

Guide Technique - Utilisation des Hotspots pour Annotation d'Images 3D

Version : 1.1 **Date :** 22/12/2025 **Objectif :** Annoter des images 3D avec des hotspots projetés depuis des positions 3D

⚠ Note importante : Ce guide documente le comportement de l'API /Hotspot.

Table des Matières

1. [Introduction](#)
 2. [Contexte Métier](#)
 3. [Architecture](#)
 4. [Format des Positions 3D](#)
 5. [Construction du Payload API](#)
 6. [Réponse de l'API](#)
 7. [Exemple Concret](#)
-

1. Introduction

1.1 Objectif

Ce guide décrit comment utiliser l'API /Hotspot de Lumiscaphe WebRender pour projeter des positions 3D en coordonnées 2D sur une image générée, permettant d'annoter visuellement des zones spécifiques d'un objet 3D.

Périmètre :

- Comprendre le principe de projection 3D → 2D
- Construire le payload JSON pour l'API /Hotspot
- Interpréter la réponse (coordonnées 2D et visibilité)
- Créer des annotations visuelles (SVG, Canvas, HTML)

1.2 Prérequis

Techniques : - APIs REST - Coordonnées 3D et 2D - JSON - Géométrie de base

Accès : - API Lumiscaphe WebRender - Base de données 3D (GUID)

2. Contexte Métier

2.1 Qu'est-ce qu'un Hotspot ?

Un hotspot est un point d'intérêt 3D projeté en coordonnées 2D sur une image générée. Il permet d'annoter des zones spécifiques d'un objet 3D avec des labels, légendes, ou zones cliquables.

Cas d'usage : - Annoter les zones de couleur d'un schéma de peinture - Créer des légendes pour des composants techniques - Identifier des zones interactives sur une image - Générer des vues documentaires annotées

2.2 Principe de Projection

Problème : En 3D, les coordonnées sont exprimées dans l'espace (X, Y, Z). Sur une image 2D, les coordonnées sont en pixels (x, y).

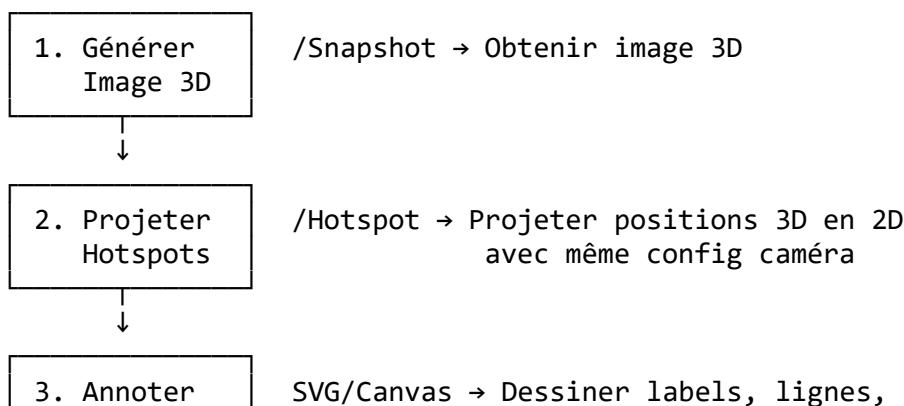
Solution : L'API /Hotspot effectue la projection 3D → 2D en utilisant :

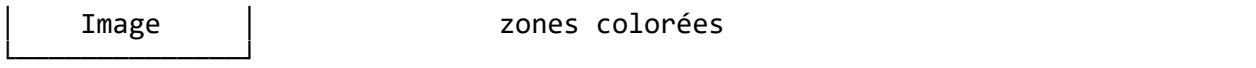
1. La configuration de la scène 3D
2. La caméra utilisée pour générer l'image
3. Les dimensions de l'image (width × height)

Résultat : Pour chaque position 3D fournie, l'API retourne :

- Les coordonnées 2D (x, y) en pixels sur l'image
- Les coordonnées 3D d'origine (echo de l'input)
- Un identifiant et nom optionnels
- La visibilité du point (Visible, Hidden, Occluded)

