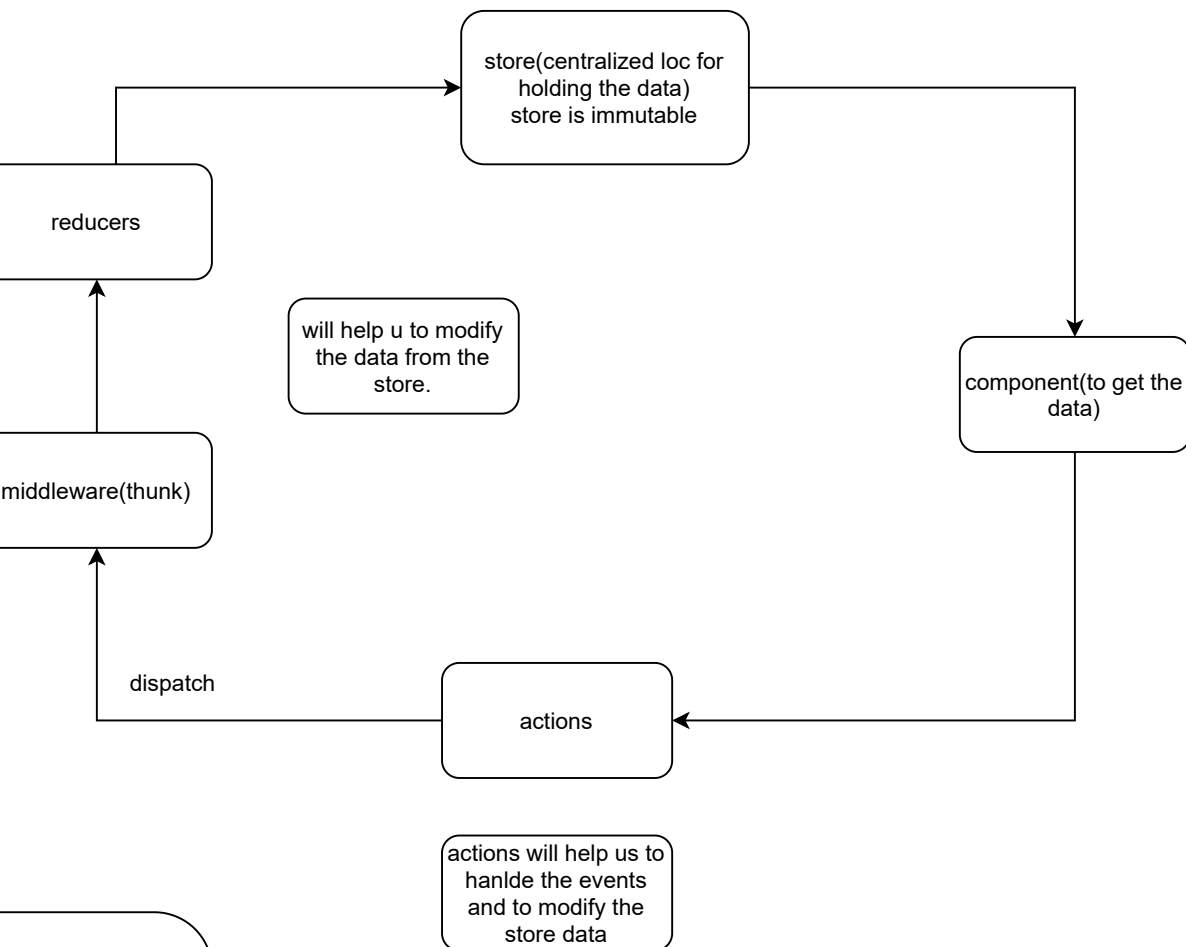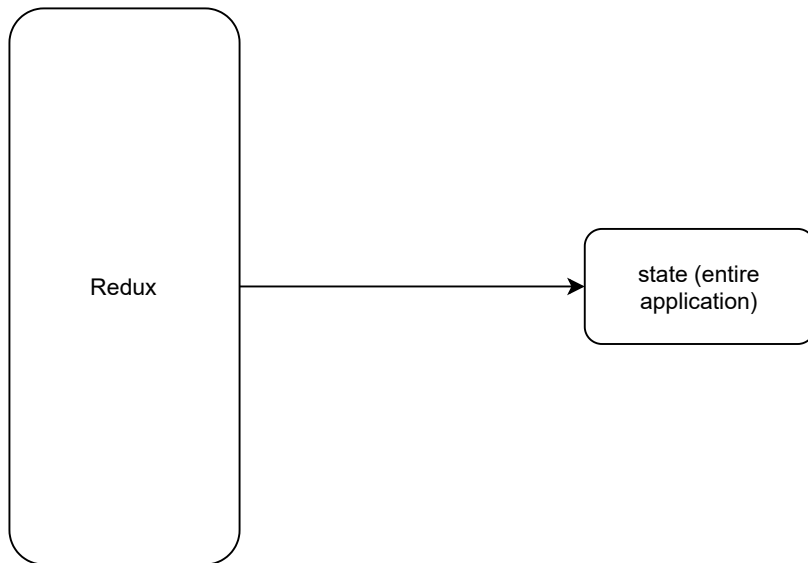Register
login
dashboard
profile(CRUD)
exp(CRUD)
edu(CRUD)
post(CRUD)
comment(CURD)
common components
creation and heroku
app deployment)

