Documentation for Bearocracy 3.0

Bearocracy is written in HTML, CSS, JavaScript, and a JavaScript framework called Angularjs version 1.2.28.

Contents

[home.html 2](#_Toc409379474)

[Moving Things 2](#_Toc409379475)

[Interpreting magic.js 2](#_Toc409379476)

[Mainbanner 3](#_Toc409379477)

[mainToggles and subToggles 3](#_Toc409379478)

[MainController (toggle logic) 3](#_Toc409379479)

# home.html

This is the home page that the user sees. The user does not see any other pages individually. Included within the home page are mainToggles.html, slideshow.html, and subToggles.html. The main toggles include the every button left of the banner. Sub toggles are the buttons that replace the header when it is toggled. “slideshow” is the banner.

# Moving Things

To move things around on the home page, first determine whether what it is that you would like to move is a directive or not. Directives are the custom tags that encase the included html. The 3 directives are <sub-toggles></sub-toggles>, <main-toggles></main-toggles>, and <mainbanner></mainbanner>. Each of these corresponds with an included html document listed above. These directives may be placed anywhere within the HTML tags and are the easiest to move. If what you would like to move is not a directive, you must be very careful and not separate html with its associated javascript. The following are javascript dependencies and their associated DOM elements. The associated DOM element in the middle must be a child of the attributes listed on the left.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Dom Element** | **Reason** |
| ng-show="$storage.banner" | <sub-toggles></sub-toggles> | Toggling of subordinate banner |
| ng-hide="$storage.banner" | <main-toggles></main-toggles>  <mainbanner></mainbanner> | Toggling of main banner |
| ng-controller="MainController"  ng-hide="$storage.stream" | <div id="livestreamPlayer"></div> | Toggling of video (visible and not) |
| ng-hide="$storage.chat" | <!—chat🡪all<!--end chat--> | Toggling chat |

To move the livestream, simply move the div with the id of “livestreamPlayer” wherever necessary.

# Interpreting magic.js

Magic is the angularjs code behind the scenes. The first directive is mainbanner and is the logic behind what happens when the banner is clicked. Additionally, it provides the location of the banner’s html with its templateURL.

The next two directives are simply pointers without logic attached to template (include) urls.

The next item is a controller called “MainController” and is the heart of the application.

## Mainbanner

This is probably the most complex of the directives, it contains a templateURL and a controller. To add a banner, name the banner one higher than the highest banner in the img/banners folder. Then, go to the mainbanner directive in magic.js and change the variable “$scope.numBanners” to the number you just added.

Mainbanner works beginning with the DOM in slideshow.html. On click of a banner, a function called pick() is called sending no parameters. The pick function selects a random number between $scope.numBanners and zero, then sets $scope.banner to be the path to the image plus the number selected. Because there is two-way binding, the value does not have to be set. The value is bound within the dom as {{banner}}.

## mainToggles and subToggles

These are together because in the back-end, they are exactly the same in structure. Each simply points to an HTML document using templateURL.

## MainController (toggle logic)

Main controller holds two things. One are the default local storage default values for whether chat, stream, or the banner is or is not toggled on or off. On first visiting the page, all of these values will be set to zero. If something is toggled off, it is set to one. Whenever any of the toggles are clicked, the local storage value is changed. The DOM elements know whether to be toggled on or off by looking directly at the local storage through this variable.

The second thing this controller holds is the remote to the livestream channel. On clicking the tron or vidya buttons, the switchChannel() function is called. It receives the name of the stream desired as a parameter. To change the desired channel destination of tron or vidya, go to mainToggles.html and change the string input of the switchChannel method calls. Additionally, it will be necessary to change the default channel on home.html in the second to last script tag.

# How the Stream Works

The stream works with the developer API found [here](http://www.livestream.com/userguide/index.php?title=API_for_HTML_Developers#Complete_example). It works by placing two separate scripts in the head, and then placing a div with the id of “livestreamPlayer” wherever it is that one wants the player to be. Changing the channels is explained in the MainController section.

# Things to Note

Within the chat script, the height is set to 95%, this is because of a CSS issue that causes the chat box to break out of its parent div and cause it to dive lower than the page length. This in turn causes a scroll bar.

A backup is located [here](https://github.com/Cameron64/bearocracy). If for whatever reason you need help with the site, feel free to contact me by [email](mailto:cameron64@ymail.com).