

TODO MVC

脚本程序设计大项目

1452776 罗雪

2017 年 6 月 23 日

目录

目录	2
1. 项目介绍	3
1.1 基本介绍	3
1.2 功能介绍指南	3
1.2.1 基本功能介绍	3
1.2.2 高级功能介绍	6
1.3 交互与页面设计	8
2. 开发过程	9
2.1 文件目录及说明	9
2.2 重要函数接口及相关说明	10
2.2.1 event	10
2.2.2 view 层	10
2.2.3 controller 层	11
2.2.4 Model 层	11
3. 亮点汇总	12

1. 项目介绍

1.1 基本介绍

这是个主要用于移动端的 todo list，同时 web 端也可以操作, web 端建议使用 chrome.

项目 demo 地址（建议使用手机 chrome 浏览器打开）：

<https://melodyluoxue.github.io/#>

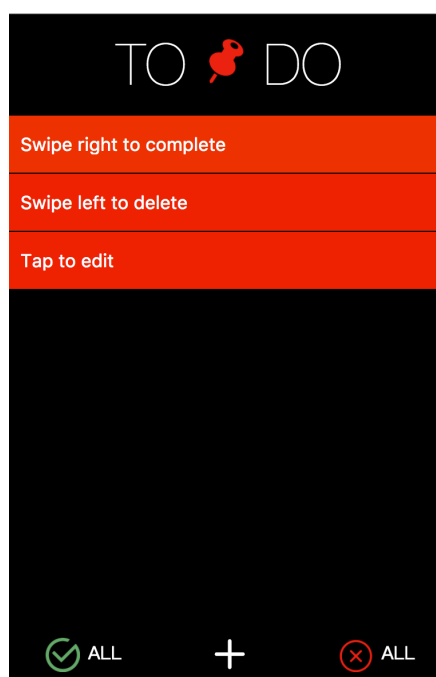
项目代码 github 地址：

<https://github.com/melodyluoxue/melodyluoxue.github.io>

1.2 功能介绍指南

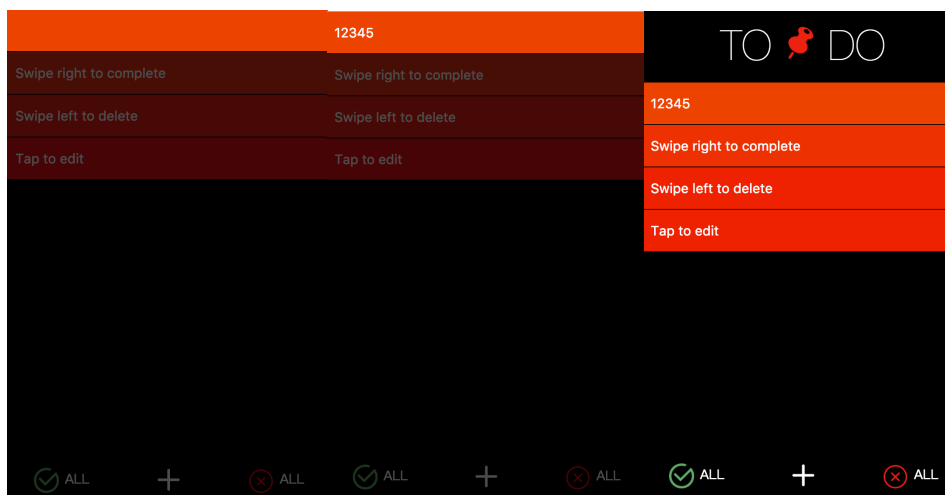
1.2.1 基本功能介绍

初始界面：默认有三条 list，用于提示用户进行相应的交互操作。



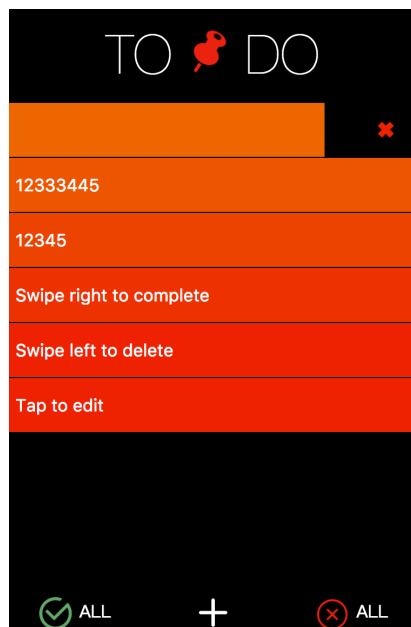
i. 新建事项（移动端／web 端）

- 点击底部中间的加号，弹出一条新的高亮 listitem，其余部分变暗。
- 点击此高亮 listitem,出现输入光标，键入内容。
- 按回车生成该条新建的 listitem。



