

DTDWeb(Babylon.js) 웹페이지 연동 가이드

버전	날짜	내용	비고
1.0	2024-07-10	신규작성	

목차

개요	6
DTDWeb 구성도	6
DTDPlayer Element	6
DTDPlayer.InitializeOptions(options)	7
DTDX 파일 Load & Close	8
DTDPlayer.OpenURL(uris, isUrl)	8
DTDPlayer.OnContentsAllLoaded() Callback	8
DTDPlayer.CloseAll()	8
Function Mode	9
DTDPlayer.GetFunctionMode()	9
DTDPlayer.SetFunctionMode(functionMode)	10
DTDPlayer.OnFunctionModeChanged(beforeFunctionMode, currentFunctionMode) Callback .	10
Key-Value 파라미터	11
속성 데이터	11
DTDPlayer.GetElementDataList(keyValueJson)	11
모델 선택	12
DTDPlayer.Select(keyValueJson)	12
DTDPlayer.UnSelect(keyValueJson)	13
DTDPlayer.UnSelectAll()	13
DTDPlayer.SelectAndFit(keyValueJson)	13
DTDPlayer.Fit(keyValueJson)	14
모델 효과	15

DTDPlayer.EffectHighlight(keyValueJson).....	15
DTDPlayer.EffectHighlightOther(keyValueJson).....	15
DTDPlayer.EffectXray(keyValueJson).....	16
DTDPlayer.EffectXrayOther(keyValueJson).....	16
DTDPlayer.EffectChangeColor(keyValueJson, colorR, colorG, colorB, alpha)	17
DTDPlayer.EffectChangeColorOther(keyValueJson, colorR, colorG, colorB, alpha)	18
DTDPlayer.EffectInvisible(keyValueJson)	18
DTDPlayer.EffectInvisibleOther(keyValueJson)	19
DTDPlayer.ResetEffect(keyValueJson).....	19
DTDPlayer.ResetAllEffect()	20
카메라 이동 및 제어	20
DTDPlayer.MoveCameraToHome().....	20
DTDPlayer.MoveCameraToUp()	21
DTDPlayer.MoveCameraToFront()	21
DTDPlayer.MoveCameraToBack().....	22
DTDPlayer.MoveCameraToLeft().....	22
DTDPlayer.MoveCameraToRight()	23
DTDPlayer.MoveCameraToTransformJSON(cameraTransform, onAfterAnimateEndCallback).....	23
DTDPlayer.PlayEffectXrayCoveredCamera(keyValueJson)	24
DTDPlayer.StopEffectXrayCoveredCamera()	24
GUI	25
DTDPlayer.GetPresetColors()	25
DTDPlayer.GetMarkupClipartImageUrls()	25

DTDPlayer.ShowToastMessage(message, showSeconds)	26
DTDPlayer.HideToastMessage()	26
마크업	27
DTDPlayer.StartMarkup()	27
DTDPlayer.StopMarkup()	27
DTDPlayer.OnAddMarkupSuccess(markupType) Callback	27
DTDPlayer.OnAfterMarkupAction(hasPrevious, hasNext) Callback	28
DTDPlayer.UndoMarkup()	29
DTDPlayer.RedoMarkup()	29
DTDPlayer.ClearMarkups()	29
DTDPlayer.SetMarkupPenColor(hexColor)	29
DTDPlayer.SetMarkupPenThickness(thickness)	30
DTDPlayer.SetMarkupFontColor(hexColor)	30
DTDPlayer.SetMarkupFontSize(fontSize)	30
DTDPlayers.SetMarkupClipartIndex(clipartIndex)	31
DTDPlayer.GetMarkupCurrentOptions()	31
DTDPlayers.SaveMarkup(onSuccessCallback)	31
DTDPlayers.LoadMarkupByURL(url)	32
DTDPlayer.LoadMarkupByBase64String(base64String)	32
패스 트래킹	33
DTDPlayer.PausePathTracking()	33
DTDPlayer.ResumePathTracking()	33
DTDPlayer.StopPathTracking()	34

간섭검사	34
DTDPlayer.OnInterferenceCheckEnd(successResult) Callback	34
DTDPlayer.StartInterferenceCheck(keyValueJson1, keyValueJson2)	35
DTDPlayer.StopInterferenceCheck()	35
DTDPlayer.MoveCameraToInterferencePoint(pointIndex)	35
북마크	36
DTDPlayer.AddBookmarkGroup(onSuccessCallback)	36
DTDPlayer.AddBookmark(groupSequenceld, onSuccessCallback)	37
DTDPlayer.RemoveBookmarkGroup(groupSequenceld, onSuccessCallback)	38
DTDPlayer.RemoveBookmark(groupSequenceld, bookmarkSequenceld, onSuccessCallback) ...	38
DTDPlayer.GetAllBookmarks()	39
DTDPlayer.SetBookmarkGlobalAnimateSpeed(animateSpeed)	40
DTDPlayer.SetBookmarkGlobalAnimateDelay(animateDelay)	40
DTDPlayer.SetBookmarkAnimateSpeed(groupSequenceld, bookmarkSequenceld, animateSpeed)	40
DTDPlayer.SetBookmarkAnimateDelay(groupSequenceld, bookmarkSequenceld, animateDelay)	40
DTDPlayer.PlayBookmarkAnimation(groupSequenceld, isReverse, onStateChangedCallback)...	41
DTDPlayer.StopBookmarkAnimation()	42
DTDPlayer.SaveBookmark()	42
DTDPlayer.LoadBookmark(bookmarks)	43

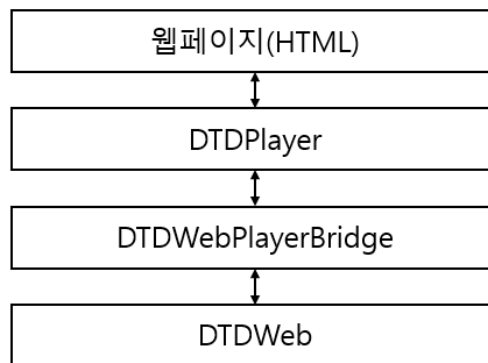
개요

DTDWeb Babylon.js 버전(이하 DTDWeb)은 기존 DTDWeb Unity WebGL 버전의 WASM(Web Assembly)와 다르게 모든 로직이 Javascript 언어로 개발됨. 이에 플러그인을 통한 기존 API 연동 방식에서의 불필요한 변수 변환없이 Javascript 객체 그대로 사용 가능하며 콜백 함수가 최소화됨.

본 문서는 DTDWeb 과 웹페이지 간 연동하는 방법에 대해 설명함. 후술할 기능들의 HTML 작성 GUI 는 각 고객사별로 Custom 요소들이 존재함에 따라 DTDSqaure 의 UI 기준으로 작성함.

DTDWeb 구성도

DTDWeb 은 아래 그림과 같이 구성됨.



웹페이지는 DTDPlayer.js(Class)의 API 를 호출, 내부적으로 DTDPlayer 는 DTDWebPlayerBridge 를 통해 DTDWeb 과 통신함.

DTDPlayer Element

DTDPlayer 는 DTDPlayer.js 와 DOM(Document Object Model), HTML 요소인 Element 을 가짐. 따라서 아래 항목들을 웹페이지 head 와 body 에 추가 필요함.

```
<head>
  <script src="/DTDWeb/js/DTDPlayer.js"></script>
</head>

<body>
  <dtd-player id="dtd-player-container" onload="onLoadDTDPlayer(this);">
  </dtd-player>
</body>
```

HTML <head></head> 내 DTDPlayer.js 를 경로에 맞게 추가하고 <body></body> 내에 Custom HTML Element 인 <dtd-player>를 추가 해야함. <dtd-player>의 onload 는 DTDPlayer 의 onload 시 호출될 함수 정의 해야 하며, 예제는 아래와 같음.

```
<body>
  <dtd-player id="dtd-player-container" onload="onLoadDTDPlayer(this);">
</dtd-player>
<script>
  function onLoadDTDPlayer(playerElement) {
    const player = playerElement.dtdPlayer;
    if (player === undefined) {
      return;
    }

    player.InitializeOptions({
      useFreeze: true,
      useInstance: true,
      usePBRMaterial: false,
      useGridGround: true,
      useScreenKeyHelper: true,
      showDebugLayer: false
    });
  }
</script>
</body>
```

DTDPlayer.js 는 <dtd-player> 로드 후 Dependencies 와 필요 항목들을 DOM 구성 요소에 로드하고, <dtdplayer onload="">에 정의된 함수를 호출함.

예제의 const player = playerElement.dtdPlayer;는 로드 된 DTDPlayer Class 를 객체화하고 이후 API 통신이 가능함.

DTDPlayer.InitializeOptions(options)

DTDPlayer 는 로드 후 설정하는 항목이 존재하며, 파라미터인 options 에 Key-Value 타입으로 값을 전달해야함. 최적화 관련 속성은 기본값을 권장함.

속성명	기본값(미설정 시)	설명
useFreeze	true	Active Meshes Freezing 여부(최적화 관련)
useInstance	true	GPU 인스턴싱 사용 여부(최적화 관련)
usePBRMaterial	false	PBR 재질 사용 여부(색감 관련)
useGridGround	false	모델 바닥에 크기 관련된 정보 표시 여부
useScreenKeyHelper	true	좌하단 Key Helper 표시 여부
showDebugLayer	false	Babylon.js 디버그용 Inspector 표시 여부

DTDX 파일 Load & Close

DTDWeb 은 DTDX 파일을 로드 할 수 있는 OpenURL 과 로드 된 DTDX 파일들을 닫을 수 있는 CloseAll 이 존재함(각 파일별로 닫을 수 있는 기능은 지원하지 않음).

DTDPlayer.OpenURL(uris, isUrl)

DTDX 파일을 로드. URL(HTTP)과 <input type="file">을 지원함.

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

사용 예) 1. URL

```
player.OpenURL(['http://localhost:9000/DTD/GASAN_7F.dtdx'], 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.OnContentsAllLoaded() Callback

DTDPlayer.OpenURL 을 통해서 성공적으로 로드가 끝나면, DTDPlayer.OnContentsAllLoaded 에 정의된 Callback 함수가 호출됨.

정의 예)

```
player.OnContentsAllLoaded = () => {  
  console.log('OnContentsAllLoaded');  
};
```

DTDPlayer.CloseAll()

현재 로드된 모든 DTDX 파일을 닫음. 예제는 아래와 같음.

사용 예)

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

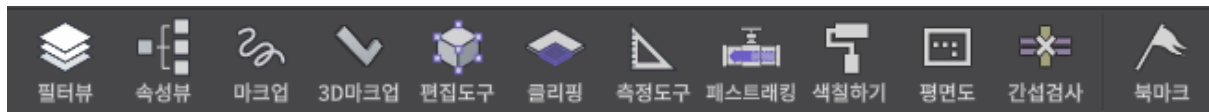

Function Mode

DTDWeb에서는 사용하는 기능(예: 패스트래킹, 간섭검사, 마크업 등)에 따른 모드가 존재함. 본문서 작성 기준으로 아래와 같은 모드들이 존재하며 기능 개발에 따라서 추가되고 있고 고객사 Custom 기능에 따라서 달라짐. 이에 DTDPlayer Class 내에 FunctionMode 확인이 필요함.

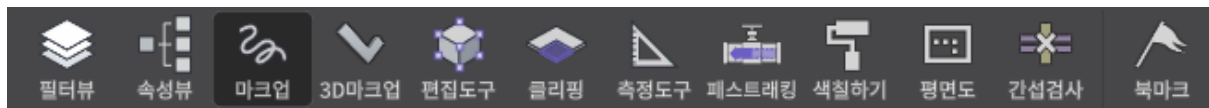
```
class DTDPlayer {
    static FunctionMode = {
        NONE: 0,
        MARKUP: 1,
        PATH_TRACKING: 2,
        INTERFERENCE_CHECK: 3,
        BOOKMARK: 4
    };
}
```

DTDPlayer.GetFunctionMode(), DTDPlayer.SetFunctionMode(functionMode) 함수를 제공하며, GUI(Toolbar) 연동 시 호출이 필요함. 기능들 사용 시, 모드 변경은 DTDWeb 내부적으로 자동 설정됨.

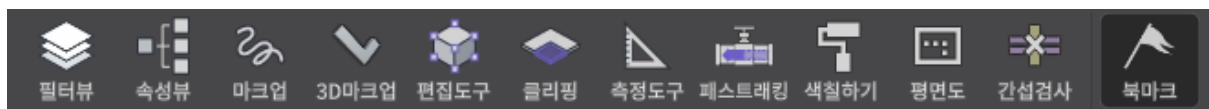
DTDPlayer의 현재 FunctionMode가 NONE일 때



DTDPlayer의 현재 FunctionMode가 MARKUP일 때



DTDPlayer의 현재 FunctionMode가 BOOKMARK일 때



DTDPlayer.GetFunctionMode()

현재 DTDPlayer의 FunctionMode를 넘겨주며 위 기술된 FunctionMode의 Key-Value 중 Value(number 타입)를 결과로 넘겨줌.

결과 타입	설명
number	DTDPlayer 현재 FunctionMode

사용 예)

