Flux Cycles

Flux loops are organized by data type. Under each data type, there may be sub-categories, and each action is listed with the sequence of events that result from its invocation, ending with the API or store. Finally, store listeners are listed at the end.

You should be able to use this document trace an **action** starting with where it was invoked, through the **API/store** involved, and finally to the **components** that update as a result. This is important because once you start implementing your flux loops, that's precisely what you'll need to do.

Event Cycles

Events API Request Actions

- `fetchAllEvents`
  i. invoked from `EventsIndex` `didMount`/`willReceiveProps`
  ii. `GET /api/events` is called.
  iii. `receiveAllEvents` is set as the callback.
- `createEvent`
  . invoked from new event button `onClick`
  i. `POST /api/events` is called.
  ii. `receiveSingleEvent` is set as the callback.
- `fetchSingleEvent`
  . invoked from `EventDetail` `didMount`/`willReceiveProps`
  i. `GET /api/events/:id` is called.
  ii. `receiveSingleEvent` is set as the callback.
- `updateEvent`
  . invoked from `EventForm` `onSubmit`
  i. `POST /api/events` is called.
  ii. `receiveSingleEvent` is set as the callback.
- `destroyEvent`
  . invoked from delete note button `onClick`
  i. `DELETE /api/events/:id` is called.
  ii. `removeEvent` is set as the callback.

Events API Response Actions

- `receiveAllEvents`

i. invoked from an API callback.
ii. `Event` store updates `_events` and emits change.

- `receiveSingleNote`

. invoked from an API callback.
i. `Event` store updates `_events[id]` and emits change.

- `removeEvent`

. invoked from an API callback.
i. `Event` store removes `_events[id]` and emits change.

Store Listeners

- `NotesIndex` component listens to `Note` store.
- `NoteDetail` component listens to `Note` store.

Showtime Cycles

Showtimes API Request Actions

- `fetchAllShowtimes`

i. invoked from `ShowtimesIndex` `didMount`/`willReceiveProps`
ii. `GET /api/showtimes` is called.
iii. `receiveAllShowtimes` is set as the callback.

- `createShowtime`

. invoked from new notebook button `onClick`
i. `POST /api/showtimes` is called.
ii. `receiveSingleShowtime` is set as the callback.

- `fetchSingleShowtime`

. invoked from `ShowtimeDetail` `didMount`/`willReceiveProps`
i. `GET /api/showtimes/:id` is called.
ii. `receiveSingleShowtime` is set as the callback.

- `updateShowtime`

. invoked from `ShowtimeForm` `onSubmit`
i. `POST /api/showtimes` is called.
ii. `receiveSingleShowtime` is set as the callback.

- `destroyShowtime`

. invoked from delete notebook button `onClick`
i. `DELETE /api/showtimes/:id` is called.
ii. `removeShowtime` is set as the callback.

Showtimes API Response Actions

- `receiveAllShowtimes`

i. invoked from an API callback.
ii. `Notebook` store updates `_notebooks` and emits change.

- `receiveSingleShowtime`
    . invoked from an API callback.
    i. `Showtime` store updates `_showtimes[id]` and emits change.
- `removeShowtime`
    . invoked from an API callback.
    i. `Showtime` store removes `_showtimes[id]` and emits change.

Store Listeners

- `ShowtimesIndex` component listens to `Showtime` store.

Ticket Cycles

Tickets API Request Actions

- `fetchAllTickets`
    i. invoked from `TicketsIndex` `didMount`/`willReceiveProps`
    ii. `GET /api/tickets` is called.
    iii. `receiveAllTickets` is set as the callback.
- `createTicket`
    . invoked from new notebook button `onClick`
    i. `POST /api/tickets` is called.
    ii. `receiveSingleTicket` is set as the callback.
- `fetchSingleTicket`
    . invoked from `TicketDetail` `didMount`/`willReceiveProps`
    i. `GET /api/tickets/:id` is called.
    ii. `receiveSingleTicket` is set as the callback.
- `updateTicket`
    . invoked from `TicketForm` `onSubmit`
    i. `POST /api/tickets` is called.
    ii. `receiveSingleTicket` is set as the callback.
- `destroyTicket`
    . invoked from delete notebook button `onClick`
    i. `DELETE /api/tickets/:id` is called.
    ii. `removeTicket` is set as the callback.

Tickets API Response Actions

- `receiveAllTickets`
    i. invoked from an API callback.
    ii. `Ticket` store updates `_tickets` and emits change.
- `receiveSingleNotebook`
    . invoked from an API callback.
    i. `Ticket` store updates `_tickets[id]` and emits change.

- `removeTicket`
  .
  invoked from an API callback.
  i. `Ticket` store removes `_tickets[id]` and emits change.

Store Listeners

- `TicketsIndex` component listens to `Tickets` store.