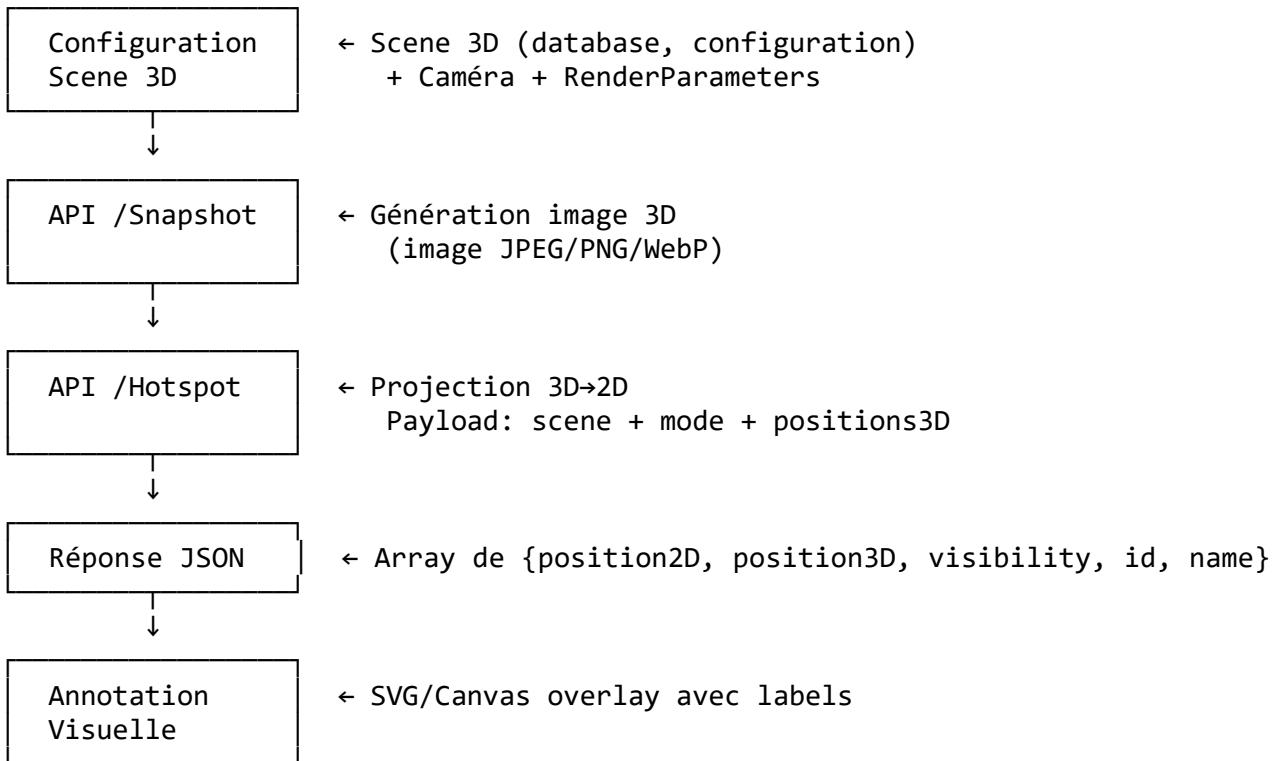
2.3 Workflow Typique





3. Architecture

3.1 Flux de Données



3.2 Coordonnées 3D

Repère 3D : - Axe X : Longitudinal - Axe Y : Vertical - Axe Z : Transversal

Format :

```
{
  "x": 3.475,
  "y": 1.620,
  "z": 0.578
}
```

Unité : Mètres (selon la base de données 3D)

3.3 Coordonnées 2D

Repère 2D :

- **Origine** : Coin supérieur gauche de l'image
- **Axe X** : Horizontal (gauche → droite)
- **Axe Y** : Vertical (haut → bas)

Format :

```
{
  "x": 1024.5,
  "y": 768.3
}
```

Unité : Pixels

Plage de valeurs : - 0 ≤ x ≤ width - 0 ≤ y ≤ height

4. Format des Positions 3D

4.1 Structure des Données

Les positions 3D doivent être fournies sous forme de tableau JSON :

