

Glacier2-Router

Glacier2::Router

Overview

interface Router extends [Ice::Router](#)

The Glacier2 specialization of the [Ice::Router](#) interface.

Operation Index

[getCategoryForClient](#) — This category must be used in the identities of all of the client's callback objects.

[createSession](#) — Create a per-client session with the router.

[createSessionFromSecureConnection](#) — Create a per-client session with the router.

[refreshSession](#) — Keep the calling client's session with this router alive.

[destroySession](#) — Destroy the calling client's session with this router.

[getSessionTimeout](#) — Get the value of the session timeout.

Operations

["nonmutating"] string getCategoryForClient()

This category must be used in the identities of all of the client's callback objects. This is necessary in order for the router to forward callback requests to the intended client. If the Glacier2 server endpoints are not set, the returned category is an empty string.

Return Value

The category.

["amd", "format:sliced"] [Glacier2::Session*](#) createSession(string userId, string password) throws [Glacier2::PermissionDeniedException](#), [Glacier2::CannotCreateSessionException](#)

Create a per-client session with the router. If a [Glacier2::SessionManager](#) has been installed, a proxy to a [Glacier2::Session](#) object is returned to the client. Otherwise, null is returned and only an internal session (i.e., not visible to the client) is created. If a session proxy is returned, it must be configured to route through the router that created it. This will happen automatically if the router is configured as the client's default router at the time the session proxy is created in the client process, otherwise the client must configure the session proxy explicitly.

Parameters

`userId` — The user id for which to check the password.

`password` — The password for the given user id.

Return Value

A proxy for the newly created session, or null if no [Glacier2::SessionManager](#) has been installed.

Exceptions

[Glacier2::PermissionDeniedException](#) — Raised if the password for the given user id is not correct, or if the user is not allowed access.

[Glacier2::CannotCreateSessionException](#) — Raised if the session cannot be created.

See Also

- [Glacier2::Session](#)
- [Glacier2::SessionManager](#)
- [Glacier2::PermissionsVerifier](#)

["amd", "format:sliced"] [Glacier2::Session*](#) createSessionFromSecureConnection() throws [Glacier2::PermissionDeniedException](#), [Glacier2::CannotCreateSessionException](#)

Create a per-client session with the router. The user is authenticated through the SSL certificates that have been associated with the connection. If a [Glacier2::SessionManager](#) has been installed, a proxy to a [Glacier2::Session](#) object is returned to the client. Otherwise, null is returned and only an internal session (i.e., not visible to the client) is created. If a session proxy is returned, it must be configured to route through the router that created it. This will happen automatically if the router is configured as the client's default router at the time the session proxy is created in the client process, otherwise the client must configure the session proxy explicitly.

Return Value

A proxy for the newly created session, or null if no [Glacier2::SessionManager](#) has been installed.

Exceptions

[Glacier2::PermissionDeniedException](#) — Raised if the user cannot be authenticated or if the user is not allowed access.
[Glacier2::CannotCreateSessionException](#) — Raised if the session cannot be created.

See Also

- [Glacier2::Session](#)
- [Glacier2::SessionManager](#)
- [Glacier2::PermissionsVerifier](#)

void refreshSession() throws [Glacier2::SessionNotExistException](#)

Keep the calling client's session with this router alive.

Exceptions

[Glacier2::SessionNotExistException](#) — Raised if no session exists for the calling client.

void destroySession() throws [Glacier2::SessionNotExistException](#)

Destroy the calling client's session with this router.

Exceptions

[Glacier2::SessionNotExistException](#) — Raised if no session exists for the calling client.

["nonmutating"] long getSessionTimeout()

Get the value of the session timeout. Sessions are destroyed if they see no activity for this period of time.

Return Value

The timeout (in seconds).
