autoupdate

This is the Meteor package that provides hot code push (HCP) functionality.

Every Meteor application that wasn't created with the `--minimal` option has this package already through `meteor-base` and HCP should work out of the box. For those running `--minimal` applications and want to benefit from this package, just add it with `meteor add autoupdate`.

> `autoupdate` adds up to 30KB on your client's production bundle.

With this package Meteor will use DDP to publish a collection called *'meteor_autoupdate_clientVersions'*. This collection will be subscribed by the user's client and every time the client identifies a change in the published version it will refresh itself.

Browser Client

The refresh will happen in the browser in two different ways: a *soft update*, and a *hard update*. If Meteor identifies that only stylesheets were changed, then it will verify if the user's browser is capable of reloading CSS on the fly, and a soft update will take place. The soft update will replace the old stylesheet with the new stylesheet without triggering a full page reload.

In cases where a change in a server's or client's compiled file happens, the hard update will take place: Meteor will force a complete browser reload using the `reload` package.

> Among other things, the `reload` package tries do reload the application preserving some unchanged cached files.

Cordova Client

There is no soft update with Cordova apps, the client is always fully refreshed once a change is detected.

`usesCleartextTraffic`

Starting with Android 9 (API level 28), cleartext support is disabled by default. During development `autoupdate` uses cleartext to publish new client versions. If your app targets Android 9 or greater, it will be necessary to create

a `mobile-config.js` file enabling the use of cleartext in order to have HCP working:

```
App.appendToConfig(`<edit-config file="app/src/main/AndroidManifest.xml"
                      mode="merge"
                      target="/manifest/application"
                      xmlns:android="http://schemas.android.com/apk/res/android">
        <application android:usesCleartextTraffic="true"></application>
    </edit-config>
`);
```

`--mobile-server`

Additionally, for the HCP functionality to work it is also mandatory to provide the address for the application server with `--mobile-server` option. If you're testing your app on an emulator you should run it with `meteor run android --mobile-server 10.0.2.2:3000`. If you're running it on a real device, the application server and the device should be on the same network, and you should run your app with `meteor run android --mobile-server XXX.XXX.XXX.XXX` where *XXX.XXX.XXX.XXX* is your local development address, *e.g. 192.168.1.4*.

> To have a better understanding of how HCP works for mobile apps already published to production refer to Hot code push on mobile