
* .html files (content) - HTML5 is the **Relay** content format. All UI and local content is defined in single .html files throughout the source code.
* .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

(The tools you need to manage a remote repository)

# 

# **INSTALL GIT (the repository manager)**

* If you already have git, skip this step.
* [Win 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)]
* Install with packages **ssh, git, python, wget, nano**Search for each package and select **install** on each library in the group

**CLONE THE REPOSITORY**

* After Cygwin is installed (or using your own console/git), open command console.
* browse (command **cd**) to your dev directory (Like C:\Dev or D:\Dev)  
  $ *cd [your dev folder]*

execute git clone [repo url]:$ *git clone* [*https://github.com/clevertree/relay*](https://github.com/clevertree/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)**

* **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]/index.html
* **Install Node.JS**<https://docs.npmjs.com/getting-started/installing-node>
* **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/>
* **Open your cygwin (or other) console and browse to your dev folder:**$ *cd [your dev folder]*
* **Once you’re in the folder, run .scripts/init-node.sh to initialize node libraries:**$ *sh .scripts/init-node.sh*
* **Run the server:**$ *node server.js*
* **If the ‘node’ command can’t be found, try running it directly:**$ *"C:\Program Files\nodejs\node.exe" server.js*
* **Try to access your localhost server:**<http://localhost/>
* **Optionally add a PHPStorm configuration (example)**  
  Optionally install ‘bash’ plugin for PHPStorm  
  Script: D:/Dev/Server/WWW/relay/server.js  
  Interpreter Path: C:\Program Files\nodejs\node.exe  
  Working Directory: D:\Dev\Server\WWW\relay\

# TESTING

**TEST IN CHROME (edit files via chrome debugger)**

* 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)
* 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 chrome browser debugger: <https://developer.chrome.com/devtools/docs/workspaces>
* 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.

DATABASE STRUCTURE

* ***IndexedDB***
  + ***keyspace***
    - ***content*** - KeySpace content
      * ***path*** - content path, if given
      * ***timestamp*** - content timestamp   
        (usually PGP signature timestamp)
      * ***published*** - status of published content (0, or 1)
    - ***message***- Private Messages
      * ***to\_id*** - recipient PGP ID
      * ***from\_id*** - sender PGP ID

DIRECTORY STRUCTURES

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

FOR DESIGNERS

* Open the test page corresponding to a design template in the browser to view  
  ex. <http://localhost/relay/client/tags/render/test-nav.html>
* Edit the template by modifying associated css/image files and refreshing the page in the browser
* Use test-\*.html files to test individual template components. Add tests as necessary
* Commit your changes back to the repo

HOW TO COMMIT BACK TO THE REPO

# **Open your favorite console (i.e. cygwin) and browse to your dev folder:**

*$ cd [your dev folder]*

# **Clone the repository if you haven’t already:**

$ *git clone* [*https://github.com/clevertree/relay*](https://github.com/clevertree/relay)

\* Contribute by adding or modifing source files in the repository \*

## Check the current repository status (should show your changed files):

$ *git status (shows the status if your commit)*

## Add only the files you want to commit (wildcards allowed):

## *$ git add path/to/file.ext*

## **Commit all added files with a message indicating what you did:**

*$ git commit -m “fixed the thing, added stuff”*

* **Push all changes to the online repository:**

*$ git push origin master*

DONE!

Once pushed, your submitted content will be peer reviewed and eventually [*merged*](https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging) into relay. No further steps are required.

*(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)*