了解NgRx StoreModule背后的魔力

<https://indepth.dev/posts/1199/understanding-the-magic-behind-ngrx-store>

我写这篇文章的时候正在浏览源代码，试图理解@ngrx/store模块的主要实体是如何绑定在一起的。结果，我发现了与模块的每个组件相关的一堆有趣的东西，这就是我将在本文中描述的内容。我们将详细检查每个实体，并解释其在整个体系结构中的作用。

在开始之前，让我们快速回顾一下主要的实体是什么:

State: 保存应用程序内部状态的数据结构

Store: 数据使用者和状态之间的桥梁

Actions: 触发状态变化的机制

Reducers: 一种通过写入状态来执行状态更改的机制

Meta-reducers: a mechanism to hook into the action -> reducer pipelineWe'll start with actions.

Actions

了解@ ngrx / effects背后的魔力

https://indepth.dev/posts/1206/understanding-the-magic-behind-ngrx-effects