

Documentation Flow load more and render Header

=0=

1. Call API get list and pull to refresh:

```
props.getHistoryPointRequest({
  skip: 0,
  take: 20,
});
```

2. Function handle load more:

```
const handleLoadMore = () => {
  if (!props.isLoading) {
    props.getHistoryPointRequest({
      skip: props.skip + 20,
      take: 20,
    });
    setIsFirst(false);
  }
};
```

3. Render FlatList: *// Disable Scroll Indicator*

```
<FlatList
  refreshControl={
    <RefreshControl refreshing={isRefresh} onRefresh={onRefresh} />
  }
  showsVerticalScrollIndicator={false}
  data={props.listHistoryPoint}
  renderItem={({item, index}) => renderItem(item, index)}
  keyExtractor={item => item.transactionDateTime + item.Reward}
  onEndReachedThreshold={0.1}
  onEndReached={handleLoadMore}
  ListEmptyComponent={() =>
    !props.isLoading ? (
      <Text style={styles.noData}>Chưa có lịch sử.</Text>
    ) : null
  }
  ListFooterComponent={() =>
    !isRefresh &&
    props.isLoading &&
    (props.total === '' ||
      props.total > props.listHistoryPoint.length) ? (
      <ActivityIndicator
        size="large"
        color="#8a8a8a"
        style={{padding: 30}}
      />
    ) : null
  }
/>
```

4. Sagas.js:

```
if (response.status === 200) {
  const {transactions, totalCount} = response.data;
  const {skip} = obj.payload;

  yield put(
    getHistoryPointSuccess({
      items: transactions,
      total: totalCount,
      skip,
    }),
  );
} else {
  yield put(getHistoryPointFailure());
}
```

You, 20 days ago • create feature history point

5. Reducers.js:

```
case GET_HISTORY_POINT_SUCCESS: {
  let {listHistoryPoint} = state;

  return {
    ...state,
    listHistoryPoint:
      payload.skip !== 0
      ? [...listHistoryPoint, ...payload.items]
      : payload.items,
    skip: payload.skip,
    total: payload.total,
  };
}
```

II Render Header

1. Import image background and ReactUtils

```
import {ReactUtils} from 'utils';  
import backgroundView from '../../assets/images/background.png';
```

2. Custom Header

```
useLayoutEffect(() => {  
  ReactUtils.headerCustom(  
    props.navigation,  
    'Lịch sử điểm Utop',  
    function eventCustom() {  
      props.navigation.goBack();  
    },  
  );  
}, [props.navigation]);
```

3. Render Header

```
return (  
  <View style={{flex: 1}}>  
    <Image source={backgroundView} style={styles.styleImageHeader} />  
  </View>  
)
```

```
styleImageHeader: {  
  alignItems: 'center',  
  position: 'absolute',  
  height: scale(32),  
  resizeMode: 'stretch',  
  width: Dimensions.get('window').width,  
},
```