Final Code

This lesson unifies the code presented in the last few lessons to give the Flux architecture the final shape

WE'LL COVER THE FOLLOWING

Full Implementation of Flux Architecture

In the last few lessons, we successfully hid the dispatcher while submitting actions. We may do it again for the store's registration:

```
var createSubscriber = function (store) {
  return dispatcher.register(store);
}
```

And instead of exporting the dispatcher we may export only these two functions `createAction` and `createSubscriber`. Here is how the final code looks like:

```
var Dispatcher = function () {
  return {
    _stores: [],
    register: function (store) {
      // expecting an `update` method for each store
      if (!store || !store.update) {
        throw new Error(
          'You should provide a store that has an `update` method'
        );
      } else {
        var consumers = [];
        var change = function () {
          consumers.forEach(function (consumer) {
            consumer(store);
          });
        };
        var subscribe = function (consumer, noInit) {
          consumers.push(consumer);
          !noInit ? consumer(store) : null;
        };
```

```
        this._stores.push({ store: store, change: change });
        return subscribe;

      }
      return false;
    },
    dispatch: function (action) {
      // check all stores for update
      if (this._stores.length > 0) {
        this._stores.forEach(function (entry) {
          entry.store.update(action, entry.change);
        });
      }
    }
  }
};

module.exports = {
  create: function () {
    var dispatcher = Dispatcher();

    return {
      createAction: function (type) {
        if (!type) {
          throw new Error('Please, provide action\'s type.');
        } else {
          return function (payload) {
            // actions are passed to dispatcher
            return dispatcher.dispatch({
              type: type,
              payload: payload
            });
          }
        }
      },
      createSubscriber: function (store) {
        return dispatcher.register(store);
      }
    }
  }
};
```

If we add the support of AMD, CommonJS and global usage we end up with 66 lines of code, 1.7KB plain or 795 bytes after minification JavaScript.

Let’s wrap up our discussion on Flux in the next lesson