```
const currentFunctionMode = player.GetFunctionMode();  
console.log(currentFunctionMode); // NONE 모드면 0, 마크업 모드면 1, ...
```

DTDPlayer.SetFunctionMode(functionMode)

GUI(예: Toolbar)연동 시, 모드를 설정함.

파라미터	타입	설명
functionMode	String[]	URI 를 배열 형태로 전달해 DTDX 파일 로드

사용 예)

```
player.SetFunctionMode(DTDPlayer.FunctionMode.NONE); // 기능 선택이 되지 않았을 때  
player.SetFunctionMode(DTDPlayer.FunctionMode.MARKUP); // 마크업 모드가 선택 됐을 때
```

DTDPlayer.OnFunctionModeChanged(beforeFunctionMode, currentFunctionMode)
Callback

FunctionMode 변경 시 호출되는 Callback 함수로 이전과 변경된(현재) FunctionMode 의 Index 를 파라미터로 전달함.

파라미터	타입	설명
beforeFunctionMode	number	이전 FunctionMode Index
currentFunctionMode	number	변경된 FunctionMode Index

정의 예)

```
player.OnFunctionModeChanged = (beforeFunctionMode, currentFunctionMode) => {  
    console.log(`OnFunctionModeChanged before:${beforeFunctionMode}  
current:${currentFunctionMode}`);  
};
```

다음은 FunctionMode 가 NONE(0)에서 MARKUP(1)로, MARKUP 에서 NONE 으로 변경된 결과임.

```
OnFunctionModeChanged before:0, current:1
```

```
OnFunctionModeChanged before:1, current:0
```

Key-Value 파라미터

후술할 함수들의 Key-Value 파라미터는 `[[{key: '아이디', value: '123456'}], [{key: '아이디', value: '7891011'}]]`와 같은 2 중 배열의 타입을 가짐. 로드된 DTDX 파일의 속성 데이터를 사용하며 AND, OR, 정규식 등을 지원함. 기본정보 Key(아이디, 파일명, 카테고리, 패밀리 이름, 패밀리 유형 등)에 한해 한글/영어를 지원하며 영어 Key 사용 시 대소문자 구분 없음. 사용 예제는 아래와 같음.

예제 1. 단일 항목(아이디가 123456 인 객체)

```
[ [ {key: '아이디', value: '123456'} ] ]
```

예제 2. AND(아이디가 123456 이면서 카테고리가 배관인 경우)

```
[ [ {key: '아이디', value: '123456'}, {key: '카테고리', value: '배관'} ] ]
```

예제 3. OR(아이디가 123456 또는 7891011 인 경우)

```
[ [ {key: '아이디', value: '123456'} ], [ {key: '아이디', value: '7891011'} ] ]
```

예제 4. AND & OR(아이디가 123456 이면서 카테고리가 배관인 경우 혹은 아이디가 7891011 인 경우)

```
[ [ {key: '아이디', value: '123456'}, {key: '카테고리', value: '배관'} ], [ {key: '아이디', value: '7891011'} ] ]
```

예제 5. 정규식(패밀리 유형이 STB 로 시작하는 문자열인 경우)

```
[ [ {key: '패밀리 유형', value: '/^STB/' } ] ]
```

기술한 예제처럼 OR 는 첫 번째 배열에, AND 는 두 번째 배열에 Key-Value 값을 추가해 사용해야 하고 정규식은 Regex 타입이 아닌 string 타입임을 주의해야 함.

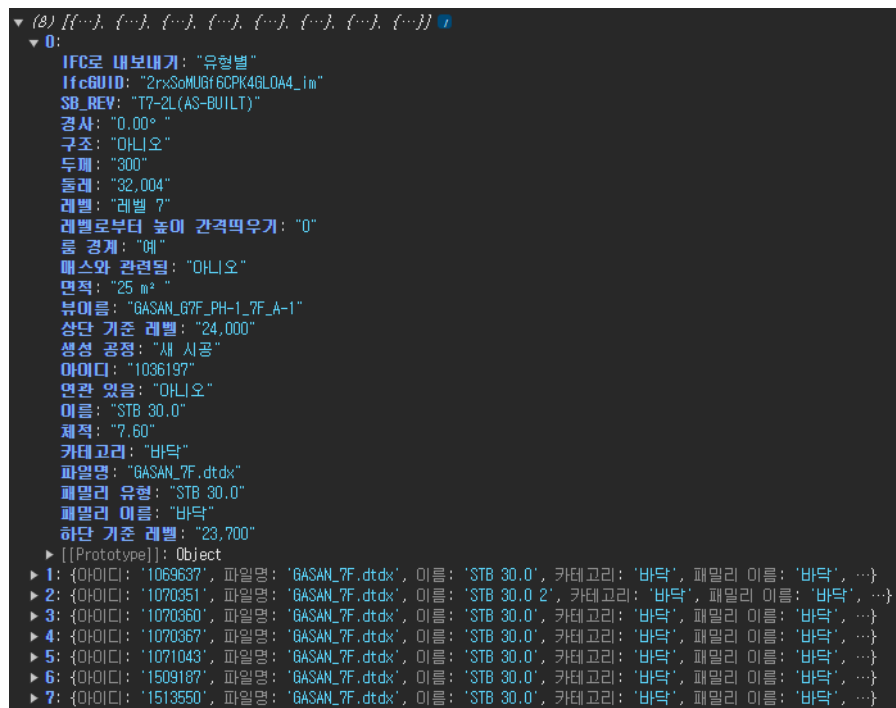
속성 데이터

`DTDPlayer.GetElementDataList(keyValueJson)`

모델의 속성을 가져오는 함수. 파라미터가 비어 있으면 로드된 모든 모델의 속성을 가지고 옴.

파라미터	타입	설명
keyValueJson	Key-Value 파라미터	대상 Key-Value 파라미터

결과 타입	설명
[{}]	keyJsonValue 으로 검색된 모델의 속성 데이터



사용 예)

```
// ID 가 123456 인 모델의 속성 데이터
const id123456 = player.GetElementDataList([{{key: 'id', value: '123456'}}]);

// 패밀리 유형이 STB 로 시작하는 모델의 속성 데이터
const stbs = player.GetElementDataList([{{key: 'Family Type', value: '/^STB/'}}]);

// 전체 모델의 속성 데이터
const parameterList = player.GetElementDataList();
```

모델 선택

모델 선택과 카메라 이동 및 맞춤 함수들을 제공함.

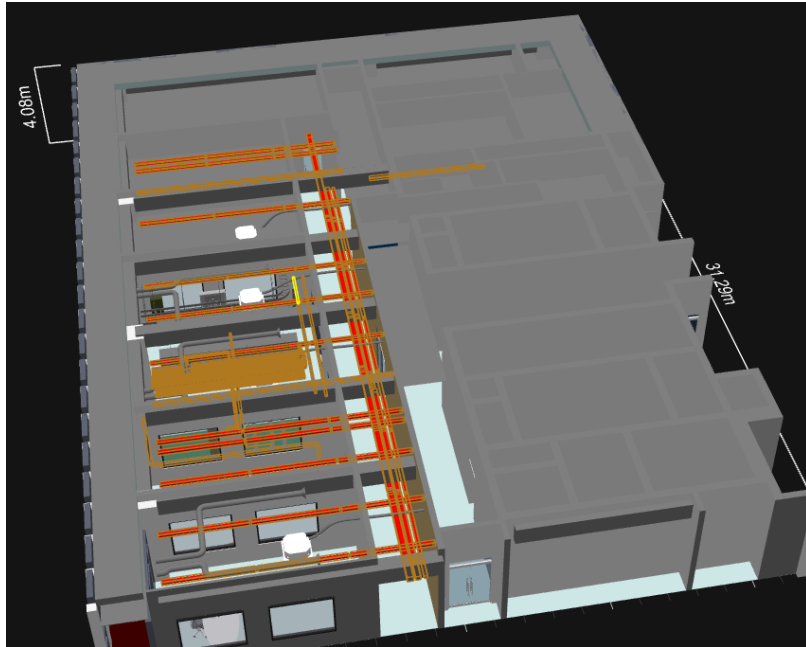
DTDPlayer.Select(keyJsonValue)

Key-Value 파라미터 대상 모델 선택. 내부 구조상 첫 번째 검색 결과는 노랑색으로 나머지 검색 결과는 갈색으로 표시.

파라미터	타입	설명
keyValueJson	Key-Value 파라미터	대상 Key-Value 파라미터

사용 예)

```
player.Select([[{key: '카테고리', value: '배관'}]]); // 카테고리가 배관인 모델 선택
```



DTDPlayer.UnSelect(keyValueJson)

Key-Value 파라미터 대상 모델 선택 해제.

사용 예)

```
player.UnSelect([[{key: 'id', value: '1502328'}]]); // 아이디 1502328 모델 선택 해제
```

DTDPlayer.UnSelectAll()

모든 선택된 모델 선택 해제.

사용 예)

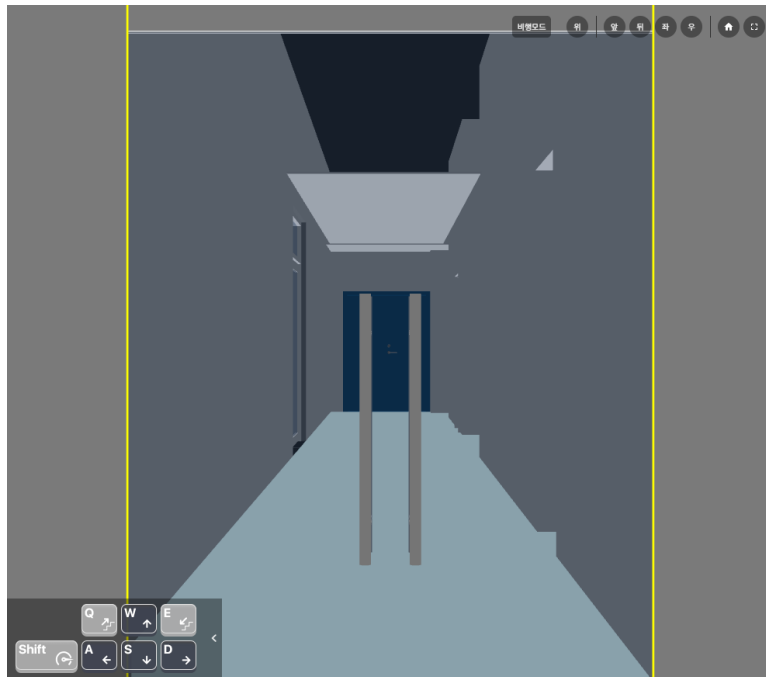
```
player.UnSelectAll();
```

DTDPlayer.SelectAndFit(keyValueJson)

Key-Value 파라미터 대상 모델을 선택 및 카메라 이동.

사용 예)

