Web Programming-Final Project

To-Do List

1552746 崔鹤洁

June 23th, 2018

1. 功能

1.手机版ToDo基本功能：

(a) 包含新增、删除、展现列表、全部完成/取消、删除已完成；

(b) 保存页面状态，刷新页面后可恢复；

(c) 使用ajax进行HTTP网络请求；

2.高级功能（亮点）：

(a) 增加隐藏的侧边栏，并将过滤器隐藏到侧边栏；

(b) 手写手势控制，未使用Hammer.js：可在手机上double-tab，编辑单条todo；

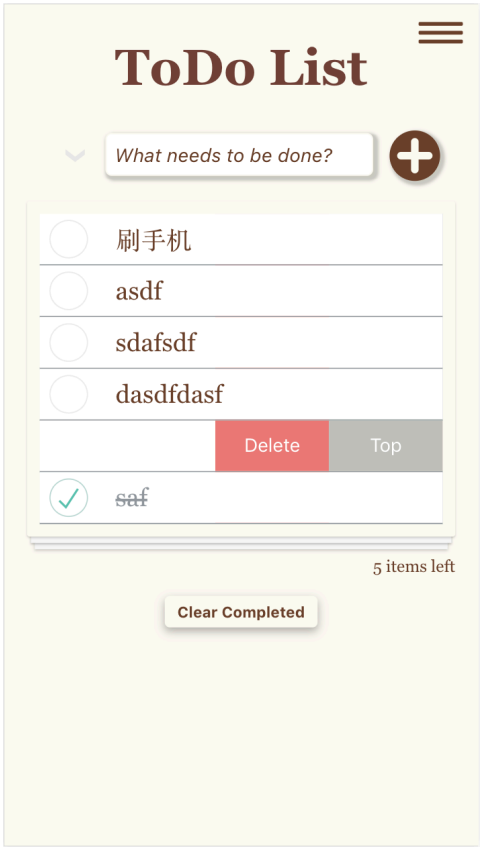
(c) 手写手势控制：类似QQ左滑单条todo，出现删除按钮。

(d) 重要单条todo置顶，左滑单条to-do可以看到置顶按钮。

(e) 改掉了clear-completed的bug

2. 亮点代码展示

3.1 自己实现左滑手势，出现删除和置顶按钮



滑动手势控制的实现

定义记录位置变化的变量：

```
//获取要绑定监听的dom元素
var front = item.querySelector('.swipe-front');
//触摸事件发生的位置
var client = 0;
//相对上一次的位置变化
var deltaX = 0;
//相对于原点的位置（原点指固定在右侧不滑动的时候）
var transX = 0;
```

touchstart函数：获取触摸事件发生的位置

```
front.addEventListener('touchstart', function (ev) {
    console.log(ev);
    client = ev.targetTouches[0].clientX;
}, true);
```

touchmove函数：计算出位置的变化量，更新触摸事件发生的位置。计算transX为发生位置改变以后的新位置，这里需要注意限制一下向左向右移动的最大范围（端点）。最后更新一下存在dom元素中的位置属性。

```
front.addEventListener('touchmove', function (ev) {
    //计算位置变化量
    deltaX = ev.targetTouches[0].clientX - client;
    //更新client
    client = ev.targetTouches[0].clientX;

    //利用正则表达式匹配出上一次的transX
    let reEx = /-?\d+/g;
    //input:translate(-180px)
    //prePer:["-180", index: 11, input: "translateX(-180px)", groups: undefined]
    let prePer = reEx.exec(front.style.transform);
    if (prePer !== null) {
        prePer = parseInt(prePer[0]);
        //transX为新的相对原点位置
        transX = deltaX + prePer;
    }
    else {
        transX = deltaX;
    }
    //限制一下端点
    if (transX <= (-window.shiftWidth))
        transX = (-window.shiftWidth);
    if (transX >= 0)
        transX = 0;
    console.log(deltaX);
    //实时更新位置属性
    front.style.transform = 'translateX( ' + transX + 'px )';
}, true);
```