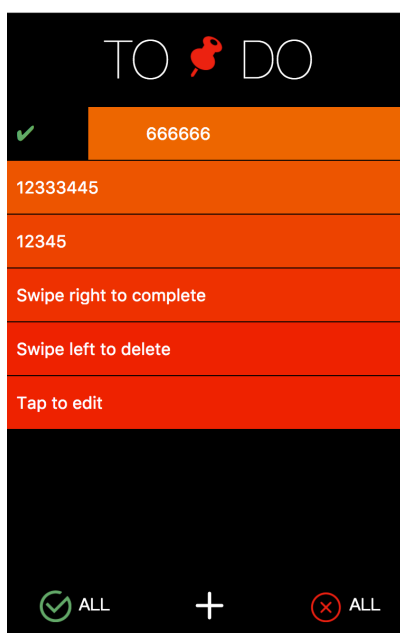
ii. 删除事项（移动端／web 端）

- 用手 / 鼠标在 listitem 右端向左拖动即可删除。



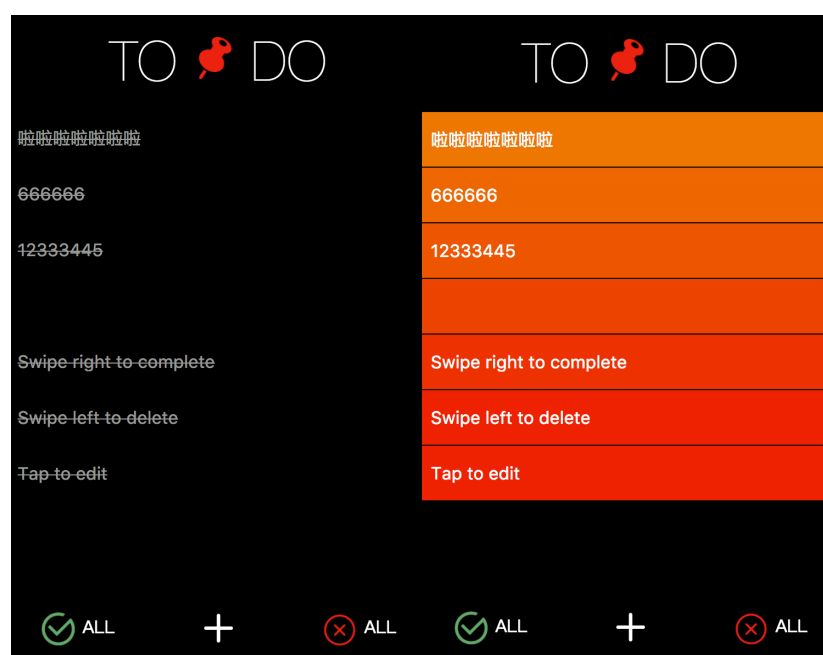
iii. 完成事项（移动端／web 端）

- 用手 / 鼠标在 listitem 左端向右拖动即可删除。



iv. 全部完成／取消（移动端／web 端）

- 点击左下方绿色勾---全部完成 / 再次点击---取消。



v. 删除已完成（移动端／web 端）

- 点击右下角红色叉，即可一键将如图中下方三条已完成内容删除。



