Smooth.State Class Reference

The state of an object: timestamp, position, rotation, scale, velocity, angular velocity. More...

Public Member Functions

void	**copyFromSmoothSync** (**SmoothSync** smoothSyncScript)
State	**copyFromState** (**State** state) Copy an existing **State**. More...
void	**resetTheVariables** ()
	State () Default constructor. Does nothing. More...

Static Public Member Functions

static **State**	**Lerp** (**State** targetTempState, **State** start, **State** end, float t) Returns a Lerped state that is between two States in time. More...

Public Attributes

Vector3	**angularVelocity** The angularVelocity of the owned object when the state was sent. More...
bool	**atPositionalRest** If this **State** is tagged as a positional rest **State**, it should stop extrapolating position on non-owners. More...
bool	**atRotationalRest** If this **State** is tagged as a rotational rest **State**, it should stop extrapolating rotation on non-owners. More...
float	**ownerTimestamp** The network timestamp of the owner when the state was sent. More...
Vector3	**position** The position of the owned object when the state was sent. More...
float	**receivedOnServerTimestamp** The time on the server when the **State** is validated. Only used by server for latestVerifiedState. More...
Vector3	**reusableRotationVector** Used in Deserialize() so we don't have to make a new Vector3 every time. More...
Quaternion	**rotation** The rotation of the owned object when the state was sent. More...
Vector3	**scale** The scale of the owned object when the state was sent. More...
bool	**serverShouldRelayAngularVelocity** = false The server will set this to true if it is received so we know to relay the information back out to

other clients. More...

bool	**serverShouldRelayPosition** = false The server will set this to true if it is received so we know to relay the information back out to other clients. More...
bool	**serverShouldRelayRotation** = false The server will set this to true if it is received so we know to relay the information back out to other clients. More...
bool	**serverShouldRelayScale** = false The server will set this to true if it is received so we know to relay the information back out to other clients. More...
bool	**serverShouldRelayVelocity** = false The server will set this to true if it is received so we know to relay the information back out to other clients. More...
bool	**teleport** If this **State** is tagged as a teleport **State**, it should be moved immediately to instead of lerped to. More...
Vector3	**velocity** The velocity of the owned object when the state was sent. More...

Detailed Description

The state of an object: timestamp, position, rotation, scale, velocity, angular velocity.

Constructor & Destructor Documentation

◆ State()

Smooth.State.State ( ) inline

Default constructor. Does nothing.

Member Function Documentation

◆ copyFromState()

State Smooth.State.copyFromState (**State** state) inline

Copy an existing **State**.

◆ Lerp()

static **State** Smooth.State.Lerp (**State** targetTempState,
State start,
State end,
float t
) inline static

Returns a Lerped state that is between two States in time.

Parameters

start Start **State**

end End **State**

t Time

Returns

Member Data Documentation

◆ angularVelocity

Vector3 Smooth.State.angularVelocity

The angularVelocity of the owned object when the state was sent.

◆ atPositionalRest

bool Smooth.State.atPositionalRest

If this **State** is tagged as a positional rest **State**, it should stop extrapolating position on non-owners.

◆ atRotationalRest

bool Smooth.State.atRotationalRest

If this **State** is tagged as a rotational rest **State**, it should stop extrapolating rotation on non-owners.

◆ ownerTimestamp

float Smooth.State.ownerTimestamp

The network timestamp of the owner when the state was sent.

◆ position

Vector3 Smooth.State.position

The position of the owned object when the state was sent.

◆ receivedOnServerTimestamp

float Smooth.State.receivedOnServerTimestamp

The time on the server when the **State** is validated. Only used by server for latestVerifiedState.

◆ reusableRotationVector

Vector3 Smooth.State.reusableRotationVector

Used in Deserialize() so we don't have to make a new Vector3 every time.

◆ rotation

Quaternion Smooth.State.rotation

The rotation of the owned object when the state was sent.

◆ scale

Vector3 Smooth.State.scale

The scale of the owned object when the state was sent.

◆ serverShouldRelayAngularVelocity

bool Smooth.State.serverShouldRelayAngularVelocity = false

The server will set this to true if it is received so we know to relay the information back out to other clients.

◆ serverShouldRelayPosition

bool Smooth.State.serverShouldRelayPosition = false

The server will set this to true if it is received so we know to relay the information back out to other clients.

◆ serverShouldRelayRotation

bool Smooth.State.serverShouldRelayRotation = false

The server will set this to true if it is received so we know to relay the information back out to other clients.

◆ serverShouldRelayScale

bool Smooth.State.serverShouldRelayScale = false

The server will set this to true if it is received so we know to relay the information back out to other clients.

◆ serverShouldRelayVelocity

bool Smooth.State.serverShouldRelayVelocity = false

The server will set this to true if it is received so we know to relay the information back out to other clients.

◆ teleport

bool Smooth.State.teleport

If this **State** is tagged as a teleport **State**, it should be moved immediately to instead of lerped to.

◆ velocity

Vector3 Smooth.State.velocity

The velocity of the owned object when the state was sent.