Connections

The server and the app are connected via Socket.IO and Laravel Passport. The authentication happens over an API call with a password using Passport. Everything gets handled by Laravel and there shouldn’t be anything to touch there. If you do need to troubleshoot or edit something, read the documentation on the official Laravel website.

Socket.IO controls the reset of the communication between the app and the server. We are using ‘emit’ and ‘on’ states with the socket to trigger the events.

The first event that gets triggered in the website flow is ‘connection’ the apps will all emit this call when they have connected to the server and joined the channel. Next the call will be ‘checkin’ and this is when the tablet has logged in. When a user disconnects they will trigger the “checkout” emit call.

When the game start and you press the start game button on the website the ‘start’ call is emitted from the website. The server then emits ‘mstatus’ to the tablets to tell them their initial state and ‘ready’ to the website to trigger the second modal and the tablets which use it to display the view to start the game.

The server will also receive a ‘preproCalcFin’ call which makes the server take the algorithm data and distribute it to the tablets via emitting ‘preproduce’ and the data required.

When the game starts the server emits ‘go’ which starts the timer, and data starts flowing. The server emits ‘graphData’ to the view with all the variables the graphs need as well as ‘mstatus’ again.

The server also has a ‘reset’ function that gets called when the reset button is pressed and emits a ‘reset’ to the apps and website.

The last thing the server emits is ‘gameEnd’ which sends data from the frontend to the database and resets all the perameters.

Important Files

The important files are:

Socket.js which covers the event triggers and important time based processes.

The .env file has the global constants for Laravel and the IP address for the JavaScript.

database\_manager.js has the queries for the MySQL database.

algo.js has the algorithm data.

Mostly you will be relying on using php artisan and composer commands to do heavy lifting, but these files are the ones you will access the most.