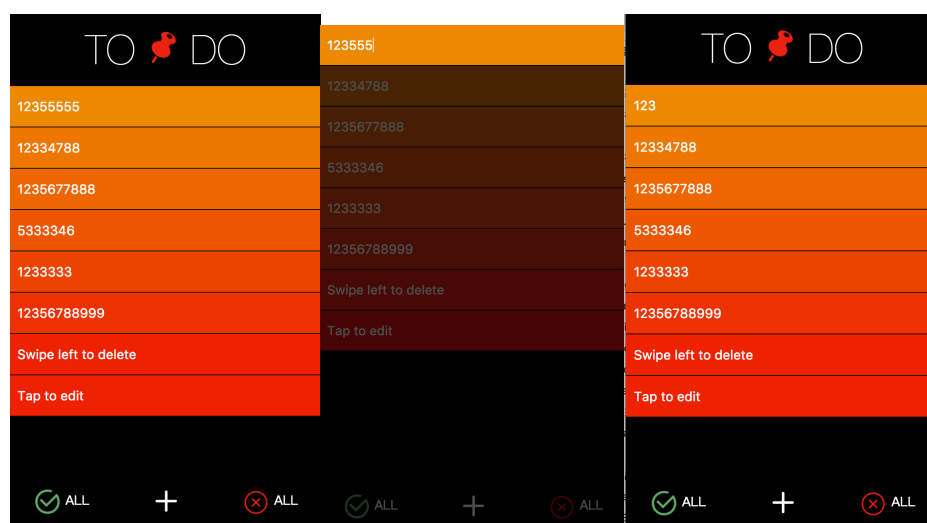
vi. 保持页面状态（移动端/web 端）

- 点击刷新页面仍然保持。

1.2.2 高级功能介绍

i. 单条编辑（移动端/web 端）

- 点击某条 listitem,触发编辑状态。
- 编辑该条内容（如下点击第一条修改成“123”）后回车。

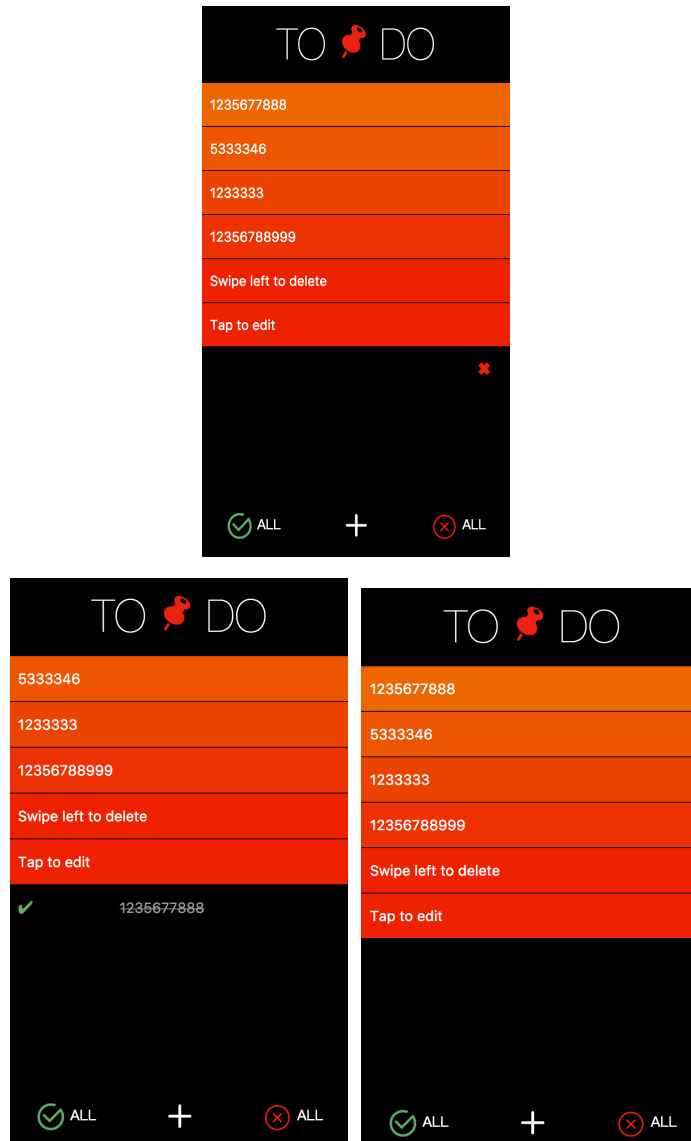


ii. 过滤

每条 listitem 都是个状态机，很自然的所有未完成的 item 在上方，已完成的在下方，避免所有 item 混杂在一起。不需要单独设定按钮来展示已完成 / 未完成。

