# ASSIGNMENT

Nginx Log Tracking Stash

## About this Assignment

In this assignment, you will be given a situational problem to solve.

We are most interested in your ability to focus on the **core** of the problem and reflect it in the development at the core of the software. You don't have to do a detailed design in this exercise, we are looking for the most **important and fundamental** of the product. The core decisions usually come down to the way in which important data is stored (data structures) and the way that valuable insights are extracted or computed (algorithms).

We also believe that simplicity and reuse of existing patterns is extremely important when doing development. We want to maximize the leverage and reuse of existing technologies, creating as little custom code whenever possible. You should consider the entire NPM and Composer catalog as the reusable building blocks for your design. A good solution starts with understanding the root of the problem and then picking the most important building blocks for the job.

Finally, we are only interested in the technical development skills in your solution.

## Product Definition

We want you to build a real-time log tracking application. This application should be usable by the user via api. We also need a simple webpage for debug and authorization purposes.

## Api Definition

We need 2 route api for this application. A public key and private key should be defined for each user upon registration. Both keys should be accessible on the application footer for logged in users.

### A. Push :

Users should be able push log data information into stash application via this method. Timestamp and log information will be saved by api.

**Example Log information ;**

212.223.102.59 - - [06/Jun/2021:00:57:44 +0000] "GET /hit/?search=3284546 HTTP/1.1" 404 197 "https://www.google.com.tr" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.77 Safari/537.36"

**Route**;

http://xxx.xx.xx.xx/push

**Json Format :**

```
{
        public : $public_key,
        private_key : $private_key,
        data_set : $data_set_name,
        data : $log_data_as_text
}
```

### B. Get :

User should be able to get his individual log data back.

Route;

http://xxx.xx.xx.xx/get

Json Format :

{

public : $public_key

data_set : $data_set_name,

}

### Pages and UI components

### A. Login / Register Screen

Simple registration and login screen.

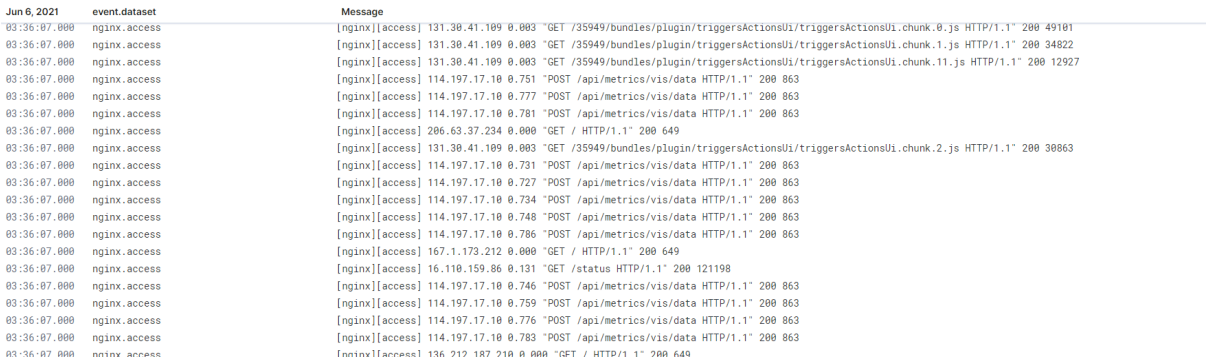
### B. Basic Hamburger Menu

Hamburger menu for operations. There should be at least 5 mock items and required buttons such as logout button.

### C. Log Tracking Page

This page will consist of a single simple datatable/table. New created logs should appear in real-time.

3 Columns are must; Timestamp,  dataset , Log Message



| Jun 6, 2021 | event.dataset | Message |
|---|---|---|
| 03:36:07.000 | nginx.access | [nginx][access] 131.30.41.109 0.003 "GET /35949/bundles/plugin/triggersActionsUi/triggersActionsUi.chunk.0.js HTTP/1.1" 200 49101 |
| 03:36:07.000 | nginx.access | [nginx][access] 131.30.41.109 0.003 "GET /35949/bundles/plugin/triggersActionsUi/triggersActionsUi.chunk.1.js HTTP/1.1" 200 34822 |
| 03:36:07.000 | nginx.access | [nginx][access] 131.30.41.109 0.003 "GET /35949/bundles/plugin/triggersActionsUi/triggersActionsUi.chunk.11.js HTTP/1.1" 200 12927 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.751 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.777 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.781 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 206.63.37.234 0.000 "GET / HTTP/1.1" 200 649 |
| 03:36:07.000 | nginx.access | [nginx][access] 131.30.41.109 0.003 "GET /35949/bundles/plugin/triggersActionsUi/triggersActionsUi.chunk.2.js HTTP/1.1" 200 30863 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.731 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.727 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.734 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.748 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.786 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 167.1.173.212 0.000 "GET / HTTP/1.1" 200 649 |
| 03:36:07.000 | nginx.access | [nginx][access] 16.110.159.86 0.131 "GET /status HTTP/1.1" 200 121198 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.746 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.759 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.776 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 114.197.17.10 0.783 "POST /api/metrics/vis/data HTTP/1.1" 200 863 |
| 03:36:07.000 | nginx.access | [nginx][access] 136.212.187.210 0.000 "GET / HTTP/1.1" 200 649 |

### Technologies Should Be Used

We expect you to complete project by using all of the following technologies;

Real-time communication library (e.g. Socket.io, Pusher)

Webpack

Scss or Less

Composer

A database of your choice

### The Solution

Final product should be sent via github or bitbucket. Also a full structured data dump should exist in the repository.