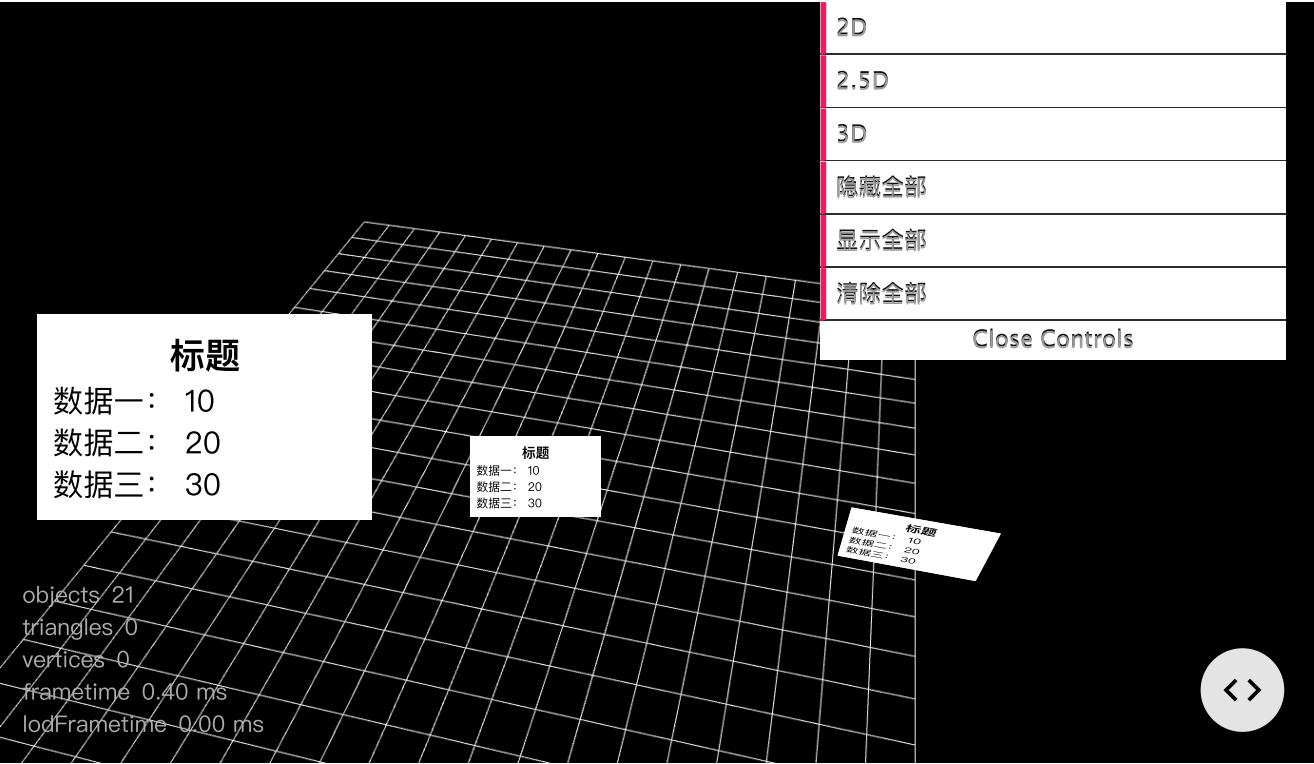


PoiNode 对象

createPoiNode

创建 `poiNode` 对象。可以插入任何 `DOM` 元素，并且保留其交互事件。

样例：



定义：

```
interface PoiNodeInfo extends BaseObject3DInfo {
  type: PoiNodeType;
  element: HTMLElement;
}

function createPoiNode(poiNodeInfo: PoiNodeInfo): PoiNode;
```

用法:

```
const el = document.createElement('div');
el.innerHTML = '一段文字';

ssp.createPoiNode(
  // poiNodeInfo
  {
    type: '3d',
    element: el,
    id: 'xx',
    name: 'xx',
    level: {
      max: 1000,
      min: null,
    },
    position: { x: 0, y: 0, z: 0 },
    rotation: { x: 0, y: 0, z: 0 },
    scale: { x: 2, y: 2, z: 2 },
    userData: {},
  }
);
```

参数:

poiNodeInfo

- 描述: 实例 `PoiNode` 对象所需信息
- 类型: PoiNodeInfo
- 必填: ✓

PoiNodeInfo

属性	描述	类型	必填	默认值
type	展示模式	PoiNodeType	✓	
element	DOM 元素	HTMLElement	✓	
id	唯一ID	string number	✓	

属性	描述	类型	必填	默认值
name	名称	string	X	
level	显示层级范围	Level	X	{ max: null, min: null }
position	位置坐标	Position	X	{ x: 0, y: 0, z: 0 }
rotation	旋转弧度	Rotation	X	{ x: 0, y: 0, z: 0 }
scale	缩放比例	Scale	X	{ x: 1, y: 1, z: 1 }
userData	用户数据	any	X	{}

