2019.6.19

Likes.js

PersisData 存储数据

localStorage.setItem('likes',JSON.stringify(this.Like));

readStorage 读取数据

const storage = JSON.parse(localStorage.getItem('likes'));

if (storage) this.like = storage;

Index.js

Window.addEventListener(‘load’, () => { //app加载时自动读取

State.likes = new Likes(); //自动生成state.like

State.likes.readStorage();

likesView.toggleLikeMenu(state.Likes.getNumLikes());

State.likes.like.forEach(like =>

likeView.renderLike();

}

2019.6.17

Fokify

Like.js

constructor(){this.like =[]}

addlike(id, title,author,img){const like ={id, title, author,img}; this.like.push(like) return like}

deleteLike(id){findIndex(el=>el.id ===id) this.like.splice()}

isLiked(id){return findindex !== -1}

numLikes() {return like.length}

Like Controller

单击Btn => 先判断有没有state.likes if(!state.likes) new

获得该recipe id ==> currentId = state.recipe.id

判断likes里有没有该recipe ,若没有则加入 if (!state.likes.isLiked(id)) {

1 state.likes. addlike 2 likeView改变收藏心图片 3 likeView List添加该recipe

LikeView

toggleLikeBtn (isliked) ? iconstring=’icon-heart’:‘-outlined’ =>dom.setAttribute(‘href’, `img/icons.svg#${iconstring}`

renderLike inserAdjacentHtml ‘beforeend’, markup

deleteLike querySelector(`.likes\_\_link[href\*=”${id}`)

Dom操作：

修改href属性：

<button class="recipe\_\_love">

<svg class="header\_\_likes">

<use href="img/icons.svg#icon-heart-outlined"></use>

</svg>

</button>

document.querySelector('.recipe\_\_love use').setAttribute('href',`img/icons.svg#${iconString}`);

根据特定href属性选择dom

<li>

<a class="results\_\_link" href="#${recipe.recipe\_id}">

<figure class="results\_\_fig">

<img src="${recipe.image\_url}" alt="Test">

</figure>

<div class="results\_\_data">

<h4 class="results\_\_name">${limitRecipeTitle(recipe.title)}</h4>

<p class="results\_\_author">${recipe.publisher}/p>

</div>

</a>

</li>

document.querySelector(`.results\_\_link[href="#${id}"]`).classList.add('results\_\_link--active');

document.querySelector(`a[href="#${id}"]`).classList.add('results\_\_link--active');