

Component Plan

Index.html (contains meta data and pulls in root) => pulls in index.js (incorporates any providers) => pulls in app.js:

NavBar (component – dropdown menu for mobile)

Logged out:

- Link to **Plan (show and explain)** Route
- Link to **Take Action (show and explain)** Route
- Link to **Sign in** Route
- Link to **Sign up** Route

Logged in:

- Link to **Plan** Route
- Link to **Take Action** Route
- Link to **Labels** Route
- Link to **Profile** Route
- **handleSignout** `post:dj-rest-auth/logout/`

Main Container:

The following routes are accessed in the logged out Nav Bar

- Route to **Sign in form** (pages-auth, `post:dj-rest-auth/login/`)
- Route to **Sign up form** (pages-auth, `post:dj-rest-auth/registration`)
- Route to **Plan (show and explain)** (pages-newUser)
- Route to **Take Action (show and explain)** (pages-newUser)

The following routes are accessed in the logged in Nav Bar

- Route to **Plan** (pages-plan):
 - `handleClick`: New focus => re-direct **Focus**:props('newFocus')
 - `get:focus` filtered:user==owner, ordered-by:rank
 - `handleClick`: Order focus areas => redirect **Order**:props('focus', focus-list=[{focus}, {focus}])
 - o **Focus-highlights** (pages-focus)
 - If miscellaneous rendered differently
 - `handleClick`: Go => re-direct **Focus**:props(focus.id) OR **Miscellaneous**
 - `get:goal` filtered:user==owner & focus==focus & parent==false)
 - **Goal-highlights** (pages-goals)
 - `handleClick`: play/pause => `patch:goal/id` change active
 - if goal.children:
 - `get:goal` filtered:parent=goal.id
 - **Goal-highlights** (pages-goals)
 - Route to **TakeAction** (pages-action)
 - `handleClick`: add tasks => redirect **Add task**
 - o **Reset** (component)
 - o **Backlog** (pages-action)
 - State: order-by = ''
 - `handleClick`: OrderBy => update state order-by

```
get:tasks filter:owner==user & active=true ordered-by: state.order-by
```

- **Task** (pages-tasks) props: 'backlog'
handleClick: today checkbox => **put:task/id** today=true, **re-render**: self & today BACKLOG SPECIFIC
handleClick: task => redirect **Plan** props (focus.id, goal.id, task.id)
 - **Focus-image** (components)
 - **Label** (components)
 - **task-info** (pages-tasks)
 state: goal = "", deadline = "", repetition-info = "", message = ""
 LOGIC to handle alerting users if deadlines near.
 if task.repeated_id **get:repeated/id**
 if repeated.week-end has passed => **put:repeated/id**
 updating tracking info => update state
 LOGIC to handle alerting users if repeated task needs completing today.
- **Today** (pages-action)
get:tasks filter: user==owner & today=true & achieved=false, ordered-by: order
HandleClick: Order-today => redirect **Order**: props('today', today-list=[{task}, {task}])
 - **Task** (pages-tasks) props: 'today'
handleClick: completed => **put:task/id** achieved=true, **re-render**: self & completed TODAY SPECIFIC
 SEE BACKLOG – TASK for further details
- **Completed** (pages-action)
get:tasks filter: user==owner & achieved=true, ordered-by: updated_at
 - **Task** (pages-tasks) props: 'completed'
handleClick: completed => **put:task/id** achieved=false, **re-render**: self & completed COMPLETED SPECIFIC
 SEE BACKLOG – TASK for further details
- Route to **Label-page** (pages-labels)
 - **Return-cross** (components)
handleClick => redirect last history item
 - **Labels-list** (pages-labels)
 State: labels-list=[]
Get:labels filter: owner==owner => update state
 - **Label-detail** (pages-labels)
 State: type=view/edit, label={}
 Render based on state.
handleClick: edit => update state type=edit
handleClick: delete => **Delete-confirm**(component) => update label list
 - **Label** (component)
 - **Label-edit** (pages-labels)
handleSubmit => **put:label/id** => update label list

- **Label-create** (pages-labels)
 handleSubmit => post:label => clear form & update label list
- Route to Profile (pages-profile)

The following routes are not accessed from the nav bar:

- Route to **Focus** (pages-plan)
 Props: Focus.id, newFocus, goal.id, goal-type, task.id, nest-type
 - **Return-cross** (components)
 handleClick => redirect last history item
 - **Focus-detail** (pages-focus)
 Props: Focus.id, create, view, edit
 State: type=create/view/edit, focus={}
 If view or edit get:focus/id => update state focus
 Render based on state.
 - **Focus-create**(pages-focus)
 handleSubmit => post:focus => update state type=view
 handleClick:cancel => redirect **Plan**
 - **Leave-page**(component)
 - **Focus-view**(pages-focus)
 handleClick:edit => update state type=edit
 handleClick:delete => **Delete-confirm**(component)
 - **Focus-edit**(pages-focus)
 handleSubmit => put:focus/id => update state type=view
 handleClick:cancel => update state type=view
 - **Leave-page**(component)
 - **Goal-list** (pages-goals)
 get:goals filter:owner==user & focus==focus & parent==false
 map(
 - goal.name
 - handleClick:goal => update state goal-id=goal.id (causing goal detail to re-render)
 - handleClick:add goal => update state goal-type=create (causing goal detail to re-render)
 - if goal.children:
 - get:goals filter:parent=goal.id ...
 - **Goal-detail**(pages-goal)
 Props: state:goal-id state:goal-type state:nest-type
 State: goal={}
 If view or edit get:goal/id => update state goal-id, update state nest-type
 Render based on state.
 - **Goal-view**(pages-goal)
 handleClick:edit => update state goal-type=edit
 handleClick:delete => **Delete-confirm**(component)
 - **Goal-edit**(pages-goal)
 handleSubmit => put:goal/id => update state goal-type=view
 handleClick:cancel => update state goal-type=view

