Error Type: Semantic

Top-Level Description: My search is not being logged to my database's history table.

Data Flow

Frontend

The entry is made here

frontend_display

- [Home](#)
- [Login](#)
- [Sign Up](#)
- [Users](#)
- Logout

First entry!

I send the relevant data from my search form's onChange event and dispatch in my searchHandler function

frontend_search_page_component

```
1   import React, { useState } from 'react';
2   import { useSelector, useDispatch } from 'react-redux';
3   import { Redirect } from 'react-router-dom';
4   import { createHistoryEntry } from '../../store/history_store';
5   import '../Main.css';
6   import './Search_Page.css';
7
8   export const SearchPage = () => {
9       const [userID, setUserID] = useState();
10      const [search, setSearch] = useState('');
11      const [updatedAt, setUpdatedAt] = useState('');
12
13      const dispatch = useDispatch();
14
15      const user = useSelector(state => state.session.user);
16
17      const searchHandler = (e) => {
18          e.preventDefault();
19
20          console.log(userID, search, updatedAt);
21          console.log(typeof(updatedAt));
22
23          dispatch(createHistoryEntry({ userID, search, updatedAt }));
24      }
25
26      return (
27          <div>
28              <form onSubmit={searchHandler}>
29                  <input
30                      type='text'
31                      placeholder='Crawl the web.'
32                      aria-label='Crawl the web.'
33                      onChange={(e) => {
34                          setUserID(user.id)
35                          setSearch(e.target.value)
36                          setUpdatedAt((new Date()).toString())
37                      }}
38                  />
39              </form>
40          </div>
41      )
```

It's sent to my store and processed by my thunk and reducer

frontend_history_thunk

```
//$ thunks
export const createHistoryEntry = (entry) => async dispatch => {
    console.log(entry);
    const response = await fetch('/creepycrawler/history', {
        method: 'POST',
        body: JSON.stringify(entry)
    })

    if (response.ok) {
        const newEntry = await response.json();
        dispatch(createHistory(newEntry));
        return newEntry;
    }
}
```

frontend_history_reducer

```
//$ reducers
const initialState = {}
export const historyReducer = (state = initialState, action) => {
    let newState;
    switch (action.type) {
        case CREATE_HISTORY:
            newState = Object.assign({}, state);
            newState[action.payload.id] = action.payload;
            return newState;
        default:
            return state
    }
}
```

By this point, I'm receiving all the values I expect. Initially, I faced an issue with new Date() unexpectedly returning an object albeit appearing as a string; however, that was rectified. Here is what my console looks like:

frontend_console

```
▼ action session/SET_USER @ 07:08:16.353                                    redux-logger.js:1
    prev state ▸ {session: {…}, history: {…}}                                redux-logger.js:1
    action      ▸ {type: 'session/SET_USER', payload: {…}}                   redux-logger.js:1
    next state ▸ {session: {…}, history: {…}}                                redux-logger.js:1
  1 'dsadsad' 'Thu Oct 21 2021 07:08:17 GMT-0700 (Pacific Daylight Time)'    Search Page.js:20
    string                                                                   Search Page.js:21
                                                                             history store.js:29
    ▸ {userID: 1, search: 'dsadsad', updatedAt: 'Thu Oct 21 2021 07:08:17 GMT-0700 (Pacific Daylight Time)'}
▼ action history_store/CREATE_HISTORY @ 07:08:18.232                         redux-logger.js:1
    prev state ▸ {session: {…}, history: {…}}                                redux-logger.js:1
    action      ▸ {type: 'history_store/CREATE_HISTORY', payload: {…}}       redux-logger.js:1
    next state ▸ {session: {…}, history: {…}}                                redux-logger.js:1
```

Backend

bug manifestation

This route is hit.

backend_history_routes

```python
@history_routes.route('/', methods=['POST'], strict_slashes=False)
def add_history_entry():
    """Add a search entry to history."""
    form = SearchForm()
    print('HITTTTT')
    print('form backend user_id:', form.data['user_id'])
    print('form backend search:', form.data['search'])
    print('form backend updated_at:', form.data['updated_at'])
    if form.validate_on_submit():
        parsed_js_date = form.data['updated_at'].split('-')[0]
        # converted_js_date = datetime.strptime(parsed_js_date, '%a %b %d %Y %H:%M:%S %Z')

        history_entry = History(
            user_id=form.data['user_id'],
            search=form.data['search'],
            updated_at=datetime.strptime(parsed_js_date, '%a %b %d %Y %H:%M:%S %Z')
        )
        db.session.add(history_entry)
        db.session.commit()
        return {"message": "successful"}
    return { "message": "unsuccessful" }
```

The print statement doesn't make it inside of the conditional, as evinced in my terminal.

backend_terminal

```
* Debugger PIN: 393-098-877
* Detected change in '/Users/alejandrogrant/Desktop/code_playgrounds/app-acad-proj
* Restarting with stat
* Debugger is active!
* Debugger PIN: 393-098-877
Outside is hit?
127.0.0.1 - - [21/Oct/2021 07:54:08] "POST /creepycrawler/history HTTP/1.1" 200 -
```

This makes me think that the form may not be validating as expected.

possible bug origin

The route summons the following form

backend_search_form

```python
"""Search form creation.

Receives form data from backend routes.
Processes data (validation, organization, etc.).
Holds the relevant data for backend routes.
"""


from flask_wtf import FlaskForm
from wtforms import StringField, IntegerField
from wtforms.validators import DataRequired, ValidationError

from app.api.routes.user_routes import user


class SearchForm(FlaskForm):
    """Responsible for processing search queries."""

    user_id = IntegerField('user_id', validators=[DataRequired()])
    search = StringField('search', validators=[DataRequired()])
    updated_at = StringField('updated_at', validators=[DataRequired()])


class VisitForm(FlaskForm):
    """Responsible for processing link clicks."""

    user_id = IntegerField('user_id', validators=[DataRequired()])
    visit = StringField('visit', validators=[DataRequired()])
    user_id = StringField('updated_at', validators=[DataRequired()])
```

here are relevant snippets from my database tables

backend_histories_table

id	user_id	search	visit	created_at	updated_at

backend_users_table

media	created_at	updated_at
images/user_icon.jpg	Oct 20 10:51:13	Oct 20 10:51:13