iii. 单条已完成 item 的操作（移动端/web 端）

- 从最右端向左滑，删除该条 item 。
- 向右滑，恢复未完成状态。

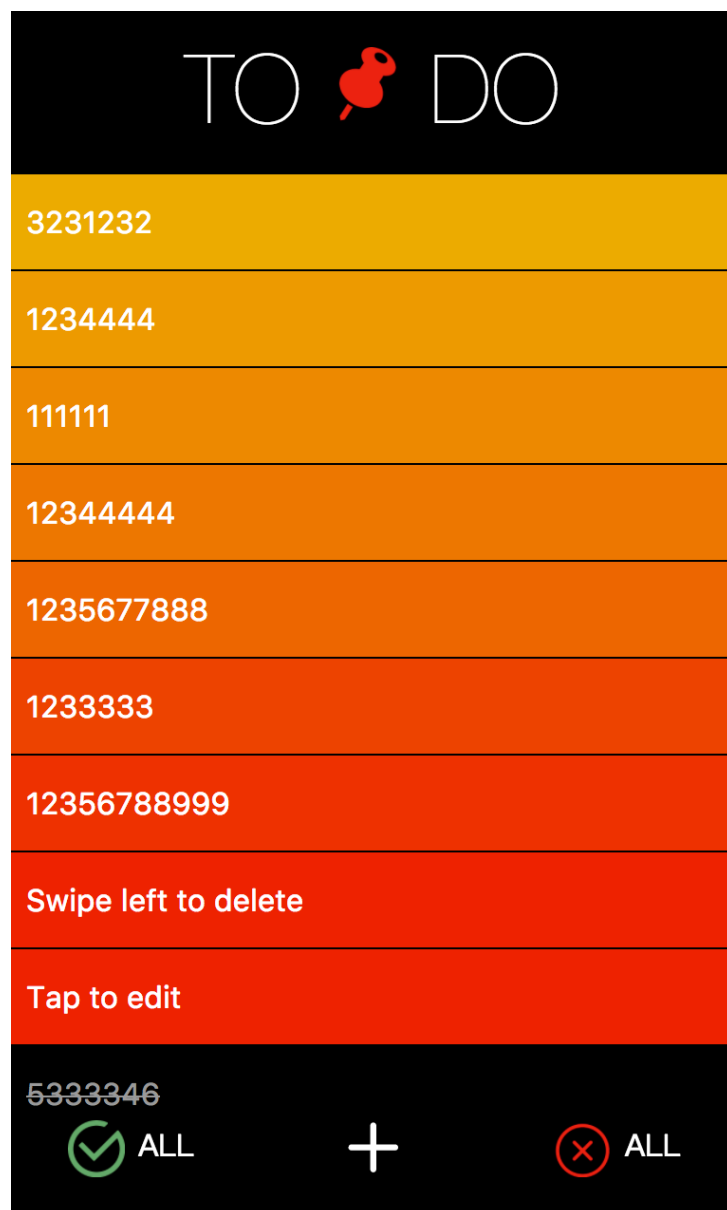


1.3 交互与页面设计

1) 最流行的移动端 UI 设计风格——极简主义

以黑为底，以红橙黄有梯度的渐变色的配色方案，大胆活泼，并使得它本身易于导航，并且用渐变色来设计 listitem 使之具备强大的视觉表现力。

同时扁平化的设计风格也使它看起来并不因此而显得杂乱。



2) 移动端注重手势的交互

尽可能地摆脱按钮的束缚，对于单条 item 仅仅左滑右滑点击可完成基本操作，践行少即是多。

3) 鲜明的层次感

用亮暗关系突出正在编辑的重点，使得界面富有层次，并且符合用户习惯，这一点在移动端显得尤为重要。

2. 开发过程

2.1 文件目录及说明

<i>File Type</i>	<i>File Name</i>	<i>Function</i>
<i>Style</i>	style.styl	css 文件预处理器，主要做颜色渐变的部分比较方便。
	style.css	自动生成的 css 文件。
<i>Model</i>	store.js	存储数据以及进行各种增删改。
<i>Controller</i>	app.js	将 model 层和 view 层绑定在一起，做状态机的处理。
<i>View</i>	ListItem.js	做状态机及相应界面的显示，对移动端触发事件的响应。
<i>Others</i>	utils.js	把一些小组件小工具都放这里，如 createDOM, mount 函数
	Event.js	把 cotroller 与 view 连在一起，把 view 发生的变化传给 controller。

hammer.min.js	移动端的库
---------------	-------

2.2 重要函数接口及相关说明

2.2.1 EVENT

文件名: Event.js

Event 类型:

- 'ADD': 增加一条数据字段事件。
- 'DELETE': 删除一条数据字段事件。
- 'EDIT': 修改一条数据字段内容事件(包括设置状态为已完成以及编辑修改 item 内容)。

Name	Function
<i>fireEvent</i>	为一个 item 生成一个事件。