Exemple (paint scheme “Zephyr”, 3 premières zones) :

```
[
  { "x": -3.475496768951416, "y": 1.6204111576080322, "z": 0.5782706141471863 },
  { "x": 3.8092002868652344, "y": 2.7245590686798096, "z": 0.0960141345858573 },
  { "x": 2.4987735748291016, "y": 1.487678050994873, "z": 0.4704386591911316 }
]
```

Propriétés : - x : (number) Coordonnée X en mètres - y : (number) Coordonnée Y en mètres
- z : (number) Coordonnée Z en mètres

4.3 Exemple de Fichier de Données

Format recommandé : Organiser par configuration ou schéma

Exemple réel (code/data/pdf-hotspots.json) :

```
{
  "Zephyr": {
    "hotspots": [
      {
        "name": "Zone A",
        "position3D": {
          "x": 1024.5,
          "y": 768.3
        }
      }
    ]
  }
}
```

```
        "x": -3.475496768951416,
        "y": 1.6204111576080322,
        "z": 0.5782706141471863
    },
},
{
    "name": "Zone B",
    "position3D": {
        "x": 3.8092002868652344,
        "y": 2.7245590686798096,
        "z": 0.09601413458585739
    },
},
{
    "name": "Zone C",
    "position3D": {
        "x": 2.4987735748291016,
        "y": 1.487678050994873,
        "z": 0.4704386591911316
    },
},
{
    "name": "Zone D",
    "position3D": {
        "x": -3.373769521713257,
        "y": 0.9474617838859558,
        "z": 0.5617594122886658
    },
},
{
    "name": "Zone A+",
    "position3D": {
        "x": -0.002033029682934284,
        "y": 0.9029922485351562,
        "z": 0.7174139022827148
    }
}
],
},
{
    "name": "Tehuano"
},
{
    "hotspots": [
        {
            "name": "Zone A",
            "position3D": {
                "x": -3.190016984939575,
                "y": 1.626016616821289,
                "z": 0.6183640956878662
            }
        },
        {
            "name": "Zone B"
        }
    ]
}
```

```

        "name": "Zone B",
        "position3D": {
            "x": 3.3159046173095703,
            "y": 1.5344599485397339,
            "z": 0.32665249705314636
        }
    }
}

```

5. Construction du Payload API

5.1 Structure du Payload

Le payload pour l'API /Hotspot contient 4 sections principales :

```
{
  "scene": { ... },
  "mode": { ... },
  "renderParameters": { ... },
  "positions": [ ... ]
}
```

5.2 Section scene

Objectif : Décrire la scène 3D (identique à celle utilisée pour /Snapshot)

Structure minimale :

```
{
  "scene": [
    {
      "database": "{DATABASE-GUID}"
    }
  ]
}
```

Propriétés : - database : (string) GUID de la base de données 3D - configuration : (string)

Note : La section scene doit être identique à celle utilisée pour générer l'image avec /Snapshot.

5.3 Section mode

Objectif : Spécifier la caméra utilisée (identique à celle utilisée pour /Snapshot)

Format 1 : Caméra unique

```
{
  "mode": {
    "image": {
    }
  }
}
```

```

        "camera": "{CAMERA-GUID}"
    }
}

```

Format 2 : Groupe de caméras

```

{
  "mode": {
    "images": {
      "cameraGroup": "{CAMERAGROUP-GUID}"
    }
  }
}

```

Note : La section mode doit utiliser la même caméra que celle utilisée pour générer l'image avec /Snapshot.

5.4 Section renderParameters

Objectif : Dimensions de l'image (identiques à celles utilisées pour /Snapshot)

```

{
  "renderParameters": {
    "width": 1920,
    "height": 1080
  }
}

```

Propriétés :

- width : (integer) Largeur de l'image en pixels
- height : (integer) Hauteur de l'image en pixels

Note : Les dimensions doivent être identiques à celles utilisées pour générer l'image avec /Snapshot.