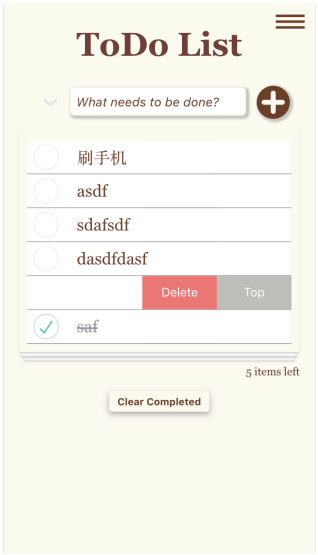
touchend函数：这里实现了对滑动的自动控制，如果计算发现用户只滑了一半没有滑到头，那么调用finishLeft或者finishRight函数，将front部分自动滑到端头。

```
front.addEventListener('touchend', function (ev) {
    //用来控制如果用户没有滑到头的话，就自动化滑到头
    if (deltaX < 0 && transX > (-window.shiftWidth)) {
        finishLeft(front, transX)
    }
    else if (deltaX > 0 && transX < 0) {
        finishRight(front, transX)
    }
    clientX = 0;
    deltaX = 0;
    transX = 0;
}, true);
```

控制自动滑动的函数中调用了requestAnimationFrame的回调函数，重复执行向左或者向右每一帧移动8个pixel的动作。

```
//利用回调来刷新，用于后面写的滑动手势动画
window.requestAnimFrame = (function () {
    //requestAnimationFrame可以传一个回调函数进去，隔一帧之后调用
    return window.requestAnimationFrame ||
        window.webkitRequestAnimationFrame ||
        window.mozRequestAnimationFrame ||
        function (callback) {
            //隔一帧之后调用callback
            window.setTimeout(callback, 1000 / 60);
        };
})();
//左滑单条to-do
function finishLeft(ob, nowTran) {
    //每一帧移动8个pixel
    let tran = nowTran - 8;
    //限制端点
    if(tran<(-window.shiftWidth))
        tran = (-window.shiftWidth);
    ob.style.transform = 'translateX( ' + tran + 'px )';
    if(nowTran!==tran)
    //重复执行，直到滑到头了
        window.requestAnimationFrame(function() {
            finishLeft(ob, tran)
        })
}

//右滑单条to-do
function finishRight(ob, nowTran) {
    let tran = nowTran + 8;
    if(tran>0)
        tran = 0;
    ob.style.transform = 'translateX( ' + tran + 'px )';
    if(nowTran!==tran)
    //重复执行，直到滑到头了
        window.requestAnimationFrame(function() {
            finishRight(ob, tran)
        })
}
```

删除和置顶单条to-do：



删除和置顶的逻辑很简单

```
//删除一条todo
item.querySelector('.destroy').addEventListener('click', function() {
    data.items.splice(index, 1);
    update();
}, false);

//置顶一条todo
item.querySelector('.top').addEventListener('click', function () {
    let tmp = data.items.splice(index, 1);
    data.items.push(tmp[0]);
    update();
},false);
```

