你是一名油猴脚本编写大师，以下是一段油猴代码，请据此根据我的要求进行修改，并给我一个符合要求的脚本

//快捷呼出客服&解决复制id的BUG

(function() {

'use strict';

// 定义要查找的元素选择器

const selector = "#modal-root > div > div > div:nth-child(5)";

// 用于记录已经复制过的元素，避免重复复制

const clonedElements = new Set();

// 用于存储模态框元素

let modalRoot;

// 定义一个函数来查找元素并执行操作

function findAndDuplicateElement() {

// 使用 document.querySelector 查找目标元素

const targetElement = document.querySelector(selector);

if (targetElement &&!clonedElements.has(targetElement)) {

// 复制目标元素

const clonedElement = targetElement.cloneNode(true);

// 清空复制元素原内容

clonedElement.textContent = '';

// 设置文本颜色为白色

clonedElement.style.color = "white";

// 使用 margin 调整文本位置，这里以左右各偏移 10px 为例

clonedElement.style.marginLeft = "0px";

clonedElement.style.marginRight = "10px";

// 设置宽度为89%

clonedElement.style.width = "89%";

// 为复制元素添加一个新的类名

clonedElement.classList.add('custom-cloned-element');

// 创建左边部分

const leftPart = document.createElement('span');

leftPart.textContent = "国服客服";

leftPart.style.display = "inline-block";

leftPart.style.width = "50%";

leftPart.style.textAlign = "center";

leftPart.style.color = "white";

leftPart.style.cursor = "pointer";

// 创建右边部分

const rightPart = document.createElement('span');

rightPart.textContent = "49客服";

rightPart.style.display = "inline-block";

rightPart.style.width = "50%";

rightPart.style.textAlign = "center";

rightPart.style.color = "white";

rightPart.style.cursor = "pointer";

// 为左右部分添加点击事件监听器

leftPart.addEventListener('click', function() {

const textElement = document.querySelector("#modal-root > div > div > div.ContextMenuStyle-menuItem.ContextMenuStyle-menuItemRank > div > div > div > span");

if (textElement) {

let textToCopy = textElement.textContent;

textToCopy = textToCopy.replace(/\[.\*?\]|\s/g, '');

const textArea = document.createElement('textarea');

textArea.value = textToCopy;

document.body.appendChild(textArea);

textArea.select();

document.execCommand('copy');

document.body.removeChild(textArea);

}

window.open('https://admin.qidian.qq.com/template/blue/mp/menu/qr-code-jump-market.html?linkType=0&env=ol&kfuin=3009072421&fid=327&key=f3321e05f6258773ceb56a6411e30ff7&cate=1&source=&isLBS=&isCustomEntry=&type=16', '\_blank');

});

rightPart.addEventListener('click', function() {

const textElement = document.querySelector("#modal-root > div > div > div.ContextMenuStyle-menuItem.ContextMenuStyle-menuItemRank > div > div > div > span");

if (textElement) {

let textToCopy = textElement.textContent;

textToCopy = textToCopy.replace(/\[.\*?\]|\s/g, '');

const textArea = document.createElement('textarea');

textArea.value = textToCopy;

document.body.appendChild(textArea);

textArea.select();

document.execCommand('copy');

document.body.removeChild(textArea);

}

window.open('https://u.4399.com/kf/im/3Dtk', '\_blank');

});

// 将左右部分添加到复制元素中

clonedElement.appendChild(leftPart);

clonedElement.appendChild(rightPart);

// 将复制的元素插入到目标元素之后

targetElement.parentNode.insertBefore(clonedElement, targetElement.nextSibling);

// 记录已经复制过的元素

clonedElements.add(targetElement);

// 获取模态框根元素

modalRoot = document.getElementById('modal-root');

if (modalRoot) {

setupMutationObserver();

}

// 为新按钮添加点击事件监听器

const newButton = document.querySelector("#modal-root > div > div > div:nth-child(5) > span");

if (newButton) {

newButton.addEventListener('click', function() {

const textElement = document.querySelector("#modal-root > div > div > div.ContextMenuStyle-menuItem.ContextMenuStyle-menuItemRank > div > div > div > span");

if (textElement) {

let textToCopy = textElement.textContent;

textToCopy = textToCopy.replace(/\[.\*?\]|\s/g, '');

const textArea = document.createElement('textarea');

textArea.value = textToCopy;

document.body.appendChild(textArea);

textArea.select();

document.execCommand('copy');

document.body.removeChild(textArea);