```
player.SelectAndFit([[{key: 'id', value: '1001008217'}]]);
```

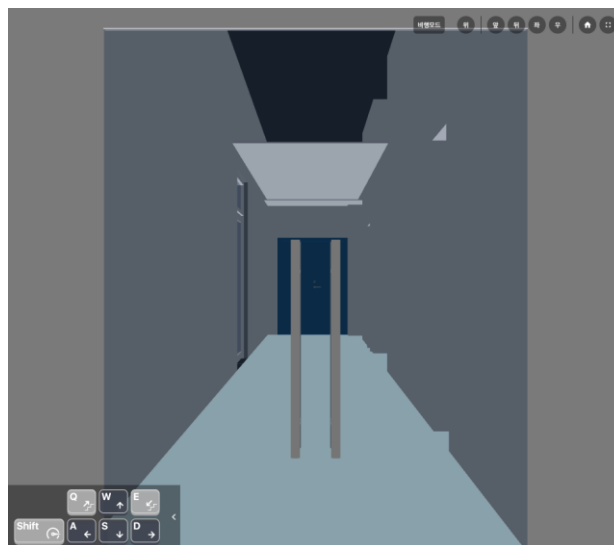


DTDPlayer.Fit(keyValueJson)

Key-Value 파라미터 대상 모델 선택 없이 카메라 이동.

사용 예)

```
player.Fit([[{key: 'id', value: '1001008217'}]]);
```



모델 효과

DTDWeb 에서 모델은 다섯 가지 중 하나의 상태를 가짐.

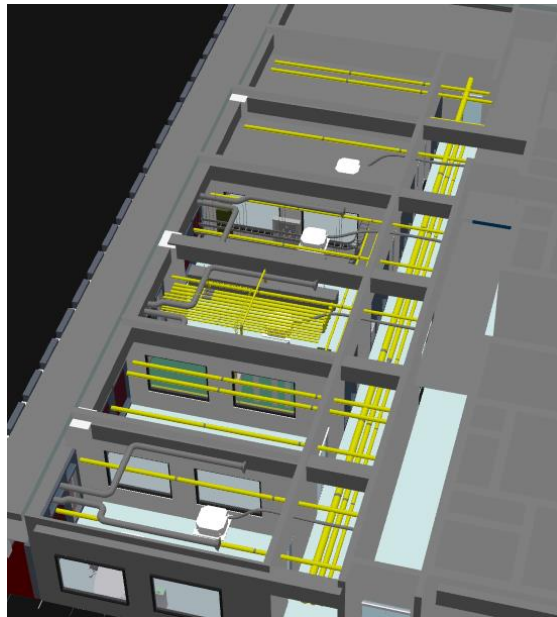
효과	설명
NORMAL	원본(효과 없음)
HIGHLIGHT	하이라이트
XRAY	X-ray
CHANGE_COLOR	색상 변경
INVISIBLE	숨김

DTDPlayer.EffectHighlight(keyValueJson)

Key-Value 파라미터 대상 모델 하이라이트.

사용 예)

```
player.EffectHighlight([[{key: '카테고리', value: '배관'}]]);
```

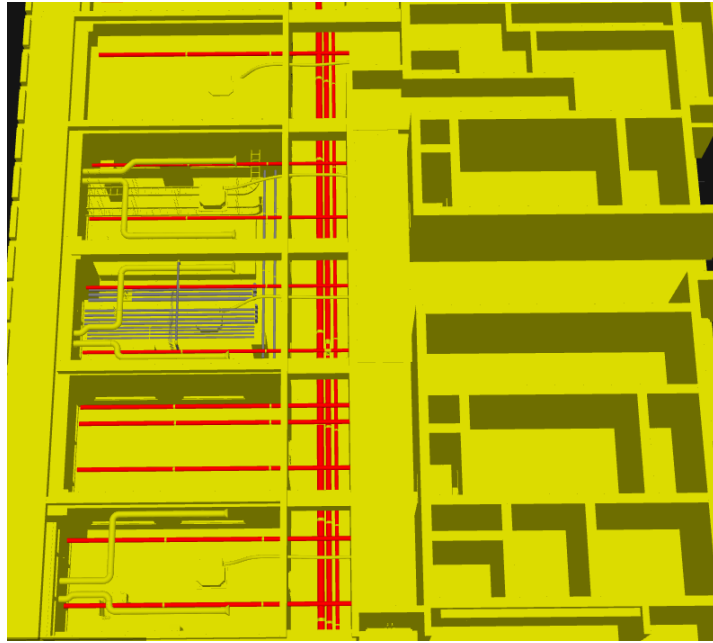


DTDPlayer.EffectHighlightOther(keyValueJson)

Key-Value 파라미터 대상 제외 모델 하이라이트.

사용 예)

```
player.EffectHighlightOther([[{key: '카테고리', value: '배관'}]]);
```

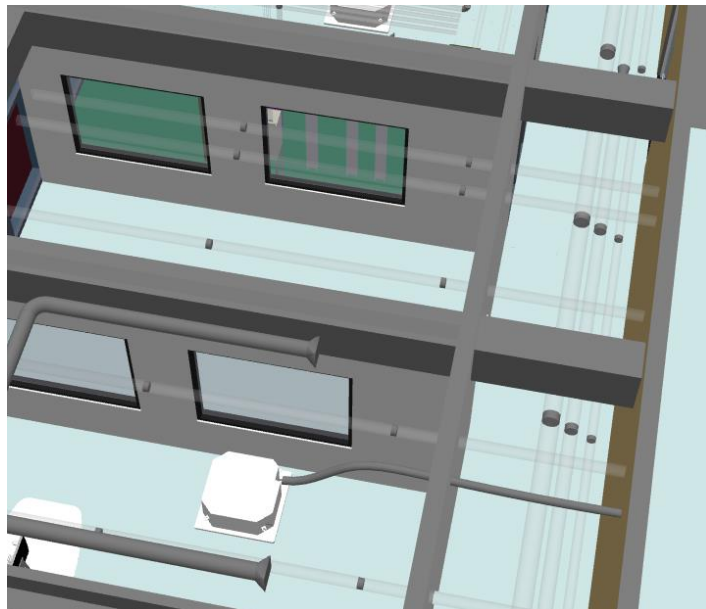


DTDPlayer.EffectXray(keyJsonValue)

Key-Value 파라미터 대상 모델 X-ray.

사용 예)

```
player.EffectXray([[{key: '카테고리', value: '배관'}]]);
```

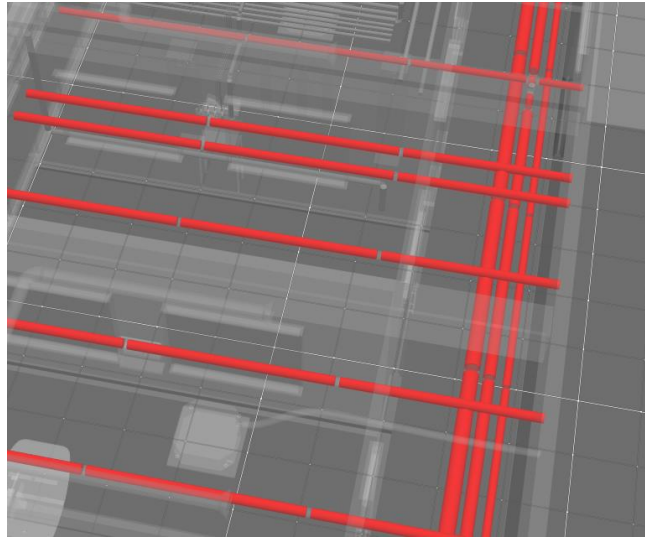


DTDPlayer.EffectXrayOther(keyJsonValue)

Key-Value 파라미터 대상 제외 모델 X-ray

사용 예)


```
player.EffectXrayOther([[{key: '카테고리', value: '배관'}]]);
```



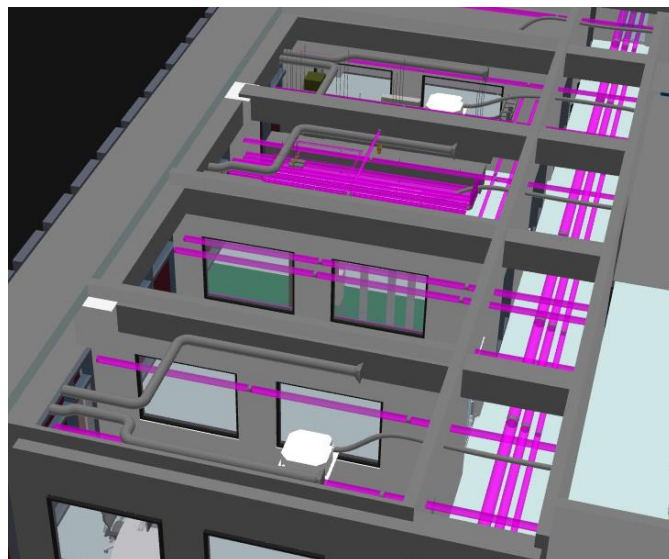
```
DTDPlayer.EffectChangeColor(keyValueJson, colorR, colorG, colorB, alpha)
```

Key-Value 파라미터 대상 모델 색상 변경.

파라미터	타입	설명
keyValueJson	Key-Value 파라미터	대상 Key-Value 파라미터
colorR	number	0~255 빨강색
colorG	number	0~255 파랑색
colorB	number	0~255 녹색
alpha	number	0~255 투명도

사용 예)

```
// 카테고리가 배관인 모델 RGBA(255, 0, 255, 150) 색상 변경
player.EffectChangeColor([[{key: '카테고리', value: '배관'}]], 255, 0, 255, 150);
```



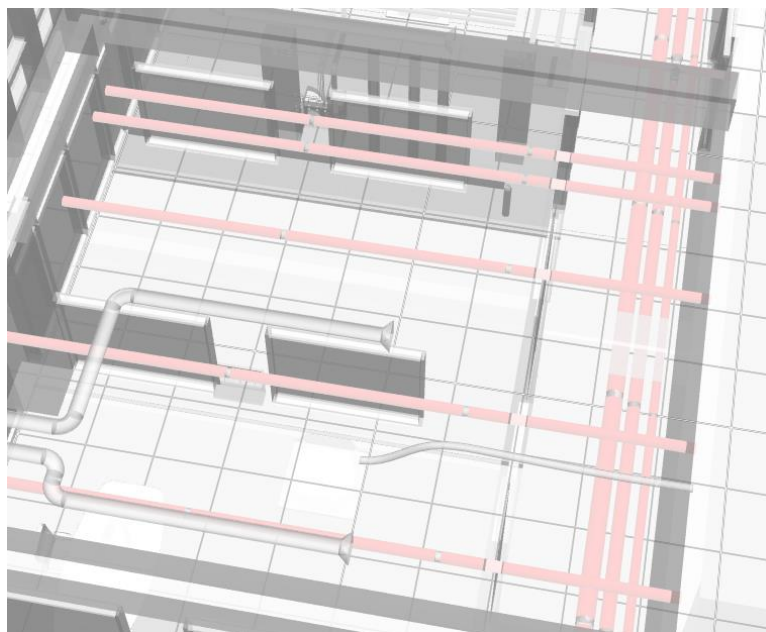
DTDPlayer.EffectChangeColorOther(keyValueJson, colorR, colorG, colorB, alpha)

Key-Value 파라미터 대상 제외 모델 색상 변경.

파라미터	타입	설명
keyValueJson	Key-Value 파라미터	대상 Key-Value 파라미터
colorR	number	0~255 빨강색
colorG	number	0~255 파랑색
colorB	number	0~255 녹색
alpha	number	0~255 투명도

사용 예)

