What is a Redux Action?

A syntactical view of a Redux Action and its components: Type and Payload

When you walk into a bank, the Cashier receives your action i.e intent for coming into the bank. In our previous example, we called this **WITHDRAWAL_MONEY**. The only way money leaves the bank Vault, is if you make your action i.e intent known to the Cashier.

Now, the same goes for the Redux Reducer. Unlike **setState()** in pure React, the only way you update the state of a Redux application, is if you make your intent known to the REDUCER.

But how? By dispatching actions!

In the real world, you know the exact action you want to perform. You could probably write that down in a slip and hand it over to the Cashier.

This works almost the same way with Redux. The only challenge is, how do you describe an action in a Redux app? Definitely not by speaking over the counter or writing it down in a slip.

Well, there's good news.

An action is accurately described with a plain Javascript object. Nothing more.

There's just one thing to be aware of. An action MUST have a **type** field. This field describes the intent of the action.

In the bank story, if we were to describe your action to the bank, it'd look like this:

```
{
     type: "withdraw_money"
}
```

That's all, really.

That's all, really.

A Redux action is described as a plain object. Please have a look at the action above.

Do you think only the type field accurately describes your supposed action to make a withdrawal at a bank? Hmmm. I don’t think so. How about the amount of money you want to withdraw?

Many times your action will need some extra data for a complete description. Consider the action below. I argue that this makes for a better described action.

```
{
     type: "withdraw_money",
     amount: "$4000"
}
```

Now, there’s sufficient information describing the action. For example sake, ignore every other detail the action may include, such as your bank account number etc.

Other than the type field, the structure of your Redux Action is really up to you. However, a common approach is to have a type field and **payload** field as shown below:

```
{
     type: " ",
     payload: {}
}
```

The type field describes the action, and every other required data/information that describes the action is put in the payload object.

For example:

```
{
     type: "withdraw_money",
     payload: {
        amount: "$4000"
     }
}
```

So, yeah! That’s what an **action** is.

Who receives and interprets an action? The reducer. We’ll go into more detail in the next lesson.