}

});

}

}

}

// 设置 MutationObserver 来监控模态框的变化

function setupMutationObserver() {

const observer = new MutationObserver((mutationsList) => {

for (const mutation of mutationsList) {

if (mutation.type === 'attributes' && mutation.attributeName === 'style') {

const displayValue = window.getComputedStyle(modalRoot).display;

if (displayValue === 'none') {

// 恢复模态框显示

modalRoot.style.display = 'block';

}

}

}

});

const config = { attributes: true, attributeFilter: ['style'] };

observer.observe(modalRoot, config);

}

// 向页面中注入 CSS 样式

const style = document.createElement('style');

style.textContent = `

.custom-cloned-element {

background-color: transparent;

background-image: none;

box-sizing: border-box; /\* 确保内边距和边框包含在元素的宽度和高度内 \*/

}

.custom-cloned-element span {

background-color: transparent;

background-image: none;

box-sizing: border-box;

}

.custom-cloned-element span:hover {

background-color: rgba(128, 128, 128, 0.3); /\* 鼠标悬停时的背景颜色，可根据需求调整 \*/

border-radius: inherit; /\* 继承原元素的边框圆角 \*/

overflow: hidden; /\* 防止背景溢出 \*/

}

`;

document.head.appendChild(style);

// 每隔 500 毫秒调用一次 findAndDuplicateElement 函数

const intervalId = setInterval(findAndDuplicateElement, 1);

})();

//复制团标

(function() {

'use strict';

function showFloatingNotification(message, isSuccess = true) {

const notification = document.createElement('div');

notification.textContent = message;

notification.style.position = 'fixed';

notification.style.top = '20px';

notification.style.left = '50%';

notification.style.transform = 'translateX(-50%)';

notification.style.padding = '10px 20px';

notification.style.backgroundColor = isSuccess ? 'rgba(0, 128, 0, 0.9)' : 'rgba(255, 0, 0, 0.9)';

notification.style.color = '#fff';

notification.style.borderRadius = '5px';

notification.style.zIndex = '9999';

notification.style.fontSize = '14px';

notification.style.fontFamily = 'Arial, sans-serif';

notification.style.boxShadow = '0 2px 5px rgba(0, 0, 0, 0.3)';

notification.style.opacity = '0';

notification.style.transition = 'opacity 0.3s ease-in-out';

document.body.appendChild(notification);

setTimeout(() => {

notification.style.opacity = '1';

}, 10);

setTimeout(() => {

notification.style.opacity = '0';

setTimeout(() => {

notification.remove();

}, 300);

}, 3000);

}

function addCopyClanTagOption(contextMenu, copyNameOption, username) {

if (!contextMenu.querySelector('.copy-clan-tag') && username && username.includes('[') && username.includes(']')) {

const copyClanTagOption = document.createElement('div');

copyClanTagOption.className = 'ksc-67 Common-flexStartAlignCenter copy-clan-tag';

const spanElement = document.createElement('span');

spanElement.textContent = '复制军团标';

copyClanTagOption.appendChild(spanElement);

if (copyNameOption) {

const computedStyle = window.getComputedStyle(copyNameOption);

const spanComputedStyle = window.getComputedStyle(copyNameOption.querySelector('span'));

copyClanTagOption.style.padding = computedStyle.padding || '5px 10px';

copyClanTagOption.style.margin = computedStyle.margin || '0px';

copyClanTagOption.style.color = computedStyle.color || '#fff';

copyClanTagOption.style.display = computedStyle.display || 'flex';

copyClanTagOption.style.justifyContent = computedStyle.justifyContent || 'flex-start';

copyClanTagOption.style.alignItems = computedStyle.alignItems || 'center';

copyClanTagOption.style.width = '100%';

copyClanTagOption.style.boxSizing = 'border-box';

copyClanTagOption.style.height = computedStyle.height || 'auto';

copyClanTagOption.style.lineHeight = computedStyle.lineHeight || 'normal';

copyClanTagOption.style.cursor = 'pointer';

copyClanTagOption.style.visibility = 'visible';

spanElement.style.padding = spanComputedStyle.padding || '0px';

spanElement.style.margin = spanComputedStyle.margin || '0px';

spanElement.style.width = spanComputedStyle.width || 'auto';

spanElement.style.whiteSpace = spanComputedStyle.whiteSpace || 'nowrap';

spanElement.style.lineHeight = spanComputedStyle.lineHeight || 'normal';

spanElement.style.visibility = 'visible';

copyNameOption.insertAdjacentElement('afterend', copyClanTagOption);

