PopupMixin API Reference

Shadow DOM 기반 팝업 시스템. 차트(ECharts), 테이블(Tabulator) 통합 지원.

개요

3D 컴포넌트 클릭 시 상세 정보를 Shadow DOM 팝업으로 표시.

```
3D 오브젝트 클릭 → showDetail() → 팝업 표시 + 데이터 로드 → 렌더링
```

빠른 시작 (PDU 예제)

1단계: Mixin 임포트 및 적용

```
const { applyShadowPopupMixin, applyEChartsMixin, applyTabulatorMixin } = PopupMixin;

// 팝업 기본 설정
applyShadowPopupMixin(this, {
    getHTML: () => this.getPopupHTML(),
    getStyles: () => this.getPopupStyles(),
    onCreated: () => this.onPopupCreated(),
});

// 차트 사용 시
applyEChartsMixin(this);

// 테이블 사용 시
applyTabulatorMixin(this);
```

2단계: HTML/CSS 제공 함수

publishCode에서 템플릿 추출:

```
function extractTemplate(htmlCode, templateId) {
    const parser = new DOMParser();
    const doc = parser.parseFromString(htmlCode, 'text/html');
    const template = doc.querySelector(`template#${templateId}`);
    return template?.innerHTML || '';
}

const { htmlCode, cssCode } = this.properties.publishCode || {};

this.getPopupHTML = () => extractTemplate(htmlCode, 'popup-pdu');
this.getPopupStyles = () => cssCode || '';
```

3단계: 팝업 생성 콜백

```
this.onPopupCreated = function() {
    // 차트 생성
    this.createChart('.chart-container');

    // 테이블 생성
```

```
    this.createTable('.table-container', {
        columns: [
            { title: 'Name', field: 'name' },
            { title: 'Value', field: 'value' }
        ]
    });

    // 이벤트 바인딩
    this.bindPopupEvents({
        click: {
            '.close-btn': () => this.hidePopup(),
            '.tab-btn': (e) => this.switchTab(e.target.dataset.tab)
        }
    });
};
```

4단계: Public Methods

```
// 팝업 표시 + 데이터 로드
this.showDetail = function() {
    this.showPopup();

    fetchData(this.page, 'assetDetail', { assetKey: this.assetKey })
        .then((response) => {
            this.renderInfo(response);
            this.updateChart('.chart-container', chartOption);
            this.updateTable('.table-container', tableData);
        });
};

// 팝업 숨김
this.hideDetail = function() {
    this.hidePopup();
};
```

---

applyShadowPopupMixin

기본 Shadow DOM 팝업.

옵션

```
applyShadowPopupMixin(this, {
    getHTML: () => string,      // 팝업 HTML
    getStyles: () => string,    // 팝업 CSS
    onCreated: (shadowRoot) => void  // 생성 완료 콜백 (선택)
});
```

제공 메서드

메서드	설명
createPopup()	팝업 생성 (자동 호출됨)

showPopup()	팝업 표시
hidePopup()	팝업 숨김
popupQuery(selector)	Shadow DOM 내 요소 선택
popupQueryAll(selector)	Shadow DOM 내 모든 요소 선택
bindPopupEvents(events)	이벤트 바인딩
destroyPopup()	팝업 및 리소스 정리

popupQuery 사용

```
// 단일 요소
const title = this.popupQuery('.pdu-name');
title.textContent = 'PDU-001';

// 여러 요소
const buttons = this.popupQueryAll('.tab-btn');
fx.each((btn) => btn.classList.remove('active'), buttons);
```

bindPopupEvents 사용

```
this.bindPopupEvents({
    click: {
        '.close-btn': () => this.hidePopup(),
        '.tab-btn': (e) => this.switchTab(e.target.dataset.tab),
        '.refresh-btn': () => this.refreshData()
    },
    change: {
        'select.filter': (e) => this.applyFilter(e.target.value)
    }
});
```

---

applyEChartsMixin

ECharts 차트 관리. `applyShadowPopupMixin` 이후 호출.

제공 메서드

메서드	설명
createChart(selector)	차트 인스턴스 생성
getChart(selector)	차트 인스턴스 조회
updateChart(selector, option)	차트 옵션 업데이트

사용 예시

```
// onPopupCreated에서 차트 생성
this.createChart('.chart-container');
```

```
// 데이터 로드 후 업데이트
this.updateChart('.chart-container', {
    xAxis: { data: ['Mon', 'Tue', 'Wed'] },
    series: [{ type: 'line', data: [120, 200, 150] }]
});
```

Chart Config 패턴

```
this.chartConfig = {
    xKey: 'timestamps',
    styleMap: {
        power: { label: '전력', unit: 'kW', color: '#3b82f6', smooth: true },
        current: { label: '전류', unit: 'A', color: '#f59e0b', smooth: true },
    },
    optionBuilder: getChartOption,  // 옵션 생성 함수
};

// 렌더링 시
const option = this.chartConfig.optionBuilder(this.chartConfig, data);
this.updateChart('.chart-container', option);
```

---

applyTabulatorMixin

Tabulator 테이블 관리. `applyShadowPopupMixin` 이후 호출.

