

DTDWeb(Babylon.js) 웹페이지 연동 가이드

XR 기반의 발전소 3D 모델 다자간 동시 협업시스템 개발

버전	날짜	내용	비고
1.0	2024-07-12	신규작성	

목차

개요	3
2D JSON 파일 Load & Close	3
DTDPlayer.OpenURL(uris, isUrl)	3
DTDPlayer.CloseAll()	3
컨텍스트 메뉴	4
DTDPlayer.On2DView(rootValveParameters) Callback	5
DTDPlayer.On2DDesignCheckResult(rootValveParameters) Callback	5
DTDPlayer.On3DDesignCheckResult(rootValveParameters) Callback	5
DTDPlayer.OnFieldDesignCheckResult(rootValveParameters) Callback	5
카메라 이동	6
DTDPlayer.MoveCameraToRootValvePosition(rootValveTag)	6

개요

DTDWeb Babylon.js 버전(이하 DTDWeb)은 공통 기능(DTDWeb(BabylonJs) 웹페이지 연동 가이드.pdf 참조) 외에 고객사 커스텀 기능을 제공하고 타 고객사 프로젝트에 포함되지 않도록 관리됨.

본 문서는 XR 기반의 발전소 3D 모델 다자간 동시 협업시스템 개발(이하 고객사 프로젝트)에서 DTDWeb 과 웹페이지 간 커스텀 기능을 연동하는 방법에 대해 설명함.

2D JSON 파일 Load & Close

본 고객사 프로젝트에서는 2D CAD 데이터(JSON)를 파일로 로드 할 수 있는 OpenURL 과 로드된 JSON 파일을 닫을 수 있는 CloseAll 이 존재함(각 파일별로 닫을 수 있는 기능은 지원하지 않음).

DTDPlayer.OpenURL(uris, isUrl)

JSON 파일을 로드. URL(HTTP)과 <input type="file">을 지원함. DTDX 파일 로드와 동일함. DTDX 파일과 달리 JSON 파일은 각 부재(이하 심볼(Symbol))들이 별도의 DTDX 파일로 로드 될 수 있어야 하며, 본 문서 작성 기준 웹페이지(index.html) 정적 파일 최상단 기준으로 상대경로 /DTD/Symbols 에 위치시켜야 함. 이는 초기 데이터 세팅 결과이며 데이터베이스 관리 시 세팅 기능 추가 요구됨.

파라미터	타입	설명
uris	string[]	URI 를 배열 형태로 전달해 JSON, DTDX 파일 로드
isUrl	boolean	true: URL 타입, false: Input File 타입으로 기본값 true

사용 예) 1. URL

```
player.OpenURL(['http://localhost:9000/DTD/ 5-321-D-166-CS302.JSON'], true);
```

사용 예) 2. 파일(로컬 파일의 절대 경로를 지원하지 않으며 <input type="file">에서 전달 해야함).

```
const files = event.target.files; // <input> DOM 요소 필요  
player.OpenURL(files, false);
```

DTDPlayer.CloseAll()

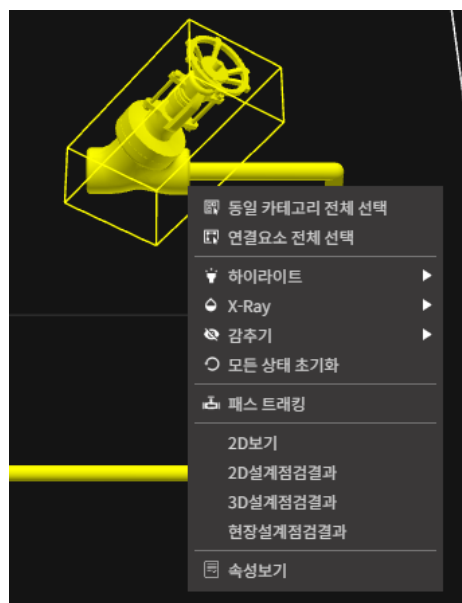
현재 로드된 모든 JSON, DTDX 파일을 닫음. 예제는 아래와 같음.

사용 예)

```
player.CloseAll();
```

컨텍스트 메뉴

본 고객사 프로젝트에서는 심볼 중 루트 밸브(Root Valve)가 존재하며, 다른 객체와 메뉴가 상이함. 2D 보기, 2D 설계점검결과, 3D 설계점검결과, 현장설계점검결과를 추가로 가지며 각 메뉴별 Callback 함수를 제공함.



아래 Callback 함수들은 루트 밸브의 속성을 Key-Value 파라미터로 가지며 예제의 결과화면은 아래와 같이 동일함.

```
{아이디: 'CC21', 파일명: '5-321-D-166-CS302.JSON',
형: 'Y-GLOBE VALVE_3_4', ...}
IFC 미리 정의된 유형: "null"
IFC로 내보내기: "유형별"
IfcGUID: "007vSBW6LFm8HV4ybsTYcu"
TAG: "V2010"
다음으로 IFC로 내보내기: "null"
레벨: "1F"
레벨로부터의 높이: "0.000 mm"
마크: "null"
뷰이름: "{3D}"
생성 공정: "신축"
아이디: "CC21"
이름: "Y-GLOBE VALVE_3_4"
체적: "0.000 m³"
카테고리: "일반 모델"
파일명: "5-321-D-166-CS302.JSON"
패밀리 유형: "Y-GLOBE VALVE_3_4"
패밀리 이름: "Y-GLOBE VALVE_3_4"
해설: "null"
호스트: "레벨 : 1F"
호스트에서의 간격막무기: "0.000 mm"
▶ [[Prototype]]: Object
```

DTDPlayer.On2DView(rootValveParameters) Callback

컨텍스트 메뉴 2D 보기 버튼 클릭 시 호출됨.

정의 예)

```
player.On2DView = (rootValveParameters) => {  
    console.log(rootValveParameters);  
};
```

DTDPlayer.On2DDesignCheckResult(rootValveParameters) Callback

컨텍스트 메뉴 2D 설계점검결과 버튼 클릭 시 호출됨.

정의 예)

```
player.On2DDesignCheckResult = (rootValveParameters) => {  
    console.log(rootValveParameters);  
};
```

DTDPlayer.On3DDesignCheckResult(rootValveParameters) Callback

컨텍스트 메뉴 3D 설계점검결과 버튼 클릭 시 호출됨.

정의 예)

```
player.On3DDesignCheckResult = (rootValveParameters) => {  
    console.log(rootValveParameters);  
};
```

DTDPlayer.OnFieldDesignCheckResult(rootValveParameters) Callback

컨텍스트 메뉴 현장설계점검결과 버튼 클릭 시 호출됨.

정의 예)

```
player.OnFieldDesignCheckResult = (rootValveParameters) => {  
    console.log(rootValveParameters);  
};
```

카메라 이동

DTDPlayer.MoveCameraToRootValvePosition(rootValveTag)

루트 밸브 속성 중 TAG 를 사용해 루트 밸브로 카메라를 이동시킴.

파라미터	타입	설명
rootValveTag	string	루트 밸브 TAG

사용 예)

```
player.MoveCameraToRootValvePosition(rootValveTag);
```