middleware that
allows you to return
the functions , rather
than just actions
within the redux.  This
allows for delayed
actions , including
work promises.

main use case for this middleware is
for handling actions that might not be
synchronous . ex axios to send a
GET request , redux thunk alllows us
to dispatch those actions
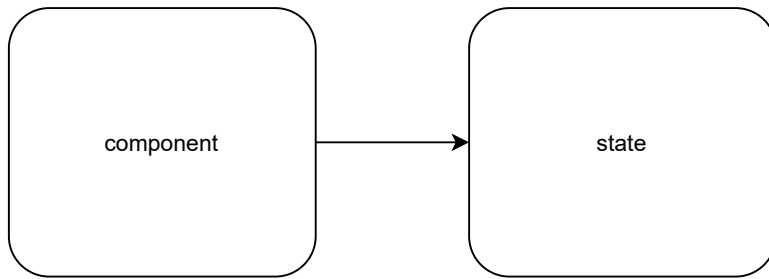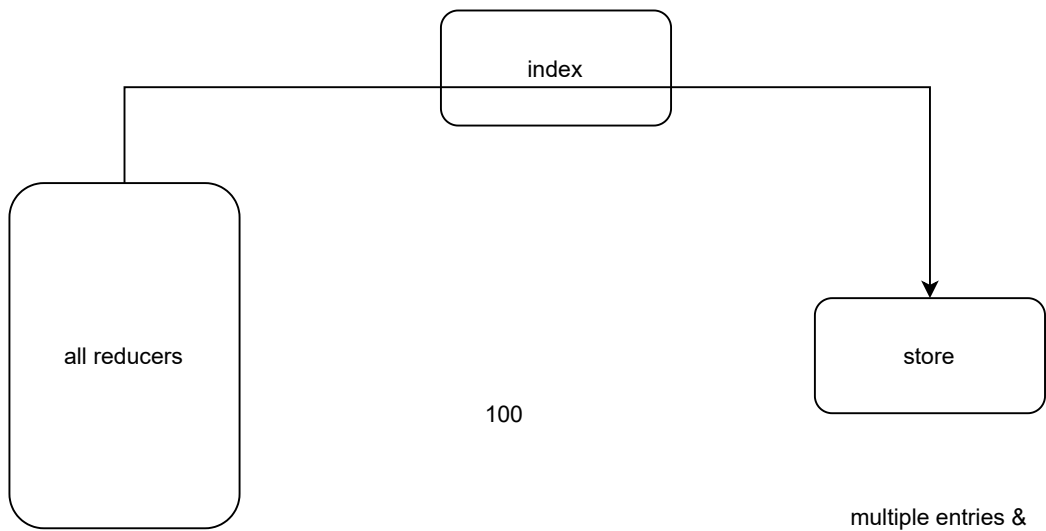asynchronously and resolve each

```
                    ┌──────────────┐
                    │              │
                    │              │              ┌──────────────────┐
                    │              │              │  state (entire   │
                    │    Redux     │─────────────▶│  application)    │
                    │              │              └──────────────────┘
                    │              │
                    │              │
                    └──────────────┘

                         ┌────────────────────────┐
                         │  store(centralized loc for │
                    ┌───▶│     holding the data)     │───────┐
                    │    │     store is immutable    │       │
                    │    └────────────────────────┘       │
  ┌──────────┐      │                                      │
  │ reducers │      │                                      │
  └──────────┘      │     ┌────────────────────┐           │
        ▲           │     │ will help u to modify │          │
        │           │     │  the data from the   │          │
        │           │     │       store.         │          ▼
        │           │     └────────────────────┘   ┌──────────────────┐
  ┌──────────────┐  │                               │ component(to get the │
  │middleware(thunk)│ │                              │       data)        │
  └──────────────┘  │                               └──────────────────┘
        ▲           │                                         │
        │           │                                         │
        │ dispatch  │     ┌──────────────┐                    │
        │           └─────│   actions    │◀───────────────────┘
        └─────────────────└──────────────┘

                          ┌────────────────────────┐
                          │  actions will help us to  │
                          │   hanlde the events      │
                          │  and to modify the       │
                          │     store data           │
                          └────────────────────────┘
```