- **Leave-page**(component)
- **Goal-create**(pages-goal)
 - Update state nest-type=create
 - handleSubmit** => **post:goal** => update state goal-type=view
 - handleClick**:cancel => redirect last history item
- **Plan-nesting**(pages-plan)
 - Props state:nest-type
 - Render depending on nest-type
 - **Set-nest**(pages-plan)
 - Props state:nest-state
 - handleClick**:tasks => update state nest-type=tasks
 - handleClick**:goals => {
 - update state nest-type=goals
 - push:goal/id/ parent=true}
 - **Nested-tasks**(pages-tasks)
 - MOBILE VIEW??
 - get:tasks** filter:goal==goal
 - **Task-detail**(pages-task)
 - Props: task, view, edit, labels
 - State: type=create/view/edit/label, task={}, repeated={}
 - If view or edit **get:task/id** => update state task
 - Render based on state.
 - **Task-view**(pages-task)
 - If task.repeated & repeated=empty
 - get: repeated/id**
 - **Task-labels**(component)
 - Displays labels contained within the task object
 - **Task-choices**(component)
 - handleClick**:edit => update state type=edit
 - handleClick**:addlabels => update state type=labels
 - handleClick**:delete => **Delete-confirm**(component)
 - **Task-edit**(pages-task)
 - If task.repeated **get: repeated/id**
 - handleSubmit** =>
 - put:task/id**
 - if new repeated **post:repeated**
 - if edit repeated **put:repeated/id**
 - if delete repeated **delete:repeated/id**
 - => update state type=view
 - handleClick**:cancel => update state type=view
 - **Leave-page**(component)
 - **Add-label**(pages-label)
 - **Current-labels**(pages-label)
 - Rendered using labels contained within the task object
 - handleClick**:delete => **patch:task/id** update labels field to remove label

POTENTIALLY TRICKY CODING ABOVE

update state task.labels

- **Label-form**(pages-label)

handleConfirm => **patch:task/id** update labels

field to remove label => update state task.labels

handleClick:alldone=> update state type=view

- **Task-create**(pages-task)

handleSubmit => **post:task** => New task needs adding to list
and form clearing

handleClick:cancel => clear form

- **Leave-page**(component)

- **Nested-goals**(pages-goals)

NEED TO THINK ABOUT!

- **dayToDay-tasks**(pages-tasks)

MOBILE VIEW??

get:tasks filter:nogoal & focus==focus

- **Task-detail**(pages-task)

SEE NESTED TASKS FOR DETAILS

- **Task-create**(pages-task)

SEE NESTED TASKS FOR DETAILS

- Route to **Miscellaneous** – view (pages-plan)

- **Return-cross** (components)

handleClick => redirect last history item

- **Task-list** (pages-tasks)

get:tasks filter:nogoal & focus==miscellaneous

- **Task-detail**(pages-task)

SEE NESTED TASKS FOR DETAILS

- **Task-create**(pages-task)

SEE NESTED TASKS FOR DETAILS

- Route to **Order** (pages-order)

Props (today, focus, todaylist, focuslist)

If Focus

Map list

- **Focus-highlights**(pages-focus)

handleSubmit => for each focus **patch:focus/id** updating rank

If Today

Map list

- **Task-view**(pages-task) – No Task-choices

handleSubmit => **delete:order** (all order instances for current user) => for
each task **post:order**

- Route to **Add task** (pages-task)

handleSubmit => **post:task** if repeat=True **post:repeated** => re-direct **Take
Action**

handleClick:cancel => re-direct **Take Action**

- **Leave-page**(component)

- Route to **Change username** (pages-auth)(**handleSubmit** => **put:dj-rest-
auth/user/**)

- Route to **Change password** (pages-auth)(handleSubmit => post:dj-rest-auth/password/change/)
- Route to **Delete account** (pages-??)

Footer (component)

Components

Delete-confirm(component):

Renders a confirmation message – Delete or return options. Return => removes message. Delete => delete http request sent. On completion success message rendered. NEEDS FURTHER THINKING ABOUT!!

Return-cross (components):

Renders cross and handles clicking of the cross => redirect to the last entry in history.

Leave-page(components):

Triggered when component tries to re-render. Checks form is empty. If empty re-render goes ahead. If form contains data, renders a confirmation message, users will need to confirm they are happy to leave data unsaved. NEEDS THINKING ABOUT!!

Focus-image (components):

Renders the focus image

Label (components):

Renders a label

task-info (components):

Renders the additional task information, including tracking data

Contexts (provide link to functionality/data needed by multiple components)

CurrentUserContext (context)

access to current user and also functionality for sorting access tokens

Hooks

useRedirect (hook)

Reset (hook??):

delete:tasks filtered:user==owner & achieved=true

put:habits filtered:user==owner & today=true, change achieved to false and today to false

put:tasks filter:user=owner & today=true, change today to false)

Re-render all of take action