```
// 카테고리가 배관인 모델 제외 흰색 변경
player.EffectChangeColorOther([[{key: '카테고리', value: '배관'}]], 255, 255, 255, 150);
```

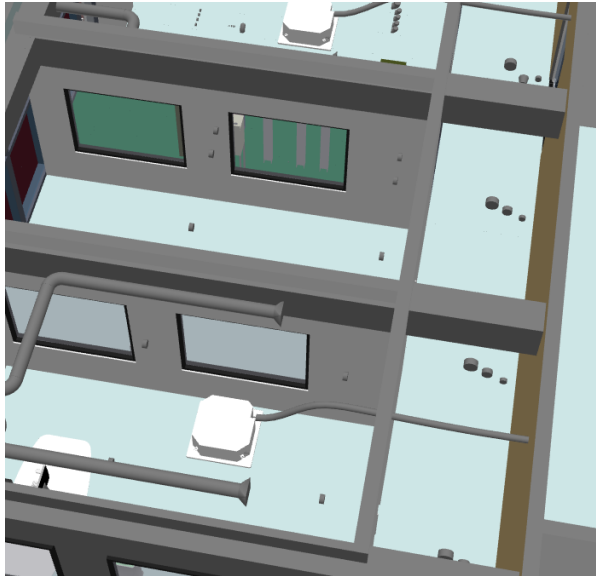


DTDPlayer.EffectInvisible(keyValueJson)

Key-Value 파라미터 대상 모델 숨김.

사용 예)

```
player.EffectInvisible([[{key: '카테고리', value: '배관'}]]);
```

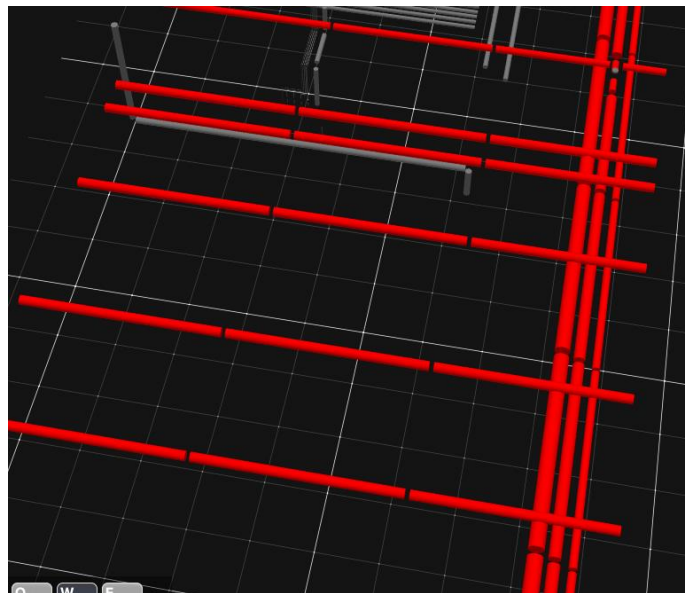


DTDPlayer.EffectInvisibleOther(keyValueJson)

Key-Value 파라미터 대상 제외 모델 숨김.

사용 예)

```
player.EffectInvisibleOther([[{key: '카테고리', value: '배관'}]]);
```

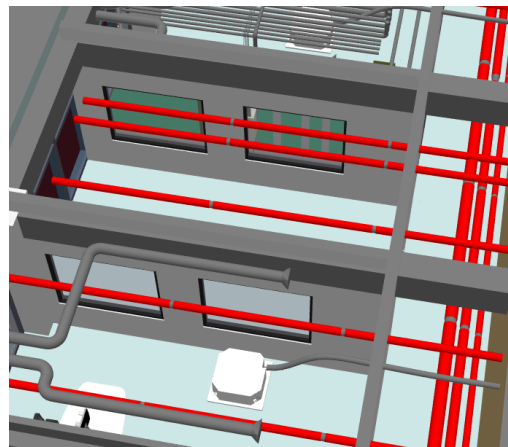
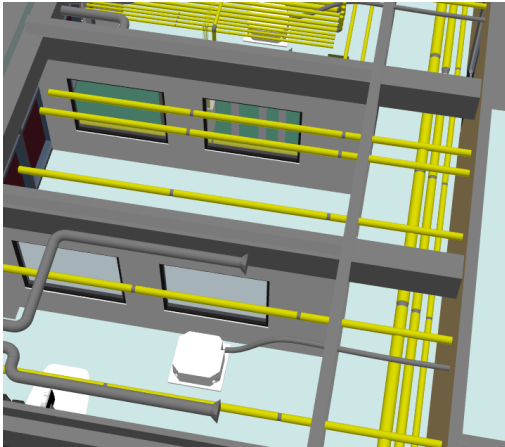


DTDPlayer.ResetEffect(keyValueJson)

Key-Value 파라미터 대상 모델 효과 제거(NORMAL).

사용 예)

```
player.ResetEffect([[{key: '카테고리', value: '배관'}]]);
```



DTDPlayer.ResetAllEffect()

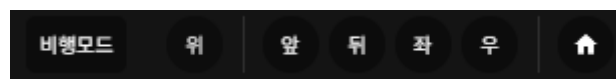
모든 모델 효과 제거(NORMAL).

사용 예)

```
player.ResetAllEffect();
```

카메라 이동 및 제어

앞서 기술한 DTDPlayer.Fit() 이외에 카메라를 이동하고 제어하는 함수들을 제공함. 카메라는 비행/걷기 모드가 존재하며 홈, 위는 비행모드를 앞, 뒤, 좌, 우는 걷기모드로 설정됨.

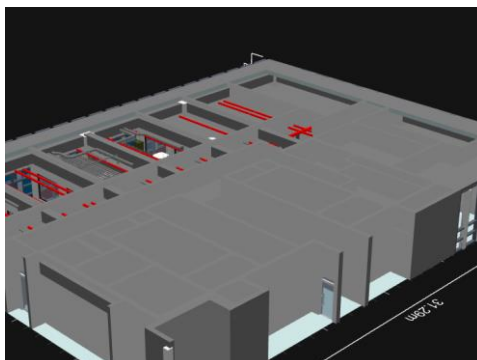


DTDPlayer.MoveCameraToHome()

카메라를 초기 위치로 이동(비행모드).

사용 예)

```
player.MoveCameraToHome();
```

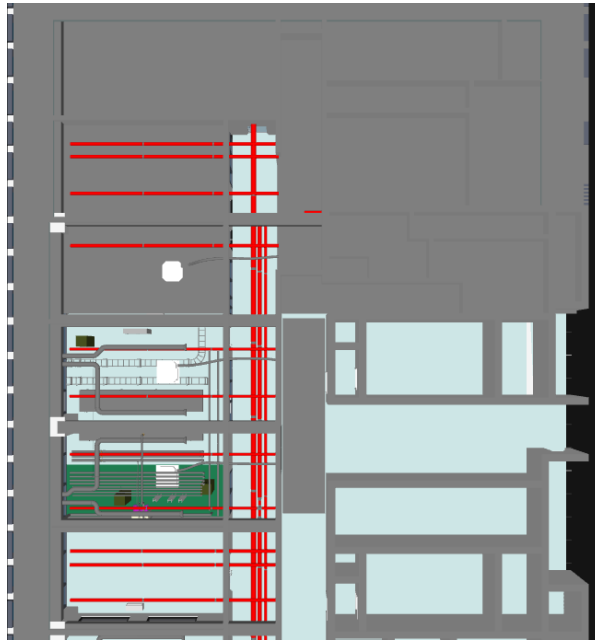


DTDPlayer.MoveCameraToUp()

카메라를 위에서 바라보도록 이동(비행모드, 평면도).

사용 예)

```
player.MoveCameraToUp();
```

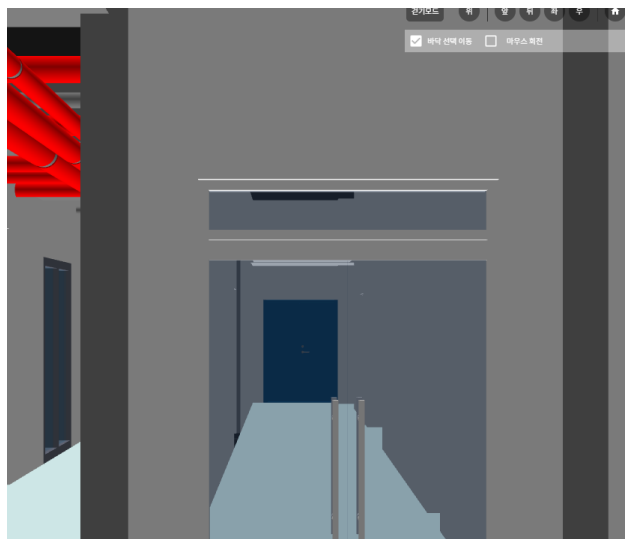


DTDPlayer.MoveCameraToFront()

카메라를 모델 전체의 앞 방향으로 이동(걷기모드).

사용 예)

```
player.MoveCameraToFront();
```

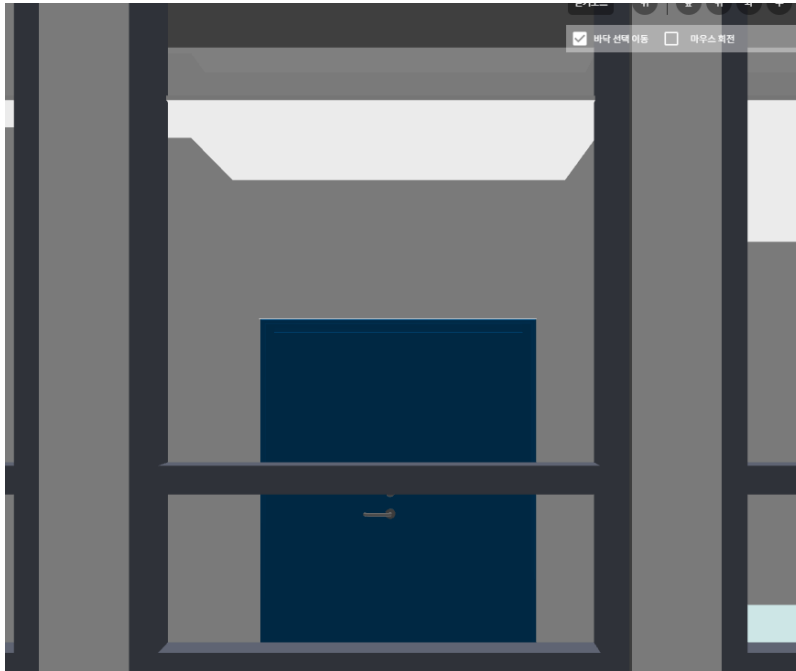


```
DTDPlayer.MoveCameraToBack()
```

카메라를 모델 전체의 뒤 방향으로 이동(걷기모드).

사용 예)

```
player.MoveCameraToBack();
```

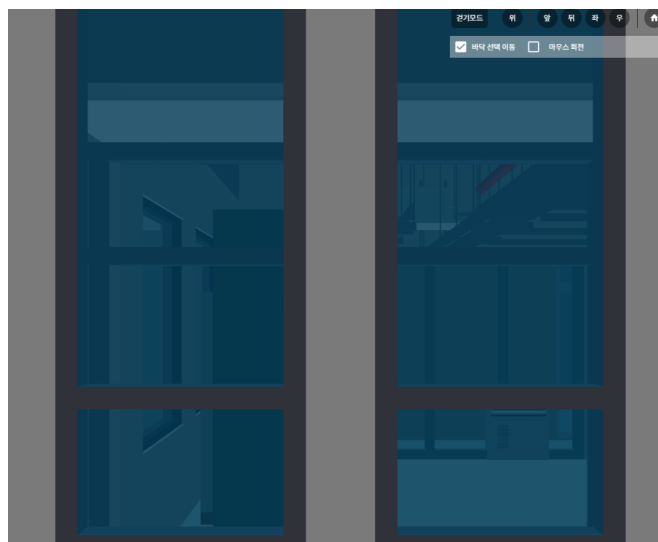


```
DTDPlayer.MoveCameraToLeft()
```

카메라를 모델 전체의 좌 방향으로 이동(걷기모드).

사용 예)