promise that gets returned.
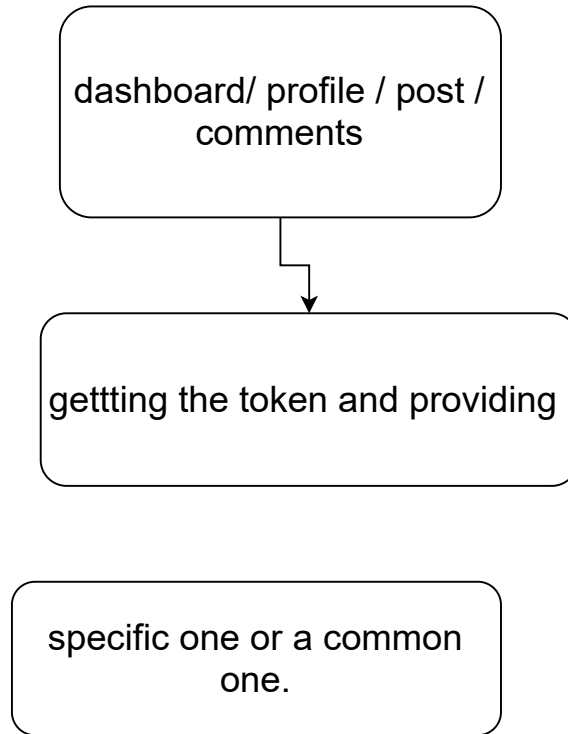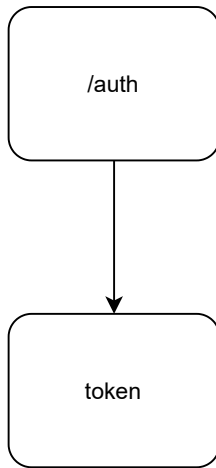
1. redu
2. stor
3.
4. usage of acti
ready the comp
store

ucer spec.
e creation
action
on in component &
ponent to use redux
e content

1. reducer work
2. action
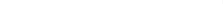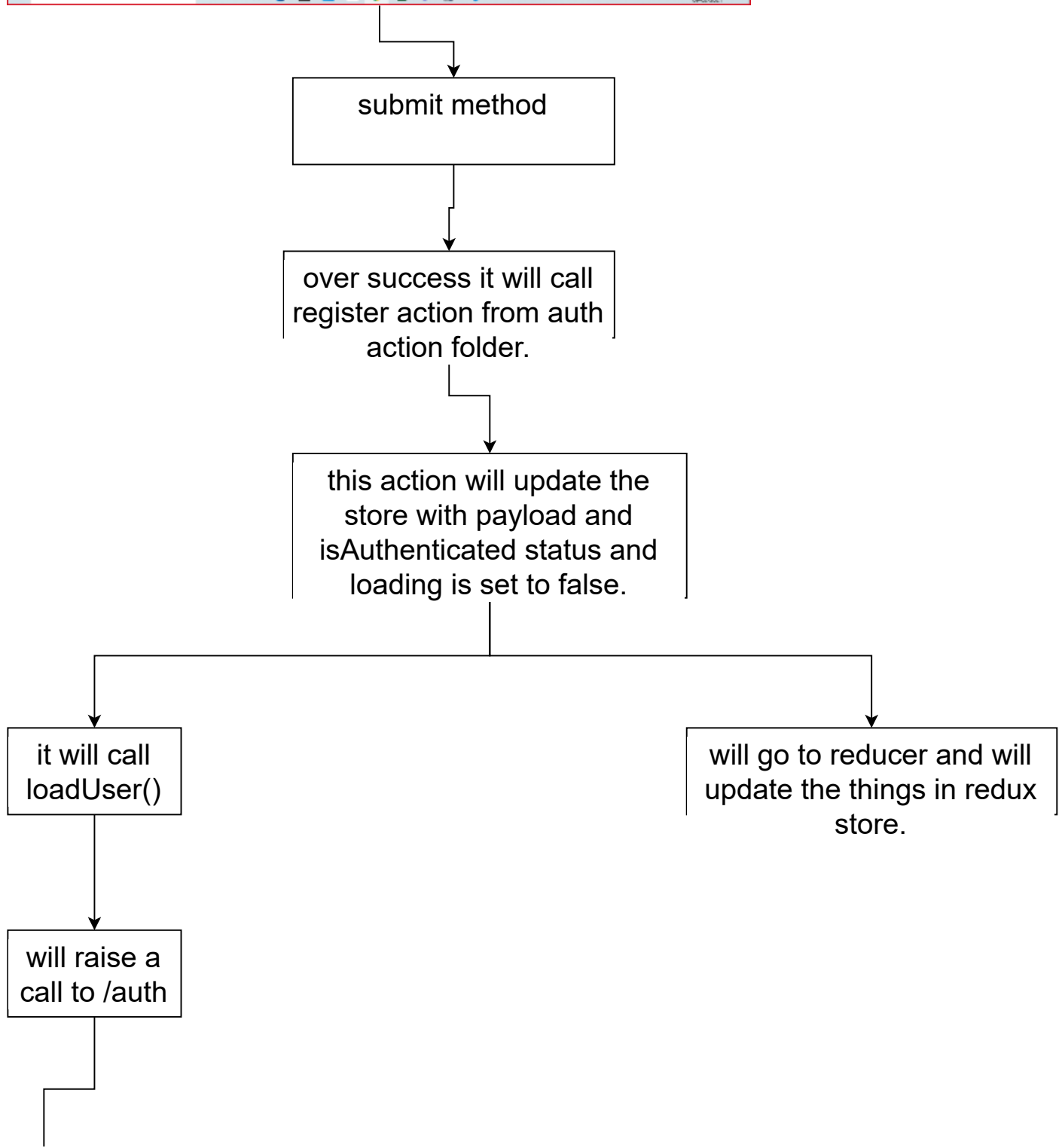3. component
consuming action
stuff and data from
store

```
┌─────────────────┐              ┌─────────────────┐
│                 │              │                 │
│                 │              │                 │
│    component    │─────────────▶│      state      │
│                 │              │                 │
│                 │              │                 │
└─────────────────┘              └─────────────────┘
```