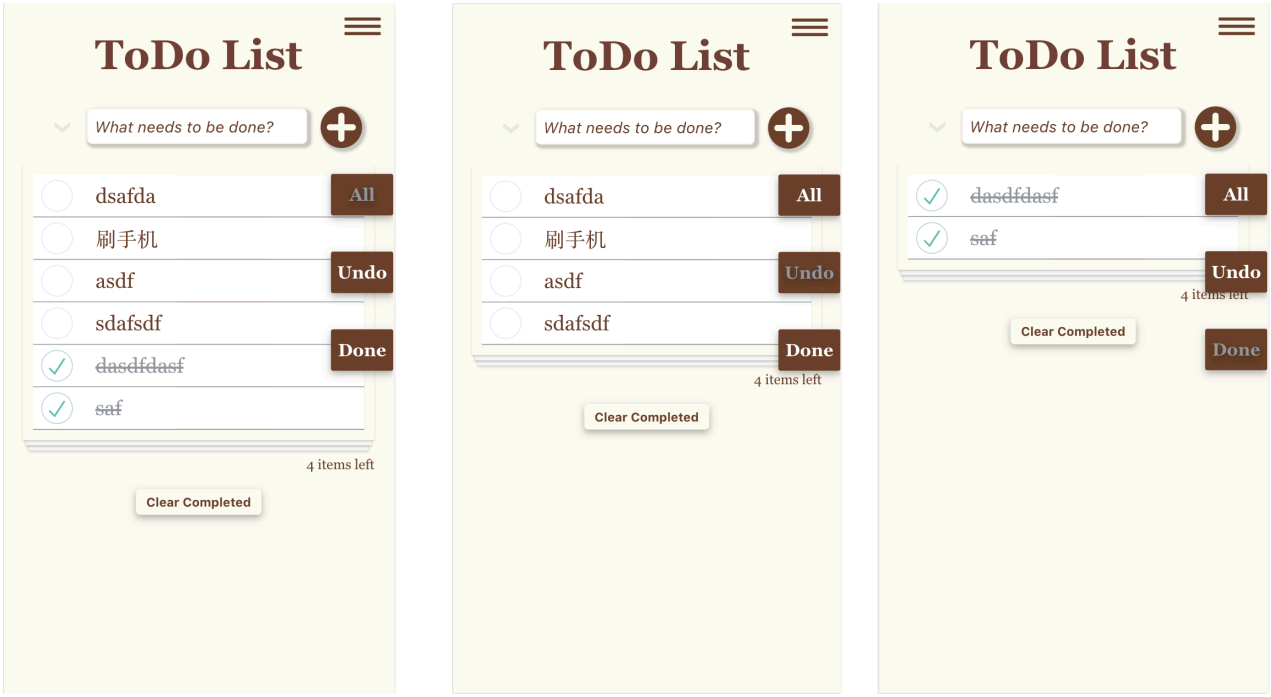
*3.2double click*手势，编辑单条*to-do*

首先要判断是否两次连续的点击是否可以归为一个double-click事件（时间间隔小于某一个特定的阈值），如果是一次dbclick，则将label移除，新建一个输入框加进去。在对输入框的编辑过程中，如果光标blur则不会存储到后端；编辑完成后，将前端数据存储到服务器。

```
//double tap编辑单条to-do
var label = item.querySelector('.todo-label');
//记录此次点击上一次点击的发生时间
var latesttap = new Date().getTime();
```

```
label.addEventListener('touchstart', function() {
    //记录当前时间
    var now = new Date().getTime();
    //检查两次点击时间间隔
    var timesince = now - latesttap;
    latesttap = now;
    //如果时间间隔过大，不判断为dbclick
    if(timesince >= 600 || timesince <= 0)
        return;
    //否则就加入editing类
    label.classList.add(CL_EDITING);
    let del = item.querySelector('.swipe-front');
    //新建一个输入框
    var edit = document.createElement('input');
    var finished = false;
    edit.setAttribute('type', 'text');
    edit.setAttribute('class', 'edit');
    edit.setAttribute('value', label.innerHTML);

    //对输入框停止编辑时（如光标移开）
    function finish() {
        if (finished) return;
        finished = true;
        del.removeChild(edit);
        del.appendChild(label);
        label.classList.remove(CL_EDITING);
    }

    //鼠标移开，编辑记录不会保存
    edit.addEventListener('blur', function() {
        finish();
    }, false);

    //编辑完成
    edit.addEventListener('keyup', function(ev) {
        if (ev.keyCode === 27) { // Esc
            finish();
        }
        else if (ev.keyCode === 13) {// Enter
            label.innerHTML = this.value;
            itemData.msg = this.value;
            //前端的数据更新到服务器
            update();
        }
    }, false);

    //把input框加进去，label移除
    del.appendChild(edit);
    del.removeChild(label);
    edit.focus();

}, false);
```

3.3 隐藏到侧边的过滤器（点击右角的菜单栏出现）



因为考虑到手势操控的话，三个过滤小按钮放在下方可能不是那么好操作，因此将其隐藏到sidebar里面，通过点击右上角的菜单按钮出现，这里主要修改的是css文件的sidebar类。

```
//展示侧边栏
function toggleSidebar(){
    document.getElementById("sidebar").classList.toggle('active');
}
```

3.4 改掉了*clear-completed*的*bug*

之前老师给的代码仓库里面的clear-completed在勾选多个删除的时候会有bug，因为它使用的是foreach，从前向后删除的话序号变化了有的元素就删除不到了，修改掉这个bug只需要将逻辑改成从后向前删除。

```
//这里之前有个小bug，把foreach改掉（从后向前删除）就好了
var clearCompleted = $('.clear-completed');
clearCompleted.addEventListener('click', function() {
    for(let i = data.items.length-1;i >= 0;i--) {
        if (data.items[i].completed) data.items.splice(i, 1);
    }
    update();
}, false);
```

3. 文件结构

3.1 目录

1. index.html
2. todo.css
3. ajax.js
4. provider-ajax.js
5. model.js
6. global.js：一些全局变量
7. server.js：服务端主要逻辑
8. util.js：一些全局的函数

3.2 Reference

* ajax和ajax-provider参考了徐老师的代码仓库，无其他来源的参考代码。

1552746 崔鹤洁
June 23th, 2018