```
player.MoveCameraToLeft();
```



DTDPlayer.MoveCameraToRight()

카메라를 모델 전체의 우 방향으로 이동(걷기모드).

사용 예)

```
player.MoveCameraToRight();
```



DTDPlayer.MoveCameraToTransformJSON(cameraTransform, onAfterAnimateEndCallback)

카메라의 Transform Key-Value 파라미터의 값으로 카메라 애니메이션을 실행하는 함수. 카메라 Transform 정보는 마크업, 북마크 저장 등에서 제공함. 카메라 애니메이션이 종료된 후 onAfterAnimateEndCallback 파라미터 Callback 함수가 호출됨.

파라미터	타입	설명
cameraTransform	Object(Key-Value)	카메라 Transform 정보
onAfterAnimateEndCallback	function	카메라 이동 후 호출 Callback 함수

cameraTransform 의 Key-Value 정보는 아래 표와 같음.

Key(타입)	Value
cameraTarget(x: number, y: number, z: number)	카메라 타겟
cameraPosition(x: number, y: number, z: number)	카메라 위치
cameraRotation(x: number, y: number, z: number, w: number)	카메라 회전 정보

사용 예)

```
const cameraTransform = { // 카메라의 X:0, Y:0, Z:0 좌표로 회전 없이 이동
  cameraTarget: { x: 0, y: 0, z: 0 },
  cameraPosition: { x: 0, y: 0, z: 0 },
  cameraRotation: { x: 0, y: 0, z: 0, w: 0 }
};
player.MoveCameraToTransformJSON(cameraTransform, () => {
  console.log('MoveCameraToTransformJSON Animate Ended.');
```

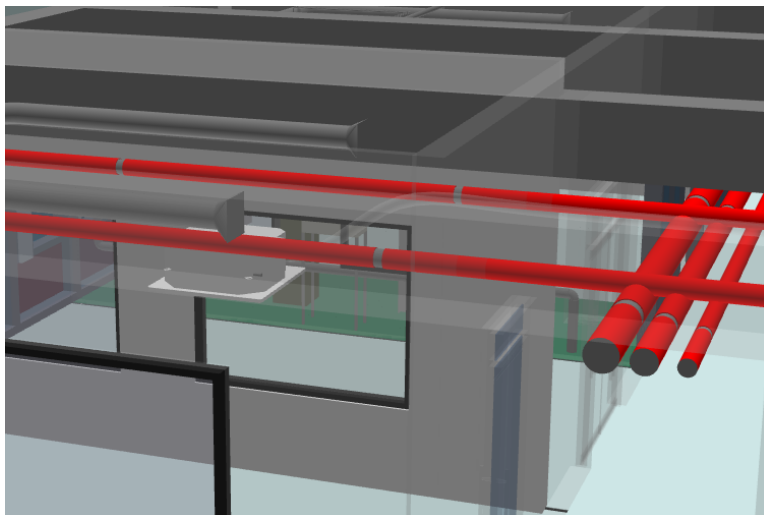
```
});
```

DTDPlayer.PlayEffectXrayCoveredCamera(keyValueJson)

카메라가 Key-Value 파라미터의 대상을 가리고 있는 경우 X-ray 처리.

사용 예)

```
player.PlayEffectXrayCoveredCamera([[{key: '카테고리', value: '배관'}]]);
```



DTDPlayer.StopEffectXrayCoveredCamera()

기실행한 DTDPlayer.PlayEffectXrayCoveredCamera 함수를 종료. 호출 즉시 X-ray 효과 처리됐던 모델들이 이전 효과 상태로 복구됨.

사용 예)

```
player.StopEffectXrayCoveredCamera();
```


GUI

기능 모드의 HTML 페이지 제작 Helper 기능과 UI 기능들을 제공함.

DTDPlayer.GetPresetColors()

기능 모드의 HTML 페이지 제작을 위한 색상 프리셋으로 DTDSquare 의 기능 Panel 들의 색상 값들을 HEX string 배열 형태로 반환함.



사용 예)

```
player.GetPresetColors();
```

```
▼ (10) [ '#FF0000', '#FF8800', '#FFFF00', '#88FF1B', '#008000', '#00B1EF', '#002080', '#70309F', '#000000', '#FFFFFF' ]  
0: "#FF0000"  
1: "#FF8800"  
2: "#FFFF00"  
3: "#88FF1B"  
4: "#008000"  
5: "#00B1EF"  
6: "#002080"  
7: "#70309F"  
8: "#000000"  
9: "#FFFFFF"
```

DTDPlayer.GetMarkupClipartImageUrls()

마크업 모드 시 필요한 클립아트의 Image URL 을 상대 경로 배열로 반환. 해당 배열 Index 순으로 UI 배치 필요.



사용 예)

```
player.GetMarkupClipartImageUrIs();
```

```
0: "/DTDWeb/images/markup/clipart/note_01_cloud.svg"  
1: "/DTDWeb/images/markup/clipart/note_02_cloud.svg"  
2: "/DTDWeb/images/markup/clipart/note_03.svg"  
3: "/DTDWeb/images/markup/clipart/note_03_bg100.svg"  
4: "/DTDWeb/images/markup/clipart/note_04.svg"  
5: "/DTDWeb/images/markup/clipart/note_04_bg100.svg"  
6: "/DTDWeb/images/markup/clipart/note_05.svg"  
7: "/DTDWeb/images/markup/clipart/note_05_bg100.svg"  
8: "/DTDWeb/images/markup/clipart/note_06.svg"  
9: "/DTDWeb/images/markup/clipart/note_06_bg100.svg"  
10: "/DTDWeb/images/markup/clipart/note_07.svg"  
11: "/DTDWeb/images/markup/clipart/note_07_bg100.svg"  
length: 12
```

DTDPlayer.ShowToastMessage(message, showSeconds)

토스트 메시지 표시 함수.

파라미터	타입	설명
message	string	표시 메시지
showSeconds	number	표시 초(기본값: 2.5)

사용 예)

```
player.ShowToastMessage('DTDWeb 토스트 메시지 기능입니다.', 2.5);
```



DTDPlayer.HideToastMessage()

토스트 메시지를 즉시 숨기는 함수.

사용 예)

```
player.HideToastMessage();
```

마크업

마크업 관련 기능으로 모든 함수에 SetFunctionMode(DTDPlayer.FunctionMode.MARKUP)으로 현재 기능 모드를 마크업으로 변경함. 히스토리 관련 이전 작업, 다음 작업등은 내부적으로 관리됨.

마크업 모드에서는 카메라 제어와 DTDWeb 내 GUI 인터랙션이 불가능함.

DTDPlayer.StartMarkup()

마크업 모드 시작 함수로 GUI(예: Toolbar)에 바인딩 될 경우 필요. 해당 함수 최초 실행 시, 그리기 기본('#FF0000' 색상, 2pt 선 두께)로 마크업 모드가 시작되며 이후 가장 마지막으로 변경된 기능(그리기, 글자 입력, 클립아트와 옵션)으로 시작됨.

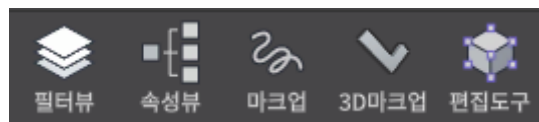


사용 예)

```
player.StartMarkup();
```

DTDPlayer.StopMarkup()

마크업 모드 종료 함수로 DTDPlayer 의 FunctionMode 가 NONE 으로 변경됨. GUI(예: Toolbar)에 바인딩 될 경우 필요.



사용 예)

```
player.StopMarkup();
```

DTDPlayer.OnAddMarkupSuccess(markupType) Callback

마크업 모드와 기능 Panel 연동에 필요한 Callback 함수로 그리기, 글자 입력, 클립아트가 추가됐을 때 정의된 함수가 호출됨. Callback 파라미터인 markupType 은 아래 표와 같은 값을 가짐.

markupType 값(string)	설명
Line	그리기 마크업이 추가된 경우

Text	글자 입력 마크업이 추가된 경우
Clipart_\$Index	클립아트 마크업이 추가된 경우('Clipart_' + index)

정의 예)

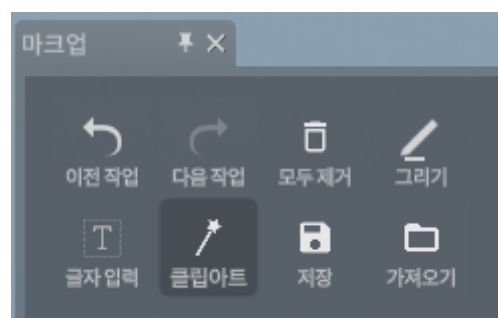
```
player.OnAddMarkupSuccess = (markupType) => {
  if (markupType === 'Line') {
  }
  else if (markupType === 'Text') {
  }
  else if (markupType.includes('Clipart_') {
    // TODO: 기능 Panel 버튼 해제
  }
};
```

그리기, 글자 입력 마크업이 추가된 경우와 다르게 클립아트는 기능 Panel 에서 버튼 해제 필요.



DTDPlayer.OnAfterMarkupAction(hasPrevious, hasNext) Callback

마크업 모드는 히스토리, 이전 작업, 다음 작업을 가짐. 따라서 마크업이 추가/제거 된 경우 해당 Callback 함수가 호출 되고 결과에 따른 기능 Panel 의 이전 작업, 다음 작업 버튼 활성화 및 비활성화 작업이 요구됨.



파라미터	타입	설명
hasPrevious	boolean	이전 작업 존재 여부
hasNext	Boolean	다음 작업 존재 여부

정의 예)

```
player.OnAfterMarkupAction = (hasPrevious, hasNext) => {
  if (hasPrevious) {
```

```

        // TODO: 기능 Panel 이전 작업 버튼 활성화
    }
    else {
        // TODO: 기능 Panel 이전 작업 버튼 비활성화
    }

    if (hasNext) {
        // TODO: 기능 Panel 다음 작업 버튼 활성화
    }
    else {
        // TODO: 기능 Panel 다음 작업 버튼 비활성화
    }
};

```

DTDPlayer.UndoMarkup()

마크업 모드 이전 작업으로 주로 추가된 마크업을 제거할 경우 사용됨.

사용 예)

```
player.UndoMarkup();
```

DTDPlayer.RedoMarkup()

마크업 모드 다음 작업으로 주로 제거된 마크업을 다시 추가할 때 사용됨.

사용 예)

```
player.RedoMarkup();
```

DTDPlayer.ClearMarkups()

모든 마크업을 제거하는 함수로 히스토리도 초기화됨.

사용 예)

```
player.ClearAllMarkups();
```

DTDPlayer.SetMarkupPenColor(hexColor)

마크업 그리기의 선 색상 설정. 설정 시 자동으로 그리기 추가 기능이 활성화됨.

파라미터	타입	설명
hexColor	string	RGB 색상 HEX(기본값: '#FF0000')

사용 예)

```
player.SetMarkupPenColor('#FF0000'); // 선 색상 변경(빨강색)
```

DTDPlayer.SetMarkupPenThickness(thickness)

마크업 그리기의 선 두께 설정. 설정 시 자동으로 그리기 추가 기능이 활성화됨.

파라미터	타입	설명
thickness	number	선 두께 Pixel (기본값: 2)

사용 예)