제공 메서드

메서드	설명
createTable(selector, options)	테이블 인스턴스 생성
getTable(selector)	테이블 인스턴스 조회
isTableReady(selector)	초기화 완료 여부
updateTable(selector, data)	테이블 데이터 업데이트
updateTableOptions(selector, options)	컬럼 등 옵션 변경

사용 예시

```
// onPopupCreated에서 테이블 생성
this.createTable('.table-container', {
    columns: [
        { title: 'ID', field: 'id' },
        { title: 'Name', field: 'name' },
        { title: 'Status', field: 'status',
          formatter: (cell) => {
              const value = cell.getValue();
              const color = value === 'active' ? '#22c55e' : '#888';
              return `<span style="color:${color}">${value}</span>`;
          }
        }
    ],
    layout: 'fitColumns',
```

```
    placeholder: 'No data'
});

// 데이터 로드 후 업데이트
this.updateTable('.table-container', [
    { id: 1, name: 'Circuit 1', status: 'active' },
    { id: 2, name: 'Circuit 2', status: 'inactive' }
]);
```

Table Config 패턴

```
this.tableConfig = {
    selector: '.table-container',
    columns: [
        { title: 'ID', field: 'id' },
        { title: 'Name', field: 'name' },
    ],
    optionBuilder: (columns) => ({
        layout: 'fitColumns',
        placeholder: 'No data',
        columns
    })
};

// onPopupCreated에서
const options = this.tableConfig.optionBuilder(this.tableConfig.columns);
this.createTable(this.tableConfig.selector, options);
```

Shadow DOM CSS 자동 주입

Tabulator CSS가 자동으로 Shadow DOM에 주입됩니다.

- 테마: midnight (다크 모드)
- 경로: `client/common/libs/tabulator/tabulator_midnight.min.css`

전체 구조 예시 (PDU)

```
const { bind3DEvents, fetchData } = Wkit;
const { applyShadowPopupMixin, applyEChartsMixin, applyTabulatorMixin } = PopupMixin;

initComponent.call(this);

function initComponent() {
    // 1. 데이터 설정
    this.datasetInfo = [
        { datasetName: 'assetDetail', render: ['renderInfo'] },
        { datasetName: 'circuits', render: ['renderTable'] },
    ];

    // 2. Config 정의
    this.tableConfig = {
        selector: '.table-container',
        columns: [...],
        optionBuilder: getTableOption,
```

```
    };

    // 3. 렌더링 함수 바인딩
    this.renderInfo = renderInfo.bind(this);
    this.renderTable = renderTable.bind(this);

    // 4. Public Methods
    this.showDetail = showDetail.bind(this);
    this.hideDetail = hideDetail.bind(this);

    // 5. 3D 이벤트
    this.customEvents = { click: '@assetClicked' };
    bind3DEvents(this, this.customEvents);

    // 6. 팝업 설정
    const { htmlCode, cssCode } = this.properties.publishCode || {};

    applyShadowPopupMixin(this, {
        getHTML: () => extractTemplate(htmlCode, 'popup-pdu'),
        getStyles: () => cssCode,
        onCreated: () => onPopupCreated.call(this),
    });

    applyEChartsMixin(this);
    applyTabulatorMixin(this);
}

function onPopupCreated() {
    this.createChart('.chart-container');
    this.createTable('.table-container', this.tableConfig.optionBuilder(this.tableConfig.columns));
    this.bindPopupEvents({
        click: {
            '.close-btn': () => this.hideDetail(),
        }
    });
}

function showDetail() {
    this.showPopup();

    fx.go(
        this.datasetInfo,
        fx.each(({ datasetName, render }) =>
            fetchData(this.page, datasetName, { assetKey: this._assetKey })
                .then((res) => fx.each((fn) => this[fn](res), render))
        )
    );
}

function hideDetail() {
    this.hidePopup();
}
```

리소스 정리

`destroyPopup()` 호출 시 자동 정리:

- 바인딩된 이벤트 리스너
- ECharts 인스턴스 + ResizeObserver
- Tabulator 인스턴스 + ResizeObserver
- Shadow DOM 호스트

```
// beforeDestroy에서
this.destroyPopup();
```

관련 문서

- COMPONENT_MIXIN_API.md - 컴포넌트 Mixin
- WKIT_API.md - bind3DEvents, fetchData
- Projects/ECO/page/components/PDU - 구현 예제