5.5 Section positions

Objectif : Tableau des positions 3D à projeter

Exemple (paint scheme “Zephyr”, 3 premières zones) :

```

{
  "positions": [
    { "x": -3.475496768951416, "y": 1.6204111576080322, "z": 0.57827061414718
63 },
    { "x": 3.8092002868652344, "y": 2.7245590686798096, "z": 0.09601413458585
739 },
    { "x": 2.4987735748291016, "y": 1.487678050994873, "z": 0.470438659191131
6 }
}

```

```
    ]  
}
```

Format : Tableau de vector3d (objets {x, y, z})

5.6 Payload Complet

Exemple avec vraies données (paint scheme “Zephyr”, 5 zones) :

```
{  
  "scene": [{  
    "database": "{DATABASE-GUID}",  
    "configuration": "Exterior_PaintScheme.Zephyr_A-0_A-D_A-D_A-D_A-D"  
  }],  
  "mode": {  
    "image": {  
      "camera": "{CAMERA-GUID}"  
    }  
  },  
  "renderParameters": {  
    "width": 1920,  
    "height": 1080  
  },  
  "positions": [  
    { "x": -3.475496768951416, "y": 1.6204111576080322, "z": 0.57827061414718  
63 },  
    { "x": 3.8092002868652344, "y": 2.7245590686798096, "z": 0.09601413458585  
739 },  
    { "x": 2.4987735748291016, "y": 1.487678050994873, "z": 0.470438659191131  
6 },  
    { "x": -3.373769521713257, "y": 0.9474617838859558, "z": 0.56175941228866  
58 },  
    { "x": -0.002033029682934284, "y": 0.9029922485351562, "z": 0.71741390228  
27148 }  
  ]  
}
```

5.7 Appel API

Endpoint : POST <https://wr-daher.lumiscaphe.com/Hotspot>

Headers :

```
Content-Type: application/json  
Authorization: Bearer {token}
```

Body : Payload JSON complet

Exemple cURL :

```
curl -X POST "https://wr-daher.lumiscaphe.com/Hotspot" \  
-H "Content-Type: application/json" \
```

```

-d '{  

  "scene": [{"database": "...", "configuration": "..."}],  

  "mode": {"image": {"camera": "..."}},  

  "renderParameters": {"width": 1920, "height": 1080},  

  "positions": [  

    {"x": -3.475496768951416, "y": 1.6204111576080322, "z": 0.5782706141471  

863},  

    {"x": 3.8092002868652344, "y": 2.7245590686798096, "z": 0.0960141345858  

5739}  

  ]  

}'

```

6. Réponse de l'API

6.1 Format de la Réponse

L'API retourne un tableau JSON contenant un objet par position 3D fournie :

```
[
  {
    "id": "00000000-0000-0000-0000-000000000000",
    "name": "",
    "position2D": { "x": 1015, "y": 254 },
    "position3D": { "x": -3.190017, "y": 1.62601662, "z": 0.6183641 },
    "visibility": "Occluded"
  },
  {
    "id": "00000000-0000-0000-0000-000000000000",
    "name": "",
    "position2D": { "x": 989, "y": 836 },
    "position3D": { "x": 3.31590462, "y": 1.53446, "z": 0.3266525 },
    "visibility": "Visible"
  },
  {
    "id": "00000000-0000-0000-0000-000000000000",
    "name": "",
    "position2D": { "x": 1021, "y": 563 },
    "position3D": { "x": 0.267177343, "y": 1.48284388, "z": 0.679250658 },
    "visibility": "Visible"
  }
]
```

6.2 Propriété position2D

Type : vector2d (objet {x, y})

Description : Coordonnées 2D du hotspot sur l'image, en pixels

Plage de valeurs : -0 ≤ x ≤ width -0 ≤ y ≤ height

Origine : Coin supérieur gauche de l'image

Exemple :

```
{  
  "x": 1015,  
  "y": 254  
}
```

6.2bis Propriété position3D

Type : vector3d (objet {x, y, z})

Description : Echo des coordonnées 3D fournies en entrée (utile pour validation)

Exemple :

```
{  
  "x": -3.190017,  
  "y": 1.62601662,  
  "z": 0.6183641  
}
```

Note : Les valeurs peuvent être légèrement arrondies par rapport à l'entrée.