```
player.SetMarkupPenThickness(5); // 선 두께 변경(5px)
```

DTDPlayer.SetMarkupFontColor(hexColor)

마크업 글자 입력 글자 색상 설정. 마크업 텍스트가 선택된 경우 선택 마크업 텍스트 색상 변경. 설정 시 자동으로 글자 입력 추가 기능이 활성화됨.

파라미터	타입	설명
hexColor	string	RGB 색상 HEX(기본값: '#FFFFFF');

사용 예)

```
player.SetMarkupFontColor('#FF0000'); // 글자 색상 변경(빨강색)
```

DTDPlayer.SetMarkupFontSize(fontSize)

마크업 글자 입력 글자 크기 설정. 마크업 텍스트가 선택된 경우 선택 마크업 텍스트 크기 변경. 설정 시 자동으로 글자 입력 추가 기능이 활성화됨.

파라미터	타입	설명
fontSize	number	글자 크기

사용 예)

```
player.SetMarkupFontSize(15); // 글자 크기 변경(15px)
```

DTDPlayers.SetMarkupClipartIndex(clipartIndex)

마크업 클립아트 Index 설정(GUI 항목 DTDPlayer.GetMarkupClipartImageUrls() 참조). 설정 시 자동으로 클립아트 입력 추가 기능이 활성화됨. 따라서 선택해제 된 경우 해당 함수에 -1 을 넣어 전달 필요함.

파라미터	타입	설명
clipartIndex	number	클립아트 Index

사용 예)

```
player.SetMarkupClipartIndex(3); // 3 번 클립아트로 현재 클립아트 Index 변경  
player.SetMarkupClipartIndex(-1); // 기능 Panel 클립아트 선택 해제 시 호출
```

DTDPlayer.GetMarkupCurrentOptions()

마크업의 현재 옵션들을 반환함.

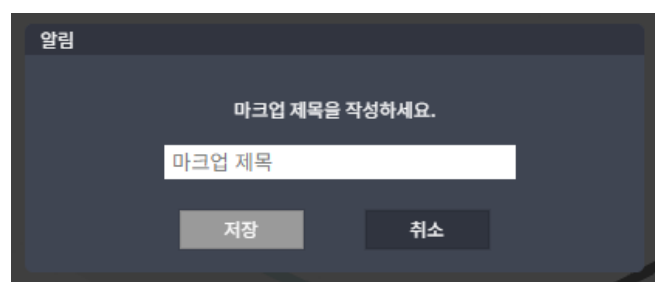
사용 예)

```
const currentMarkupOptions = player.GetMarkupCurrentOptions();
```

```
▼ {penColor: '#FF0000', perColorPreset:  
  clipartIndex: -1  
  fontColor: '#FFFFFF'  
  fontColorPreset Index: 9  
  fontSize: 14  
  penColor: '#FF0000'  
  penThickness: 2  
  perColorPreset Index: 0  
  ▶ [[Prototype]]: Object
```

DTDPlayers.SaveMarkup(onSuccessCallback)

마크업 저장 함수로 사용자가 마크업 제목 입력 후 저장에 성공하면 onSuccessCallback 으로 마크업 제목과 Base64 인코딩 된 스크린샷 이미지의 string 을 넘겨줌.



파라미터	타입	설명
base64String	string	마크업 Base64 string

사용 예(결과화면 위와 동일))

```
const base64String = ''; // 저장된 Base64 String 변수로
player.LoadMarkupByBase64String(base64String);
```

패스 트래킹

연결 정보가 있는 모델에 대해 카메라가 경로를 따라가는 기능으로 컨텍스트 메뉴로 진입하며 Pause, Resume, Stop 기능을 함수로 제공함(카메라 애니메이션 불가능).



DTDPlayer.PausePathTracking()

패스 트래킹 애니메이션을 일시 정지하는 함수로 패스 트래킹 모드 시, 키보드 P 키로 사용 가능.

사용 예)

```
player.PausePathTracking();
```

DTDPlayer.ResumePathTracking()

패스 트래킹 애니메이션의 일시정지를 재개하는 함수로 패스 트래킹 모드 시, 키보드 R 키로 사용 가능.

사용 예)

```
player.ResumePathTracking();
```

DTDPlayer.StopPathTracking()

패스 트래킹 모드가 종료되며 FunctionMode 가 NONE 으로 설정됨.

사용 예)

```
player.StopPathTracking();
```

간섭검사

모델(부재)간에 간섭 유무를 판정하는 기능.

DTDPlayer.OnInterferenceCheckEnd(successResult) Callback

간섭검사 후 호출되는 Callback 함수로 successResult 는 아래 표와 같은 Key-Value 배열을 가짐.

Key(타입)	Value
intersectCenter(Object)	간섭지점 3D 포인트(x, y, z)
intersectRadius(number)	간섭지점으로부터 반경
mesh1FileName, mesh2FileName	간섭 부재 1,2 의 각 파일명
mesh1Id, mesh2Id	간섭 부재 1,2 의 각 아이디
pointIndex	결과 Sphere 의 Index

정의 예)

```
player.OnInterferenceCheckEnd = (successResult) => {  
    console.log(`OnInterferenceCheckEnd: ${successResult}`);  
};
```

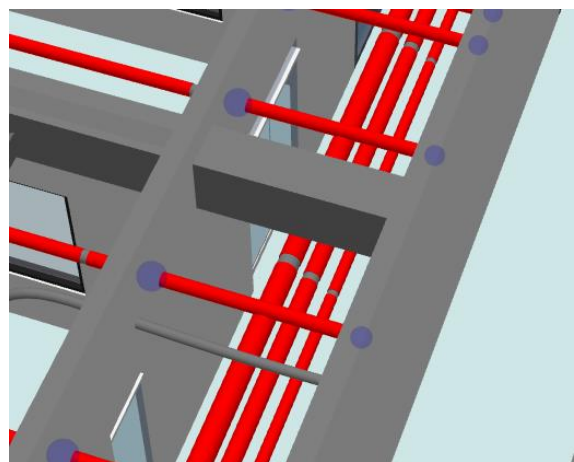
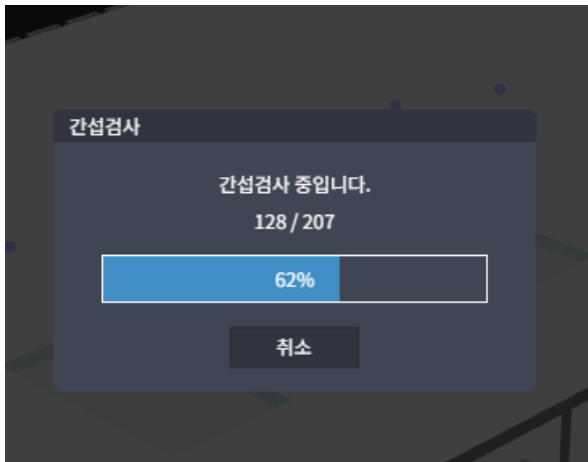
```
(27) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...},  
  {...}, {...}, {...}, {...}, {...}]  
  0:  
    ▶ intersectCenter: {x: -4.8898234367370605, y: 26.933199882507324, z: 26.48177146911621}  
    ▶ intersectRadius: 0.3190653098523731  
    ▶ mesh1FileName: "GASAN_7F.dtdx"  
    ▶ mesh1Id: "967611"  
    ▶ mesh2FileName: "GASAN_7F.dtdx"  
    ▶ mesh2Id: "955496"  
    ▶ pointIndex: 0  
    ▶ [[Prototype]]: Object  
  1:  
    ▶ intersectCenter: {x: -3.972852945327759, y: 27.038146018981934, z: 4.617244243621826}  
    ▶ intersectRadius: 0.199830009912283  
    ▶ mesh1FileName: "GASAN_7F.dtdx"  
    ▶ mesh1Id: "1013306"  
    ▶ mesh2FileName: "GASAN_7F.dtdx"  
    ▶ mesh2Id: "956084"  
    ▶ pointIndex: 1  
    ▶ [[Prototype]]: Object  
  2: {pointIndex: 2, mesh1Id: '1013306', mesh2Id: '1098105', mesh1FileName: 'GASAN_7F.dtdx', m  
  3: {pointIndex: 3, mesh1Id: '1013332', mesh2Id: '956084', mesh1FileName: 'GASAN_7F.dtdx', me  
  4: {pointIndex: 4, mesh1Id: '1013332', mesh2Id: '1098105', mesh1FileName: 'GASAN_7F.dtdx', m  
  5: {pointIndex: 5, mesh1Id: '1013332', mesh2Id: '956084', mesh1FileName: 'GASAN_7F.dtdx', m
```

DTDPlayer.StartInterferenceCheck(keyValueJson1, keyValueJson2)

간섭검사를 시작하는 함수로 두 비교 대상의 Key-Value 파라미터가 필요함. 간섭검사 후 간섭 부분에 반투명(Blue) Sphere 로 표시되며 앞서 기술한 DTDPlayer.OnInterferenceCheckEnd Callback 함수로 간섭결과 전달됨.

사용 예)

```
player.StartInterferenceCheck([[{key: '카테고리', value: '배관'}]],  
[[{key: '카테고리', value: '벽'}]]); // 카테고리 배관, 벽 간 간섭검사
```



DTDPlayer.StopInterferenceCheck()

간섭검사 모드가 종료되며 FunctionMode 가 NONE 으로 설정됨(결과 삭제).

사용 예)

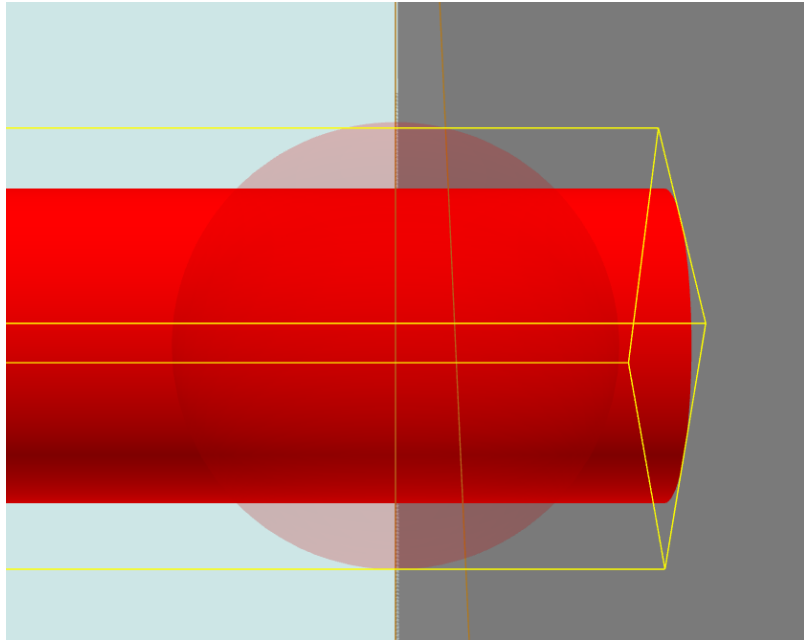
```
player.StopInterferenceCheck();
```

DTDPlayer.MoveCameraToInterferencePoint(pointIndex)

간섭검사 후 받은 DTDPlayer.OnInterferenceCheckEnd Callback 함수의 successResult 값 중 pointIndex 를 파라미터로 사용하고, 호출 시 두 간섭 부재가 선택 및 간섭지점으로 카메라 이동됨.

사용 예)

