**Here is the redux thunk concept with vanilla redux:**

**Index.js**

|  |
| --- |
| const { createStore, applyMiddleware } = require("redux");  const { delayActionMiddleware, fetchasyncMiddleware } = require("./middleware");  const { fetchTodos } = require("./function");  // initialState  const initialState = {    todos: [],  };  // reducer  const todoReducer = (state = initialState, action) => {    switch (action.type) {      case "todos/added": {        return {          ...state,          todos: [...state.todos, { title: action.payload }],        };      }      case "todos/todoLoadded": {        return {          ...state,          todos: [...state.todos, ...action.payload],        };      }      default:        break;    }  };  // store  const store = createStore(    todoReducer,    applyMiddleware(delayActionMiddleware, fetchasyncMiddleware)  );  // subscribe to state changes  store.subscribe(() => {    console.log("inside subscribe ");    console.log(store.getState());  });  // // dispatch action  // store.dispatch({ type: "todos/todoLoadded", payload: "learn redux with lws" });  // dispatch action to fetch todo data from API server, dispatch with a thunk function  store.dispatch(fetchTodos); |

**Middleware.js**

|  |
| --- |
| const fetch = require("node-fetch");  const delayActionMiddleware = (store) => (next) => (action) => {    if (action.type === "todos/todoLoadded") {      console.log("i am delaying you");      setTimeout(() => {        next(action);        return; // since this is asynchronous, so below return will work. so stop immediately      }, 2000);    }    return next(action);  };  // Node: node support fetch api to call api request but it is experimental. need to install npm i node-fetch@2 version 2  // fetch API requests for todos  const fetchasyncMiddleware = (store) => (next) => async (action) => {    if (typeof action === "function") {      return action(store);    }    return next(action);  };  module.exports = {    delayActionMiddleware,    fetchasyncMiddleware,  }; |

**Function.js**

|  |
| --- |
| const fetch = require("node-fetch");  const fetchTodos = async (store) => {    const response = await fetch(      "https://jsonplaceholder.typicode.com/todos?\_limit=5"    );    // convert response to json object, because we go stream in response    const todo = await response.json();    // after fetch todo data now dispatch todoLoaded    store.dispatch({      type: "todos/todoLoadded",      payload: todo,    });    console.log(`number of updated todos:${store.getState().todos.length}}`);  };  module.exports = {    fetchTodos,  }; |

**Flow:** dispatch(function) a function, then this dispatch checked into middleware function, if dispatch function is a function then call this action with dispatch, and getState() method. Inside this function, fetch data and then pass to the next middleware.

**We can do the similar think** with third party package. That is called **redux thunk.** Npm I redux-thunk.

Here we don’t need any middleware handler. Redux thunk do this job. We need to just call a dispatch(function) with function.

// store

const store = createStore(

  todoReducer,

  applyMiddleware(delayActionMiddleware, fetchasyncMiddleware)

);

No need here fetchAsyncMiddleware.

**Here is the github documentation:** <https://github.com/reduxjs/redux-thunk>

// store

const store = createStore(

  todoReducer,

  applyMiddleware(thunk)

);