6.3 Propriété visibility

Type : string

Valeurs possibles :

Valeur	Description
"Visible"	Le point est visible sur l'image
"Hidden"	Le point est hors du champ de vision de la caméra
"Occluded"	Le point est masqué par un autre objet 3D

Usage : Filtrer les hotspots à afficher

Exemple :

```
// Afficher uniquement les hotspots visibles  
const visibleHotspots = apiResponse.filter(  
  hotspot => hotspot.visibility === "Visible"  
)
```

6.4 Ordre des Résultats

Règle : Les résultats sont retournés **dans le même ordre** que les positions 3D fournies.

Exemple :

```
// Input positions (paint scheme "Zephyr", 2 premières zones)
const positions = [
  { x: -3.475496768951416, y: 1.6204111576080322, z: 0.5782706141471863 }, /
  // Index 0 - Zone A
  { x: 3.8092002868652344, y: 2.7245590686798096, z: 0.09601413458585739 }
  // Index 1 - Zone B
];

// Output results
const results = [
  {
    position2D: { x: 1015, y: 254 },
    position3D: {...},
    visibility: "Visible"
  }, // Index 0 → correspond à positions[0]
  {
    position2D: { x: 989, y: 836 },
    position3D: {...},
    visibility: "Occluded"
  } // Index 1 → correspond à positions[1]
];
```

Utilisation :

```
for (let i = 0; i < positions.length; i++) {
  const input3D = positions[i];
  const output = results[i];
  console.log(`Position 3D (${input3D.x}, ${input3D.y}, ${input3D.z}) → 2D
  (${output.position2D.x}, ${output.position2D.y})`);
  console.log(`Visibilité: ${output.visibility}`);
}
```

7. Exemple Concret

7.1 Contexte

Objectif : Annoter 5 zones de couleur sur un avion TBM 980 avec schéma de peinture “Tehuano” et immatriculation “N980TB”

