

Story UFC-146: Implementing Phase 1

As a web user, I want to be able to use the app while I'm offline.

This story pertains to completing the entirety of Phase 1. As such, it encompasses making the app function transparently when the user is offline.

Acceptance Criteria

For reference, the following are the acceptance criteria for this ticket:

- 1, 2, and 3 of [Epic UFC-145: Offline First](#)

Design

See the Design (Finalized) section of the [Phase 1: Transparent Offline First](#) doc.

Tasks

- Add the healthcheck route to the backend API.
- Create the `OfflineRequestSlice`.
 - Create all of the new actions.
 - `effect/start`, `effect/success`, `effect/failure`
 - `rollback/start`, `rollback/success`, `rollback/failure`
 - Create the new saga wrappers.
 - **commit** saga wrapper
 - Tests
 - **effect** saga wrapper
 - Tests
 - **rollback** saga wrapper
 - Tests
 - Re-use the `RequestSlice` reducer and state.
 - Re-use the `RequestSlice` selectors.

- Create the `OfflineRequestSliceSet`.
 - Extend the `RequestSliceSet`.
- Create the `OfflineRequestManager`.
 - Create all of the actions.
 - `enqueue`
 - `dequeue`
 - `incrementAttempts`
 - `update_connectivity`
 - `process_queue`
 - Create the reducer with all of the state.
 - `effectsById`
 - Tests
 - `queue`
 - Tests
 - `isOnline`
 - Tests
 - Create the selectors.
 - `getEffectById`
 - `getQueueHead`
 - `getIsOnline`
 - Create the sagas.
 - `processQueue saga`
 - Tests
 - `connectivityCheck saga`
 - Tests
 - `triggerQueueWhenOnline saga`

- Tests
- Register the `OfflineRequestManager` slice.
- Register the `OfflineRequestManager` sagas.
- Add the `OfflineRequestManager` slice to `redux-persist`.
- Add native event listeners for triggering the connectivity check.
- Migrate all of the regular `RequestSlices` (except those concerning authentication).
 - Accounts
 - `fetchAll`
 - `commit`
 - `effect`
 - `rollback`
 - `create`
 - `commit`
 - `effect`
 - `rollback`
 - `update`
 - `commit`
 - `effect`
 - `rollback`
 - `destroy`
 - `commit`
 - `effect`
 - `rollback`
 - Transactions
 - `create`
 - `commit`

- effect
 - rollback
 - update
 - commit
 - effect
 - rollback
 - destroy
 - commit
 - effect
 - rollback
- Import Profiles
 - fetchAll
 - commit
 - effect
 - rollback
 - create
 - commit
 - effect
 - rollback