getNodeById

通过 `id` 查找

定义:

```
function getNodeById(id: PoiNodeInfo['id']): PoiNode | null;
```

ts

用法:

```
const poiNode = ssp.getNodeById('xxx');
```

js

getNodeByName

通过 `name` 查找

定义:

```
function getNodeByName(name: string): PoiNode[];
```

ts

用法:

```
const poiNodeList = ssp.getPoiNodeByName('xxx');
```

js

getAllPoiNode

获取所有 PoiNode 对象

定义:

```
function getAllPoiNode(): PoiNode[];
```

ts

用法:

```
const allPoiNodeList = ssp.getAllPoiNode();
```

js

getPoiNodeByUserDataProperty

通过 userData 属性查找

定义:

```
function getPoiNodeByUserDataProperty(  
  propNameOrFindFunc: string | UserDataPropertyFindFunc,  
  value?: any  
): PoiNode[];
```

ts

用法:

```
const poiNodeList = ssp.getPoiNodeByUserDataProperty('propKey', 'propVal')  
// or  
const poiNodeList = ssp.getPoiNodeByUserDataProperty(item => item['itemPropKey'] =
```

js

参数:

propNameOrFindFunc

- 描述: `userData` 内属性名 或 `find` 函数
- 类型: `string` | `function`
- 必填: ✓

propValue

- 描述: `userData` 内属性值。
- 类型: `any`
- 必填: ✗

find 函数使用场景

```
poiNode.userData = {  
  people: {  
    name: 'xiaoming',  
    age: 18,  
  },  
};  
const poiNodeList = ssp.getPoiNodeByUserDataProperty(  
  (userData) => userData?.people?.name === 'xiaoming'  
);
```

js

removePoiNodeById

通过 `id` 移除

定义:

```
function removePoiNodeById(id: PoiNodeInfo['id']): boolean;
```

ts

用法:

```
ssp.removePoiNodeById('xxx');
```

js

createPoiNodeToGroup

创建 `poiNode` 到一个组内。

定义:

```
function createPoiNodeToGroup(  
  groupInfo: GroupInfo,  
  poiNodeInfoList: PoiNodeInfo[]  
): Group;
```

ts

用法:

```
ssp.createPoiNodeToGroup(  
  // groupInfo  
  {  
    id: 'firstPoiNodeGroup',  
    name: 'name_firstPoiNodeGroup',  
    // ...  
  },  
  // poiNodeInfoList  
  [poiNodeInfo1, poiNodeInfo2, poiNodeInfo3]  
);
```

js

参数

groupInfo

- 描述: 实例组对象所需信息
- 类型: **GroupInfo**

- 必填: ✓

poiNodeInfoList

- 描述: poiNodeInfo 集合
- 类型: poiNodeinfo[]
- 必填: ✓

createGroupForPoiNode

为 poiNode 提前创建一个空组。

使用场景

与 createPoiNodeToGroup 不同，有些时候可能你还没有具体的 poiNodeInfo 数据，但你想提前创建一个批量管理的空组，当有数据时再使用 addPoiNodeForGroup 插入。

定义：

```
function createGroupForPoiNode(groupInfo: GroupInfo): Group;
```

ts

用法：

```
ssp.createGroupForPoiNode({  
  id: 'firstPoiNodeGroup',  
  name: 'name_firstPoiNodeGroup',  
  // ...  
});
```

js

参数

groupInfo

- 描述: 实例组对象所需信息
- 类型: GroupInfo
- 必填: ✓

addPoiNodeForGroup

向一个已经存在的组内添加 `poiNode` 对象。

定义：

```
function addPoiNodeForGroup(  
  groupId: GroupInfo['id'],  
  poiNodeInfoList: PoiNodeInfo[]  
): Group | null;
```

ts

用法：

```
ssp.addPoiNodeForGroup(  
  // groupId  
  'firstPoiNodeGroup',  
  // poiNodeInfoList  
  [poiNodeInfo4, poiNodeInfo5],  
  // onProgress  
  (progress) => console.log('进度信息: ', progress)  
);
```

js

参数

groupId

- 描述: 组 `id`
- 类型: `GroupInfo['id']`
- 必填: ✓

poiNodeInfoList

- 描述: `poiNodeInfo` 集合
- 类型: `poiNodeinfo[]`
- 必填: ✓

getPoiNodeGroupById

通过 `id` 查找 `poiNode` 组

定义:

```
function getPoiNodeGroupById(id: GroupInfo['id']): Group | null;
```

ts

用法:

```
const group = ssp.getPoiNodeGroupById('firstPoiNodeGroup');
```

js

getPoiNodeGroupByName

通过 `name` 查找 `poiNode` 组

定义:

```
function getPoiNodeGroupByName(name: string): Group[];
```

ts

用法:

```
const groupList = ssp.getPoiNodeGroupByName('name_firstPoiNodeGroup');
```

js

removePoiNodeGroupById

通过 `id` 移除 `poiNode` 组

定义:

```
function removePoiNodeGroupById(id: GroupInfo['id']): boolean;
```

ts

用法:

```
const isRemoveSuccess = ssp.removePoiNodeGroupById('firstPoiNodeGroup');
```

js

clearPoiNode

清除当前场景内所有 `poiNode` 对象。

定义:

```
function clearPoiNode(): void;
```

ts

用法:

```
ssp.clearPoiNode();
```

js

showAllPoiNode

显示当前场景内所有 `poiNode` 对象。

定义:

```
function showAllPoiNode(): void;
```

ts

用法:

```
ssp.showAllPoiNode();
```

js

hideAllPoiNode

隐藏当前场景内所有 `poiNode` 对象。

定义:

```
function hideAllPoiNode(): void;
```

ts

用法:

```
ssp.hideAllPoiNode();
```

js

[← Poi 对象](#)[空间画布对象 →](#)