7.2 Étape 1 : Générer l’Image avec /Snapshot

Payload /Snapshot complet (incluant configuration complète + immatriculation) :

```
{
  "scene": [
    "database": "b0492c20-95b9-4094-85ff-dae8250440ad",
    "configuration": "Version.980/Exterior_Colors_ZoneA+.SocataWhite-29017-#d"
```

```

cdcd7-#D9D7C8-A+-29017-socata-white-solid-light/Exterior_Colors_ZoneA.CobaltBlue-05013-#3c495f-#282742-noA+-05013-cobalt-blue-dark-solid/Exterior_Colors_ZoneB.SocataWhite-29017-#cdcd7-#D9D7C8-A+-29017-socata-white-solid-light/Exterior_Colors_ZoneC.MelonYellow-01028-#ed9936-#FF9700-noA+-01028-orange-melon-yellow-solid/Exterior_Colors_ZoneD.SilverGrey-07001-#939a9e-#8A8F90-noA+-07001-silver-grey-gray-solid/Exterior_PaintScheme.Tehuano_6_A-0_A-D_A-D_A-D_A-D/Interior_Carpet.LightBrown_carpet_Premium/Interior_CentralSeatMaterial.Leather_Premium/Interior_LowerSidePanel.BeigeGray_2176_Leather_Premium/Interior_Metal_Finish.BushedStainless_metal_Premium/Interior_PerforatedSeatOptions.NoSeatPerforation_Premium/Interior_SeatCovers.BeigeGray_2176_Leather_Premium/Interior_Seatbelts.OatMeal_belt/Interior_Stitching.LightSand_Premium/Interior_TabletFinish.SapelliMat_table_wood_Premium/Interior_Ultra-SuedeRibbon.Elephant_3367_Suede_Premium/Interior_UpperSidePanel.WhiteSand_2192_Leather_Premium/Decor.Studio_Ground/Position.Studio/Exterior_Spinner.PolishedAluminium_1/SunGlass.SunGlassOFF/Tablet.Closed/Lighting_mood.undefined/Door_pilot.Closed/Door_passenger.Closed",
  "materials": [
    { "name": "RegL0", "filename": "Style_A_Right_N" },
    { "name": "RegR0", "filename": "Style_A_Left_N" },
    { "name": "RegL1", "filename": "Style_A_Right_9" },
    { "name": "RegR1", "filename": "Style_A_Left_9" },
    { "name": "RegL2", "filename": "Style_A_Right_8" },
    { "name": "RegR2", "filename": "Style_A_Left_8" },
    { "name": "RegL3", "filename": "Style_A_Right_0" },
    { "name": "RegR3", "filename": "Style_A_Left_0" },
    { "name": "RegL4", "filename": "Style_A_Right_T" },
    { "name": "RegR4", "filename": "Style_A_Left_T" },
    { "name": "RegL5", "filename": "Style_A_Right_B" },
    { "name": "RegR5", "filename": "Style_A_Left_B" }
  ],
  "materialMultiLayers": [
    { "name": "Style_A_Left_N", "layer": 0, "diffuseColor": "#282742" },
    { "name": "Style_A_Right_N", "layer": 0, "diffuseColor": "#282742" },
    { "name": "Style_A_Left_N", "layer": 1, "diffuseColor": "#282742" },
    { "name": "Style_A_Right_N", "layer": 1, "diffuseColor": "#282742" },
    { "name": "Style_A_Left_9", "layer": 0, "diffuseColor": "#282742" },
    { "name": "Style_A_Right_9", "layer": 0, "diffuseColor": "#282742" },
    { "name": "Style_A_Left_9", "layer": 1, "diffuseColor": "#282742" },
    { "name": "Style_A_Right_9", "layer": 1, "diffuseColor": "#282742" },
    { "name": "Style_A_Left_8", "layer": 0, "diffuseColor": "#282742" },
    { "name": "Style_A_Right_8", "layer": 0, "diffuseColor": "#282742" },
    { "name": "Style_A_Left_8", "layer": 1, "diffuseColor": "#282742" },
    { "name": "Style_A_Right_8", "layer": 1, "diffuseColor": "#282742" },
    { "name": "Style_A_Left_0", "layer": 0, "diffuseColor": "#282742" },
    { "name": "Style_A_Right_0", "layer": 0, "diffuseColor": "#282742" },
    { "name": "Style_A_Left_0", "layer": 1, "diffuseColor": "#282742" },
    { "name": "Style_A_Right_0", "layer": 1, "diffuseColor": "#282742" },
    { "name": "Style_A_Left_T", "layer": 0, "diffuseColor": "#282742" },
    { "name": "Style_A_Right_T", "layer": 0, "diffuseColor": "#282742" },
    { "name": "Style_A_Left_T", "layer": 1, "diffuseColor": "#282742" }
  ]
}

```

```

        {
            "name": "Style_A_Right_T", "layer": 1, "diffuseColor": "#282742" },
        {
            "name": "Style_A_Left_B", "layer": 0, "diffuseColor": "#282742" },
        {
            "name": "Style_A_Right_B", "layer": 0, "diffuseColor": "#282742" },
        {
            "name": "Style_A_Left_B", "layer": 1, "diffuseColor": "#282742" },
        {
            "name": "Style_A_Right_B", "layer": 1, "diffuseColor": "#282742" }
    ],
    "surfaces": [
        {
            "tag": "RegL",
            "labels": [
                {
                    "index": 0, "translation": { "x": -0.657, "y": 0.004 } },
                {
                    "index": 1, "translation": { "x": -0.407, "y": 0.004 } },
                {
                    "index": 2, "translation": { "x": -0.157, "y": 0.004 } },
                {
                    "index": 3, "translation": { "x": 0.093, "y": 0.004 } },
                {
                    "index": 4, "translation": { "x": 0.343, "y": 0.004 } },
                {
                    "index": 5, "translation": { "x": 0.593, "y": 0.004 } }
            ]
        },
        {
            "tag": "RegR",
            "labels": [
                {
                    "index": 0, "translation": { "x": -0.59, "y": 0.004 } },
                {
                    "index": 1, "translation": { "x": -0.34, "y": 0.004 } },
                {
                    "index": 2, "translation": { "x": -0.09, "y": 0.004 } },
                {
                    "index": 3, "translation": { "x": 0.16, "y": 0.004 } },
                {
                    "index": 4, "translation": { "x": 0.41, "y": 0.004 } },
                {
                    "index": 5, "translation": { "x": 0.66, "y": 0.004 } }
            ]
        }
    ],
    "mode": {
        "image": {
            "camera": "ab99e549-e289-4b38-9af9-b3b224962ab1"
        }
    },
    "renderParameters": {
        "width": 1080,
        "height": 1080,
        "antialiasing": true,
        "superSampling": "2"
    },
    "encoder": {
        "png": { "compression": 6 }
    }
}