<i>addEvent</i>	绑定一个事件到对应的 model 数据操作函数。

2.2.2 VIEW 层

文件名: ListItem.js

ListItem.state 状态机:

- completed: 该事项完成与否
- alive: 该事项被删除与否

ListItem.msg: 事项具体内容

ListItem.key: 根据它来唯一对应某一条 listitem

<i>Name</i>	<i>Function</i>
<i>render</i>	进行页面渲染，包括 web 端和移动端。
<i>setState</i>	更新状态，每更新一次状态就重新渲染一次。
<i>pan</i> 与 <i>input</i> 相关函数	根据页面具体操作进行相应的状态更新和生成事件。

2.2.3 CONTROLLER 层

文件名：app.js

其中调用的 mount:

将 view 层和 controller 层绑定，状态改变重新渲染。

<i>Name</i>	<i>Function</i>
<i>window.onload</i>	页面和数据 store 初始化。默认有三条 item；如果本身有数据则显示数据，若无则显示默认的一条 item。
事件监听系列函数	对监听的操作通过进行对应状态的改变生成事件及 view 的响应。
绑定事件系列函数	绑定一个事件到对应的 model 数据操作函数。

2.2.4 MODEL 层

文件名：store.js

对数据字段增删改。

3. 亮点汇总

- ✧ **移动端的实现**: hammerJS 库, 优秀、轻量级的触屏设备手势库。
- ✧ **交互界面设计**: 极简主义、渐变梯度颜色、高亮明暗鲜明对比、用预编译 css 工具实现。
- ✧ **状态机**: 将每一个 listitem 设计为一个状态机, 易于管理, 操作方便, 避免将代码暴露在外层。状态改变重新渲染。Mvc 之间通过 event 进行通信, 通过 view 通过 event 告诉 controller 指示 model 层的操作。
- ✧ **组件化**: 比起课上教的方式多了视图层, 彻底分成 MVC 三层, 防止所有代码都耦合在一起。Listitem 已实现组件化。
- ✧ **每一个 listitem 拥有一个唯一的 key 值**: 使得 msg 一样的事项也能被一一对应, 方便删改操作。