index

all reducers

store

100

multiple entries &
we should do the
less manipulation in
store.js

```
/auth
```

```
token
```

```
dashboard/ profile / post /
comments
```

```
gettting the token and providing
```

```
specific one or a common
one.
```

will perform
the rest call
@the client

axios

submit method

over success it will call register action from auth action folder.

this action will update the store with payload and isAuthenticated status and loading is set to false.

it will call loadUser()

will go to reducer and will update the things in redux store.

will raise a call to /auth
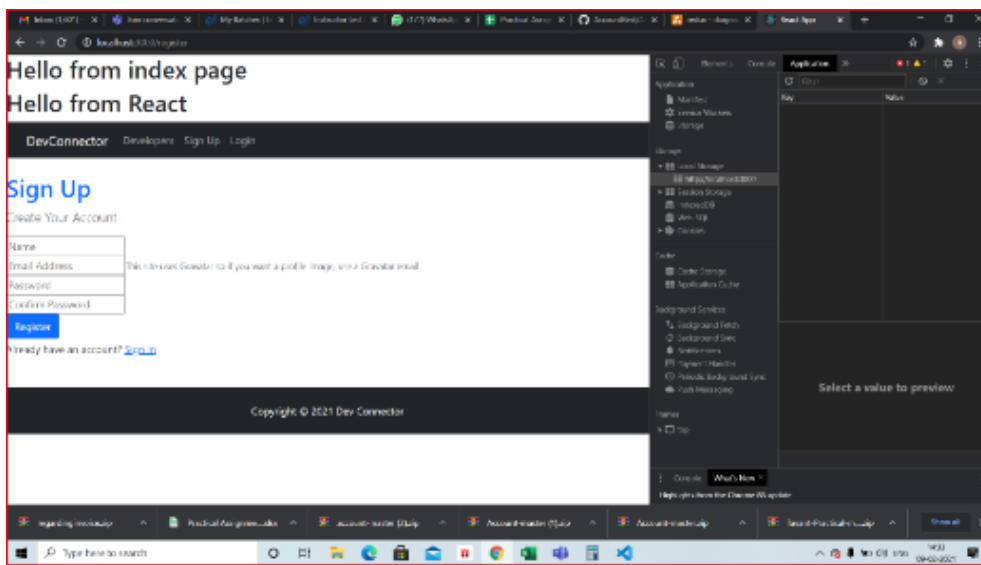
setAuthToken()
from **store**/app.js

rest call

(fail) will
throw error

(success)
1. it will update
ur store (with
token and user
details )