} else {

copyClanTagOption.style.display = 'flex';

copyClanTagOption.style.justifyContent = 'flex-start';

copyClanTagOption.style.alignItems = 'center';

copyClanTagOption.style.padding = '5px 10px';

copyClanTagOption.style.margin = '0px';

copyClanTagOption.style.color = '#fff';

copyClanTagOption.style.width = '100%';

copyClanTagOption.style.boxSizing = 'border-box';

copyClanTagOption.style.height = 'auto';

copyClanTagOption.style.lineHeight = 'normal';

copyClanTagOption.style.cursor = 'pointer';

copyClanTagOption.style.visibility = 'visible';

spanElement.style.padding = '0px';

spanElement.style.margin = '0px';

spanElement.style.width = 'auto';

spanElement.style.whiteSpace = 'nowrap';

spanElement.style.lineHeight = 'normal';

spanElement.style.visibility = 'visible';

contextMenu.appendChild(copyClanTagOption);

}

copyClanTagOption.addEventListener('click', () => {

console.log(`点击“复制军团标”，当前用户名: ${username}`);

if (!username) {

console.log('用户名为空，无法提取军团标');

showFloatingNotification('未找到用户名', false);

return;

}

let clanTag = '';

const clanTagMatch = username.match(/^\[([^\]]+)\]/);

if (clanTagMatch && clanTagMatch[1]) {

clanTag = `[${clanTagMatch[1]}]`;

} else if (username.includes('[') && username.includes(']')) {

const startIndex = username.indexOf('[');

const endIndex = username.indexOf(']');

if (startIndex !== -1 && endIndex !== -1 && endIndex > startIndex) {

clanTag = username.substring(startIndex, endIndex + 1);

}

}

if (clanTag) {

GM\_setClipboard(clanTag);

console.log(`已复制军团标: ${clanTag}`);

showFloatingNotification(`成功复制军团标：${clanTag}`, true);

} else {

console.log('未找到军团标，当前用户名: ' + username);

showFloatingNotification('未找到军团标', false);

}

});

}

}

// 优化性能支持多次添加

const observer = new MutationObserver((mutations) => {

mutations.forEach((mutation) => {

const contextMenus = document.querySelectorAll('.ContextMenuStyle-menu');

contextMenus.forEach(contextMenu => {

if (contextMenu && !contextMenu.dataset.processed) {

const spans = contextMenu.querySelectorAll('span');

const copyNameOption = Array.from(spans)

.find(span => ['复制名称', 'Copy Name', 'Copiar Nombre'].includes(span.textContent.trim()))?.parentElement;

const usernameElement = contextMenu.querySelector('span.Common-whiteSpaceNoWrap');

const currentUsername = usernameElement ? usernameElement.textContent.trim() : '';

addCopyClanTagOption(contextMenu, copyNameOption, currentUsername);

contextMenu.dataset.processed = 'true'; // 标记已处理，避免重复添加

}

});

});

});

observer.observe(document.body, {

childList: true,

subtree: true

});

window.addEventListener('beforeunload', () => {

observer.disconnect();

});

document.addEventListener('click', () => {

const contextMenus = document.querySelectorAll('.ContextMenuStyle-menu');

contextMenus.forEach(menu => delete menu.dataset.processed);

});

})();

整合以上两部分，并优化代码

最终给我一个全新的脚本

要求：

1.删去原有的复制名字选项

2.添加复制ID按钮，实现复制ID的功能

其中[]包裹住的是团标，[]右边的是ID

如[S.H-T]为团标,QJ-TST为ID

3.不影响原有复制军团标功能，修改按钮为“复制团标”

4.复制ID，客服链接跳转实现与复制团标类似的提示信息

5.点击客服链接时先复制id，再跳转链接

6.复制ID位于复制团标按钮之上，同时复制团标按钮位于呼出客服的两个按钮之上

7.复制ID和复制团标的按钮的悬停效果为相应按钮区域背景整个变色，而客服部分则是对应区域一半变色

8.要求复制ID和复制团标按钮的长度和宽度与右键菜单中其他按钮完全对其

9.时刻牢记你是一名严谨的油猴脚本编写大师，检查你写好的代码，并进行优化，至少重复五i遍