```
player.MoveCameraToInterferencePoint(0);
```



북마크

사용자가 바라보는 시점(카메라)을 저장해 애니메이션 재생하는 기능으로 DTDWeb 에서 북마크는 그룹과 북마크(실제 카메라 데이터)로 구성됨. DTDWeb 은 북마크 그룹과 북마크를 관리하는 기능을 제공하며 각각 Sequence ID 로 관리됨. 또한 애니메이션을 재생/정지하는 기능을 제공함. 애니메이션 설정으로 재생속도와 시간을 가지며 전역 설정과 각 북마크 설정이 가능함. 북마크에 설정된 값이 0 이면 전역 설정을 사용해 재생되며 값이 있는 경우 북마크 설정을 사용해 재생됨.

DTDPlayer.AddBookmarkGroup(onSuccessCallback)

북마크 그룹 추가 함수로 호출 시 북마크 그룹명을 입력할 수 있는 팝업이 DTDWeb 에 활성화됨. 파라미터로 Callback 함수를 가지고, 북마크 그룹 추가 성공 시 호출됨.

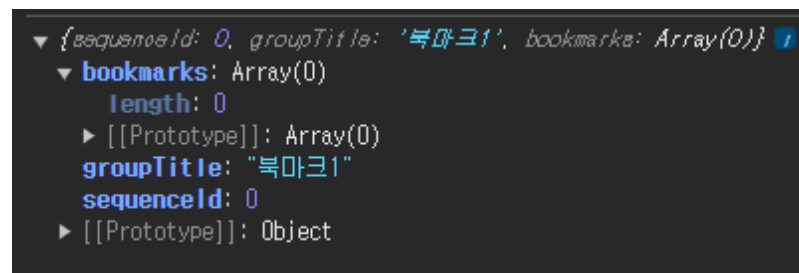
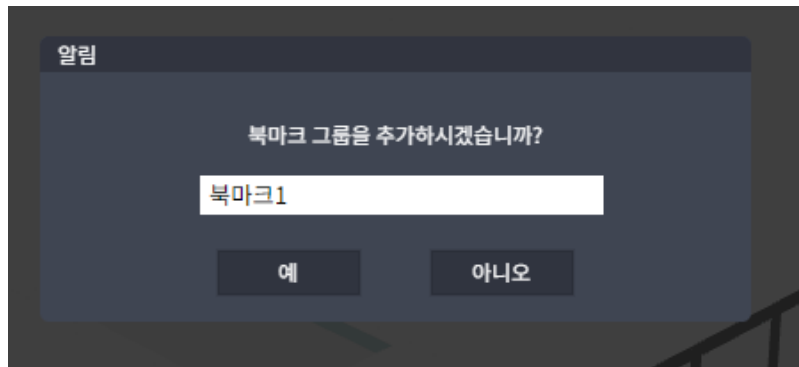
파라미터	타입	설명
onSuccessCallback	function	북마크 그룹 추가 성공 시 호출

파라미터인 onSuccessCallback 함수는 생성된 북마크 그룹을 파라미터(Key-Value)로 가짐.

Key(타입)	Value
bookmarks(Object[])	북마크 그룹에 속한 북마크 데이터
groupTitle(string)	북마크 그룹명
sequenceId	북마크 그룹 Sequence ID

사용 예)

```
player.AddBookmarkGroup((bookmarkGroup) => {
    console.log(bookmarkGroup);
});
```



DTDPlayer.AddBookmark(groupSequenceId, onSuccessCallback)

북마크 추가 함수로 호출 시 북마크가 추가됨. 파라미터로 북마크가 추가될 그룹의 Sequence ID 인 groupSequenceId 와 추가 성공 시 호출되는 Callback 함수가 요구됨.

파라미터	타입	설명
groupSequenceId	number	북마크가 추가될 북마크 그룹 Sequence ID
onSuccessCallback	function	북마크 추가 성공 시 호출

파라미터인 onSuccessCallback 함수는 생성된 북마크를 파라미터(Key-Value)로 가짐.

Key(타입)	Value
animateDelay(number)	애니메이션 재생 간격(초)
animateSpeed(number)	애니메이션 재생 속도(초), 클수록 느리게 재생
cameraPosition(Object)	카메라 위치 정보(x, y, z)
cameraRotation(Object)	카메라 회전 정보(x, y, z, w)
cameraTarget(Object)	카메라 타겟 정보(x, y, z)
sequenceId	생성된 북마크 Sequence ID

사용 예)

```
player.AddBookmark((bookmark) => {
    console.log(bookmark);
});
```

```

▼ {sequenceId: 0, cameraTarget: {...}, cameraPosition: {...}, cameraRotation: {...}, animateSpeed: 0, ...}
  animateDelay: 0
  animateSpeed: 0
  ▶ cameraPosition: {x: 37.0463757335729, y: 52.40576047511398, z: -23.180984479147867}
  ▶ cameraRotation: {x: 0.19996412247074147, y: -0.37361230677718404, z: 0.08282785278038998, w: 0.9019798932371963}
  ▶ cameraTarget: {x: -3.0558533668518066, y: 25.960000038146973, z: 16.921244621276855}
  sequenceId: 0
  ▶ [[Prototype]]: Object

```

DTDPlayer.RemoveBookmarkGroup(groupSequenceId, onSuccessCallback)

북마크 그룹 삭제 함수로 삭제 시 그룹에 종속된 북마크들 전체 삭제됨. 삭제할 그룹의 Sequence ID 와 삭제 성공 시 호출될 Callback 함수 요구됨. 북마크 그룹 삭제 시 Sequence ID 가 재설정됨.

파라미터	타입	설명
groupSequenceId	number	삭제할 북마크 그룹 Sequence ID
onSuccessCallback	function	북마크 그룹 삭제 시 호출

파라미터인 onSuccessCallback 함수는 Sequence ID 가 재설정된 모든 북마크 그룹들을 Key-Value 배열로 가짐(DTDPlayer.AddBookmarkGroup 참조).

사용 예)

```

player.RemoveBookmarkGroup(1, (bookmarkGroups) => {
  console.log(bookmarkGroups);
});

```

```

▼ (2) [{...}, {...}]
  ▼ 0:
    ▶ bookmarks: [{...}]
    ▶ groupTitle: "북마크1"
    ▶ sequenceId: 0
    ▶ [[Prototype]]: Object
  ▼ 1:
    ▶ bookmarks: []
    ▶ groupTitle: "북마크3"
    ▶ sequenceId: 1
    ▶ [[Prototype]]: Object
    ▶ length: 2
    ▶ [[Prototype]]: Array(0)

```

북마크 1(Sequence ID: 0), 북마크 2(1), 북마크 3(2)에서 북마크 2(1)를 삭제한 경우로 북마크 3의 Sequence ID 가 2 에서 1 로 재설정된 결과임.

DTDPlayer.RemoveBookmark(groupSequenceId, bookmarkSequenceId, onSuccessCallback)

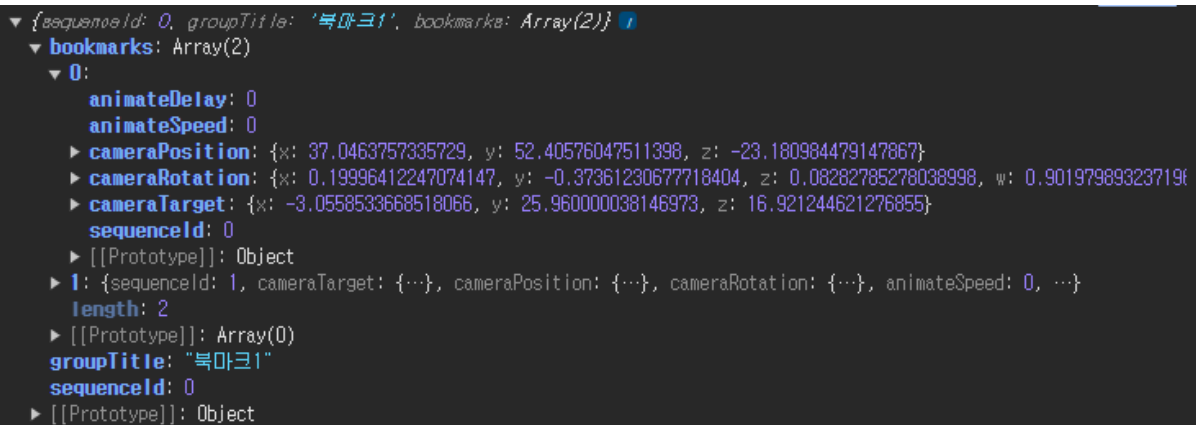
북마크 삭제 함수로 삭제 시 북마크 그룹과 동일하게 Sequence ID 가 재설정됨.

파라미터	타입	설명
groupSequenceId	number	삭제할 북마크가 종속된 그룹 Sequence ID
bookmarkSequenceId	number	삭제할 북마크 Sequence ID
onSuccessCallback	function	북마크 삭제 시 호출

파라미터인 onSuccessCallback 함수는 삭제 후 Sequence ID 가 재설정된 북마크 그룹을 Key-Value 로 가짐(DTDPlayer.AddBookmark 참조).

사용 예)

```
player.RemoveBookmark(0, 1, (bookmarkGroup) => {
    console.log(bookmarkGroup);
});
```



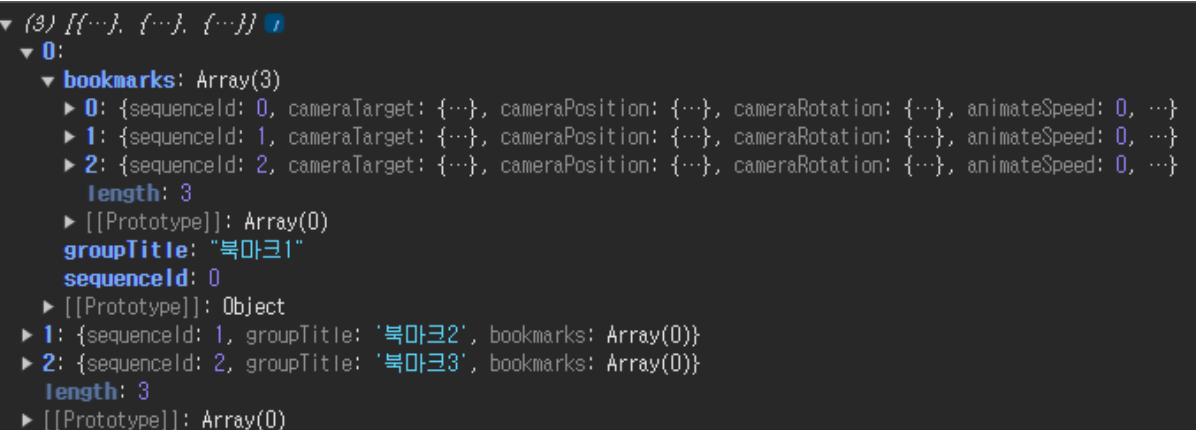
```
{sequenceId: 0, groupTitle: '북마크1', bookmarks: Array(2)}
  bookmarks: Array(2)
    0: {
      animateDelay: 0
      animateSpeed: 0
      cameraPosition: {x: 37.0463757335729, y: 52.40576047511398, z: -23.180984479147867}
      cameraRotation: {x: 0.19996412247074147, y: -0.37361230677718404, z: 0.08282785278038998, w: 0.901979893237196}
      cameraTarget: {x: -3.0558533668518066, y: 25.960000038146973, z: 16.921244621276855}
      sequenceId: 0
      [[Prototype]]: Object
    }
    1: {sequenceId: 1, cameraTarget: {...}, cameraPosition: {...}, cameraRotation: {...}, animateSpeed: 0, ...}
      length: 2
      [[Prototype]]: Array(0)
    groupTitle: "북마크1"
    sequenceId: 0
    [[Prototype]]: Object
```

DTDPlayer.GetAllBookmarks()

모든 북마크에 대한 정보를 가지고 오는 함수로 북마크 그룹 Key-Value 배열을 가짐(DTDPlayer.AddBookmarkGroup, DTDPlayer.AddBookmark 참조).

사용 예)

