

# IceGrid-Session

---

## IceGrid::Session

### Overview

#### interface Session extends [Glacier2::Session](#)

A session object is used by IceGrid clients to allocate and release objects. Client sessions are either created with the [IceGrid::Registry](#) object or the registry client [Glacier2::SessionManager](#) object.

See Also

- [IceGrid::Registry](#)
- [Glacier2::SessionManager](#)

### Operation Index

[keepAlive](#) — Keep the session alive.  
[allocateObjectById](#) — Allocate an object.  
[allocateObjectByType](#) — Allocate an object with the given type.  
[releaseObject](#) — Release an object.  
[setAllocationTimeout](#) — Set the allocation timeout.

### Operations

#### void keepAlive()

Keep the session alive. Clients should call this operation regularly to prevent the server from reaping the session.

See Also

- [IceGrid::Registry::getSessionTimeout](#)

#### [ "amd" ] Object\* allocateObjectById([Ice::Identity](#) id) throws [IceGrid::ObjectNotRegisteredException](#), [IceGrid::AllocationException](#)

Allocate an object. Depending on the allocation timeout, this operation might hang until the object is available or until the timeout is reached.

Parameters

id — The identity of the object to allocate.

Return Value

The proxy of the allocated object.

Exceptions

[IceGrid::ObjectNotRegisteredException](#) — Raised if the object with the given identity is not registered with the registry.  
[IceGrid::AllocationException](#) — Raised if the object can't be allocated.

See Also

- [setAllocationTimeout](#)
- [releaseObject](#)

#### [ "amd" ] Object\* allocateObjectByType(string type) throws [IceGrid::AllocationException](#)

Allocate an object with the given type. Depending on the allocation timeout, this operation can block until an object becomes available or until the timeout is reached.

Parameters

`type` — The type of the object.

Return Value

The proxy of the allocated object.

Exceptions

[IceGrid::ObjectNotRegisteredException](#) — Raised if no objects with the given type can be allocated.

[IceGrid::AllocationException](#) — Raised if the object could not be allocated.

See Also

- [setAllocationTimeout](#)
- [releaseObject](#)

## **void releaseObject([Ice::Identity](#) id) throws [IceGrid::ObjectNotRegisteredException](#), [IceGrid::AllocationException](#)**

Release an object.

Parameters

`id` — The identity of the object to release.

Exceptions

[IceGrid::ObjectNotRegisteredException](#) — Raised if the object with the given identity is not registered with the registry.

[IceGrid::AllocationException](#) — Raised if the given object can't be released. This might happen if the object isn't allocatable or isn't allocated by the session.

See Also

- [allocateObjectById](#)
- [allocateObjectByType](#)

## **void setAllocationTimeout(int timeout)**

Set the allocation timeout. If no objects are available for an allocation request, the request will hang for the duration of this timeout.

Parameters

`timeout` — The timeout in milliseconds.

See Also

- [allocateObjectById](#)
  - [allocateObjectByType](#)
-