Skip to content

XrSessionState(bpy_struct)

base class — `bpy_struct`

class bpy.types.**XrSessionState(bpy_struct)**

Runtime state information about the VR session

actionmaps

TYPE:

`XrActionMaps` `bpy_prop_collection` of `XrActionMap`, (readonly)

active_actionmap

TYPE:

int in [-inf, inf], default 0

navigation_location

Location offset to apply to base pose when determining viewer location

TYPE:

`mathutils.Vector` of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

navigation_rotation

Rotation offset to apply to base pose when determining viewer rotation

TYPE:

`mathutils.Quaternion` rotation of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

navigation_scale

Additional scale multiplier to apply to base scale when determining viewer scale

TYPE:

float in [-inf, inf], default 0.0

selected_actionmap

TYPE:

int in [-inf, inf], default 0

viewer_pose_location

Last known location of the viewer pose (center between the eyes) in world space

TYPE:

`mathutils.Vector` of 3 items in [-inf, inf], default (0.0, 0.0, 0.0), (readonly)

viewer_pose_rotation

Last known rotation of the viewer pose (center between the eyes) in world space

TYPE:

`mathutils.Quaternion` rotation of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0), (readonly)

classmethod is_running(context)

Query if the VR session is currently running

RETURNS:

Result

RETURN TYPE:

boolean

classmethod reset_to_base_pose(context)

Force resetting of position and rotation deltas

classmethod action_set_create(context, actionmap)

Create a VR action set

RETURNS:

Result

RETURN TYPE:

boolean

classmethod action_create(context, actionmap, actionmap_item)

Create a VR action

RETURNS:

Result

RETURN TYPE:

boolean

classmethod action_binding_create(context, actionmap, actionmap_item, actionmap_binding)

Create a VR action binding

RETURNS:

Result

RETURN TYPE:

boolean

classmethod active_action_set_set(context, action_set)

Set the active VR action set

PARAMETERS:

action_set (*string, (never None)*) – Action Set, Action set name

RETURNS:

Result

RETURN TYPE:

boolean

classmethod controller_pose_actions_set(context, action_set, grip_action, aim_action)

Set the actions that determine the VR controller poses

PARAMETERS:

- **action_set** (*string, (never None)*) – Action Set, Action set name
- **grip_action** (*string, (never None)*) – Grip Action, Name of the action representing the controller grips
- **aim_action** (*string, (never None)*) – Aim Action, Name of the action representing the controller aims

RETURNS:

Result

RETURN TYPE:

boolean

classmethod action_state_get(context, action_set_name, action_name, user_path)

Get the current state of a VR action

PARAMETERS:

- **action_set_name** (*string, (never None)*) – Action Set, Action set name
- **action_name** (*string, (never None)*) – Action, Action name
- **user_path** (*string, (never None)*) – User Path, OpenXR user path

RETURNS:

Action State, Current state of the VR action. Second float value is only set for 2D vector type actions.

RETURN TYPE:

float array of 2 items in [-inf, inf], (never None)

classmethod haptic_action_apply(context, action_set_name, action_name, user_path, duration, frequency, amplitude)

Apply a VR haptic action

PARAMETERS:

- **action_set_name** (*string, (never None)*) – Action Set, Action set name
- **action_name** (*string, (never None)*) – Action, Action name
- **user_path** (*string, (never None)*) – User Path, Optional OpenXR user path. If not set, the action will be applied to all paths.
- **duration** (*float in [0, inf]*) – Duration, Haptic duration in seconds. 0.0 is the minimum supported duration.
- **frequency** (*float in [0, inf]*) – Frequency, Frequency of the haptic vibration in hertz. 0.0 specifies the OpenXR runtime's default frequency.
- **amplitude** (*float in [0, 1]*) – Amplitude, Haptic amplitude, ranging from 0.0 to 1.0

RETURNS:

Result

RETURN TYPE:

boolean

classmethod haptic_action_stop(context, action_set_name, action_name, user_path)

Stop a VR haptic action

PARAMETERS:

- **action_set_name** (*string, (never None)*) – Action Set, Action set name
- **action_name** (*string, (never None)*) – Action, Action name
- **user_path** (*string, (never None)*) – User Path, Optional OpenXR user path. If not set, the action will be stopped for all paths.

classmethod controller_grip_location_get(context, index)

Get the last known controller grip location in world space

PARAMETERS:

index (*int in [0, 255]*) – Index, Controller index

RETURNS:

Location, Controller grip location

RETURN TYPE:

`mathutils.Vector` of 3 items in [-inf, inf], (never None)

classmethod controller_grip_rotation_get(context, index)

Get the last known controller grip rotation (quaternion) in world space

PARAMETERS:

index (*int in [0, 255]*) – Index, Controller index

RETURNS:

Rotation, Controller grip quaternion rotation

Rotation, Controller grip quaternion rotation

RETURN TYPE:

`mathutils.Quaternion` rotation of 4 items in [-inf, inf], (never None)

classmethod controller_aim_location_get(context, index)

Get the last known controller aim location in world space

PARAMETERS:

index (*int in [0, 255]*) – Index, Controller index

RETURNS:

Location, Controller aim location

RETURN TYPE:

`mathutils.Vector` of 3 items in [-inf, inf], (never None)

classmethod controller_aim_rotation_get(context, index)

Get the last known controller aim rotation (quaternion) in world space

PARAMETERS:

index (*int in [0, 255]*) – Index, Controller index

RETURNS:

Rotation, Controller aim quaternion rotation

RETURN TYPE:

`mathutils.Quaternion` rotation of 4 items in [-inf, inf], (never None)

classmethod bl_rna_get_subclass(id, default=None)

PARAMETERS:

id (*str*) – The RNA type identifier.

RETURNS:

The RNA type or default when not found.

RETURN TYPE:

`bpy.types.Struct` subclass

classmethod bl_rna_get_subclass_py(id, default=None)

PARAMETERS:

id (*str*) – The RNA type identifier.

RETURNS:

The class or default when not found.

RETURN TYPE:

type

Inherited Properties

- `bpy_struct.id_data`

Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- bpy_struct.get
- bpy_struct.id_properties_clear
- bpy_struct.id_properties_ensure
- bpy_struct.id_properties_ui
- bpy_struct.is_property_hidden
- bpy_struct.is_property_overridable_library
- bpy_struct.is_property_readonly
- bpy_struct.is_property_set
- bpy_struct.keys
- bpy_struct.path_from_id
- bpy_struct.path_resolve
- bpy_struct.pop
- bpy_struct.property_overridable_library_set
- bpy_struct.property_unset
- bpy_struct.type_recast
- bpy_struct.values

References

- WindowManager.xr_session_state
- XrActionMaps.find
- XrActionMaps.new
- XrActionMaps.new_from_actionmap
- XrActionMaps.remove