```

Résultat : Image PNG de l'avion TBM 980 avec paint scheme Tehuano et immatriculation N980TB (1080×1080)

7.3 Étape 2 : Projeter les Hotspots avec /Hotspot

Payload /Hotspot :

```
{  
  "scene": [{}  
    "database": "b0492c20-95b9-4094-85ff-dae8250440ad"  
  ],  
  "mode": {  
    "image": {  
      "camera": "ab99e549-e289-4b38-9af9-b3b224962ab1"  
    }  
  },  
  "renderParameters": {  
    "width": 1080,  
    "height": 1080,  
    "antialiasing": true,  
    "superSampling": "2"  
  },  
  "positions": [  
    { "x": -3.190016984939575, "y": 1.626016616821289, "z": 0.618364095687866  
  2 },  
    { "x": 3.3159046173095703, "y": 1.5344599485397339, "z": 0.32665249705314  
  636 },  
    { "x": 0.2671773433685303, "y": 1.4828438758850098, "z": 0.67925065755844  
  12 },  
    { "x": -3.2331554889678955, "y": 0.9674192667007446, "z": 0.5862522125244  
  141 },  
    { "x": 0.04127908870577812, "y": 0.8967769145965576, "z": 0.7187203764915  
  466 }  
  ]  
}
```

Note : Le payload /Hotspot est simplifié et ne contient que :

- La base de données (sans configuration complète)
- La caméra (identique à /Snapshot)
- Les dimensions de l'image (identiques à /Snapshot) - Les 5 positions 3D à projeter

Réponse de l'API (réelle) :

```
[  
  {  
    "id": "00000000-0000-0000-0000-000000000000",  
    "name": "",  
    "position2D": { "x": 595, "y": 254 },  
    "position3D": { "x": -3.190017, "y": 1.62601662, "z": 0.6183641 },  
    "visibility": "Occluded"  
  },
```

```

{
  "id": "00000000-0000-0000-0000-000000000000",
  "name": "",
  "position2D": { "x": 569, "y": 836 },
  "position3D": { "x": 3.31590462, "y": 1.53446, "z": 0.3266525 },
  "visibility": "Visible"
},
{
  "id": "00000000-0000-0000-0000-000000000000",
  "name": "",
  "position2D": { "x": 601, "y": 563 },
  "position3D": { "x": 0.267177343, "y": 1.48284388, "z": 0.679250658 },
  "visibility": "Occluded"
},
{
  "id": "00000000-0000-0000-0000-000000000000",
  "name": "",
  "position2D": { "x": 574, "y": 758 },
  "position3D": { "x": -3.23315549, "y": 0.967419267, "z": 0.586252213 },
  "visibility": "Occluded"
},
{
  "id": "00000000-0000-0000-0000-000000000000",
  "name": "",
  "position2D": { "x": 598, "y": 795 },
  "position3D": { "x": 0.0412790887, "y": 0.896776915, "z": 0.718720376 },
  "visibility": "Visible"
}
]

```

Analyse des résultats :

- **Zone A** : Position2D (595, 254) - Occluded
- **Zone B** : Position2D (569, 836) - Visible
- **Zone C** : Position2D (601, 563) - Occluded
- **Zone D** : Position2D (574, 758) - Occluded
- **Zone A+** : Position2D (598, 795) - Visible