```
const bookmarkGroups = player.GetAllBookmarks();
console.log(bookmarkGroup);
```



```
(3) [{...}, {...}, {...}]
  0: {
    bookmarks: Array(3)
      0: {sequenceId: 0, cameraTarget: {...}, cameraPosition: {...}, cameraRotation: {...}, animateSpeed: 0, ...}
      1: {sequenceId: 1, cameraTarget: {...}, cameraPosition: {...}, cameraRotation: {...}, animateSpeed: 0, ...}
      2: {sequenceId: 2, cameraTarget: {...}, cameraPosition: {...}, cameraRotation: {...}, animateSpeed: 0, ...}
      length: 3
      [[Prototype]]: Array(0)
    groupTitle: "북마크1"
    sequenceId: 0
    [[Prototype]]: Object
  }
  1: {sequenceId: 1, groupTitle: '북마크2', bookmarks: Array(0)}
  2: {sequenceId: 2, groupTitle: '북마크3', bookmarks: Array(0)}
    length: 3
    [[Prototype]]: Array(0)
```

DTDPlayer.SetBookmarkGlobalAnimateSpeed(animateSpeed)

북마크 전역 재생속도(초)를 설정하는 함수로 재생되는 북마크의 재생속도가 0 일 때 사용됨.

사용 예)

```
player.SetBookmarkGlobalAnimateSpeed(1); // 전역 재생속도를 1 초로 설정
```

DTDPlayer.SetBookmarkGlobalAnimateDelay(animateDelay)

북마크 전역 재생간격(초)을 설정하는 함수로 재생되는 북마크의 재생속도가 0 일 때 사용됨.

사용 예)

```
player.SetBookmarkGlobalAnimateDelay(1); // 전역 재생간격을 1 초로 설정
```

DTDPlayer.SetBookmarkAnimateSpeed(groupSequenceId, bookmarkSequenceId, animateSpeed)

북마크의 재생속도(초)를 별도 설정하는 함수로 0 이 아닐 경우 전역 재생속도보다 우선 적용됨.
그룹 Sequence ID 와 그룹 내 북마크 Sequence ID 요구됨.

파라미터	타입	설명
groupSequenceId	number	북마크가 그룹 Sequence ID
bookmarkSequenceId	number	북마크 Sequence ID

사용 예)

```
player.SetBookmarkAnimateSpeed(1); // 북마크 재생속도를 1 초로 설정
```

DTDPlayer.SetBookmarkAnimateDelay(groupSequenceId, bookmarkSequenceId, animateDelay)

북마크의 재생간격(초)을 별도 설정하는 함수로 0 이 아닐 경우 전역 재생간격보다 우선 적용됨.
그룹 Sequence ID 와 그룹 내 북마크 Sequence ID 요구됨.

파라미터	타입	설명
groupSequenceId	number	북마크가 그룹 Sequence ID
bookmarkSequenceId	number	북마크 Sequence ID

사용 예)

```
player.SetBookmarkAnimateDelay(1); // 북마크 재생간격을 1 초로 설정
```

```
DTDPlayer.PlayBookmarkAnimation(groupSequenceId, isReverse,  
onStateChangedCallback)
```

북마크 애니메이션을 재생하는 함수로 isReverse 파라미터로 역재생 가능함. 그룹 내 북마크 애니메이션 재생이 완료될 때마다 onStateChangedCallback 파라미터 함수 호출.

파라미터	타입	설명
groupSequenceId	number	북마크가 그룹 Sequence ID
isReverse	boolean	역재생 여부(true: 역재생)
onStateChangedCallback	function	북마크 애니메이션 상태 변화 Callback 함수

onStateChangedCallback 은 아래 표와 같은 파라미터를 가짐.

파라미터	타입	설명
bookmarkSequenceId	number	애니메이션 상태 변경된 북마크 Sequence ID
state	string	변경된 애니메이션 상태

사용 예)

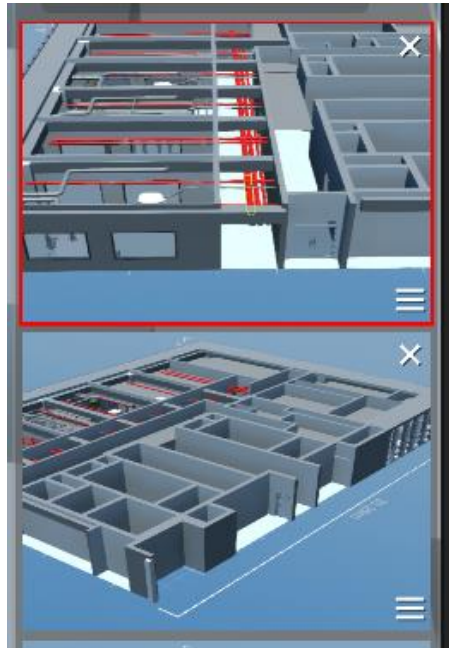
```
player.PlayBookmarkAnimation(0, false, (bookmarkSequenceId, state) => {  
    console.log(bookmarkSequenceId, state);  
}) // 정방향 재생
```

사용 예)

```
player.PlayBookmarkAnimation(0, true, (bookmarkSequenceId, state) => {  
    console.log(bookmarkSequenceId, state);  
}) // 역방향 재생
```

0 'Start'	3 'Start'
0 'End'	3 'End'
1 'Start'	2 'Start'
1 'End'	2 'End'
2 'Start'	1 'Start'
2 'End'	1 'End'
3 'Start'	0 'Start'
3 'End'	0 'End'

위 결과화면은 정방향(좌), 역방향(우) 재생을 나타냄. 따라서 기능 Panel 사용 시, 해당 북마크의 하이라이트 기능 표시/미표시 예상됨.



DTDPlayer.StopBookmarkAnimation()

북마크 애니메이션 정지 함수.

사용 예)

```
player.StopBookmarkAnimation();
```

DTDPlayer.SaveBookmark()

북마크 저장 함수로 모든 북마크들을 JSON.stringify 를 거쳐 반환함.

사용 예)

```
player.SaveBookmark();
```

```
{
  "sequenceId": 0,
  "groupTitle": "북마크1",
  "bookmarks": [
    {
      "sequenceId": 0,
      "cameraTarget": {
        "x": 2.8442057506782277,
        "y": 28.233206415854802,
        "z": 9.337759008285797
      },
      "cameraPosition": {
        "x": 15.694413163599346,
        "y": 55.27442918012815,
        "z": -39.37930269703091
      },
      "cameraRotation": {
        "x": 0.2417845195522811,
        "y": -0.12471153934379832,
        "z": 0.03135185591567208,
        "w": 0.9617714558343241
      },
      "animateSpeed": 0,
      "animateDelay": 0
    },
    {
      "sequenceId": 1,
      "cameraTarget": {
        "x": 1.3947671650314057,
        "y": 25.406915361422257,
        "z": 14.395148720230093
      },
      "cameraPosition": {
        "x": 11.543982165939326,
        "y": 66.3078660395207,
        "z": -33.71768613251254
      },
      "cameraRotation": {
        "x": 0.33816404319382865,
        "y": -0.09758043114109592,
        "z": 0.035279000632900216,
        "w": 0.9353494137133714
      },
      "animateSpeed": 0,
      "animateDelay": 0
    },
    {
      "sequenceId": 2,
      "cameraTarget": {
        "x": 3.8986419991802013,
        "y": 26.94561376869304,
        "z": 9.986034276080066
      },
      "cameraPosition": {
        "x": 47.15053901679679,
        "y": 55.47754546514689,
        "z": -22.836149884941886
      },
      "cameraRotation": {
        "x": 0.2145736841345164,
        "y": -0.4317374251939069,
        "z": 0.10653058993787391,
        "w": 0.8696045997988191
      },
      "animateSpeed": 0,
      "animateDelay": 0
    },
    {
      "sequenceId": 3,
      "cameraTarget": {
        "x": 7.6252802214538296,
        "y": 25.14645665500519,
        "z": 10.064441980790185
      },
      "cameraPosition": {
        "x": 62.76886590191174,
        "y": 58.775912850268526,
        "z": -0.20650921331368544
      },
      "cameraRotation": {
        "x": 0.20518055168119828,
        "y": -0.6159363886541942,
        "z": 0.17049269672288062,
        "w": 0.7412526592629899
      },
      "animateSpeed": 0,
      "animateDelay": 0
    }
  ]
}
```

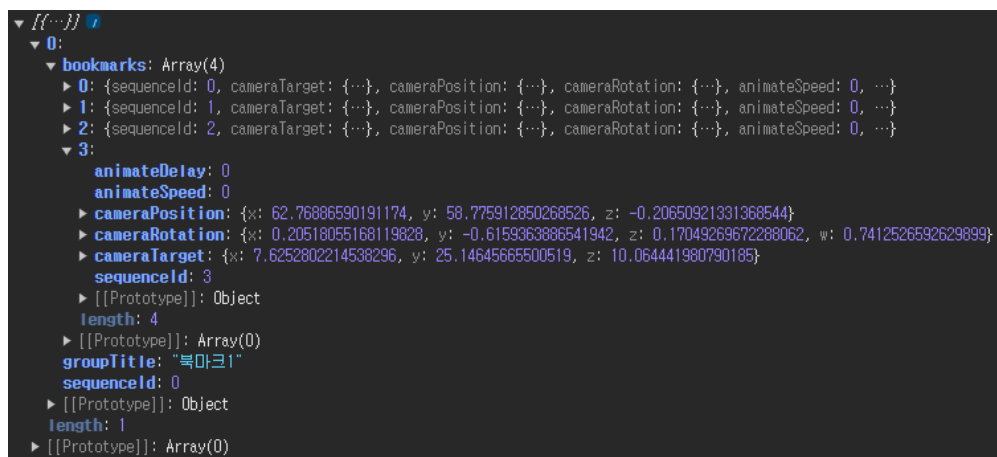
DTDPlayer.LoadBookmark(bookmarks)

북마크 로드 함수로 DTDPlayer.SaveBookmark()를 통해서 반환된 값을 파라미터로 사용함.
성공적으로 로드 시, 로드된 모든 북마크를 반환함. 기존 데이터가 덮어쓰기 되기 때문에 기능
Panel 사용 시 초기화 요구됨.

파라미터	타입	설명
bookmarks	string	저장된 북마크 JSON string

사용 예)

```
const savedBookmark = // 저장된 북마크 값  
player.LoadBookmark(savedBookmark);
```



```
▼ [{"...}]  
  ▼ 0:  
    ▼ bookmarks: Array(4)  
      ► 0: {sequenceId: 0, cameraTarget: {...}, cameraPosition: {...}, cameraRotation: {...}, animateSpeed: 0, ...}  
      ► 1: {sequenceId: 1, cameraTarget: {...}, cameraPosition: {...}, cameraRotation: {...}, animateSpeed: 0, ...}  
      ► 2: {sequenceId: 2, cameraTarget: {...}, cameraPosition: {...}, cameraRotation: {...}, animateSpeed: 0, ...}  
      ▼ 3:  
        animateDelay: 0  
        animateSpeed: 0  
        ► cameraPosition: {x: 62.76886590191174, y: 58.775912850268526, z: -0.20650921331368544}  
        ► cameraRotation: {x: 0.20518055168119828, y: -0.6159363886541942, z: 0.17049269672288062, w: 0.7412526592629899}  
        ► cameraTarget: {x: 7.6252802214538296, y: 25.14645665500519, z: 10.064441980790185}  
        sequenceId: 3  
        ► [[Prototype]]: Object  
        length: 4  
        ► [[Prototype]]: Array(0)  
        groupTitle: "북마크1"  
        sequenceId: 0  
        ► [[Prototype]]: Object  
        length: 1  
        ► [[Prototype]]: Array(0)
```