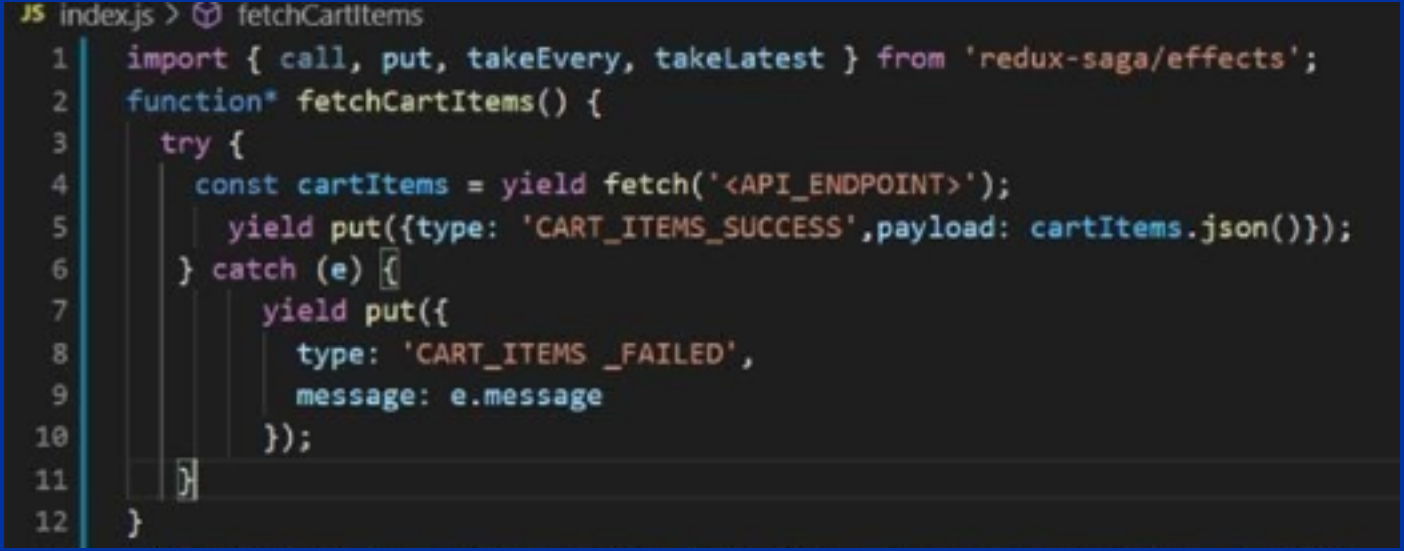
Redux Sagas

redux-sagas give lets you dispatch() actions in an asynchronous way using the **yield put, yield call, yield takeLatest** methods, INSTEAD of using thunks callback methods. Yield put replaces the dispatch()

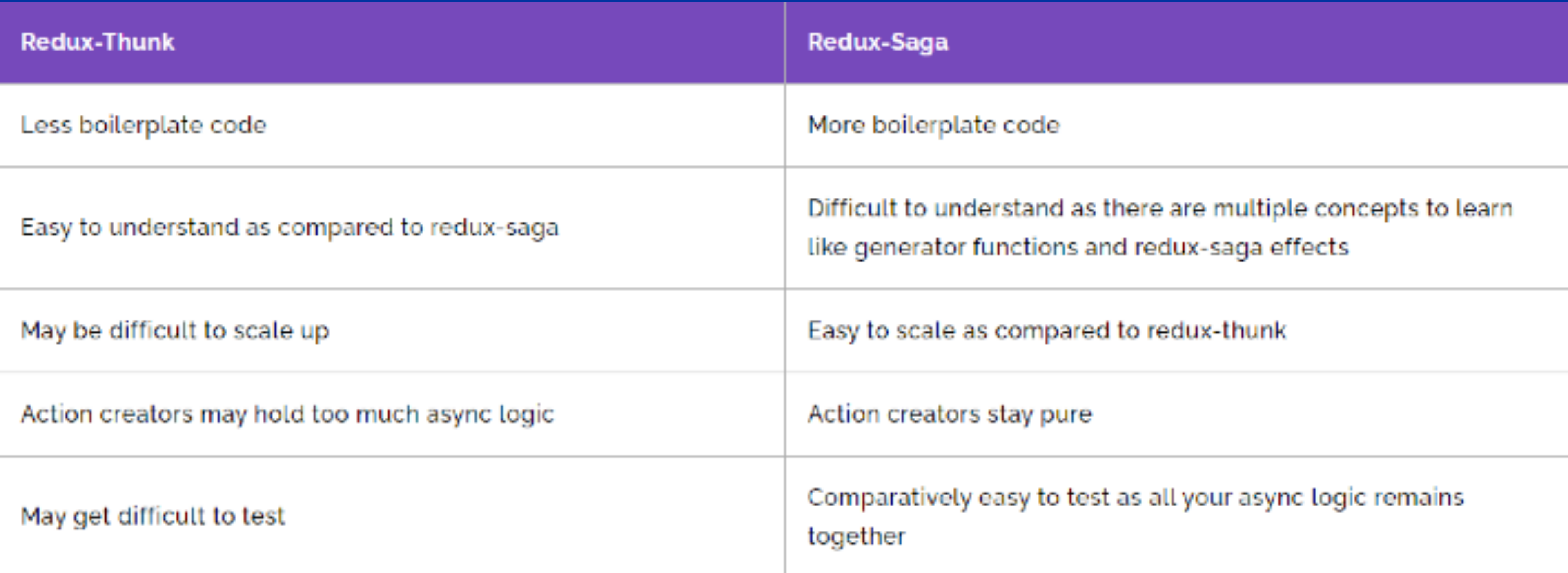


```
import { call, put, takeEvery, takeLatest } from 'redux-saga/effects';
function* fetchCartItems() {
  try {
    const cartItems = yield fetch('<API_ENDPOINT>');
      yield put({type: 'CART_ITEMS_SUCCESS',payload: cartItems.json()});
   } catch (e) {
        yield put({
          type: 'CART_ITEMS _FAILED',
          message: e.message
        });
   }
}
```

Redux Sagas vs Redux Thunks

It is difficult to explain all the benefits of both here and to explain all the tools of Redux-sagas without getting heavy in syntax.

In short for smaller apps use Redux-Thunks, for larger apps use Redux-Sagas



Redux-Thunk	Redux-Saga
Less boilerplate code	More boilerplate code
Easy to understand as compared to redux-saga	Difficult to understand as there are multiple concepts to learn like generator functions and redux-saga effects
May be difficult to scale up	Easy to scale as compared to redux-thunk
Action creators may hold too much async logic	Action creators stay pure
May get difficult to test	Comparatively easy to test as all your async logic remains together