Document serial number: 000004892117

NP Toolkit Library Reference

000004892117

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

© 2015 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

Document serial number: 000004892117

Document serial number: 000004892117

Table of Contents

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Introduction

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Library Summary

Library Contents

Item	Description
sce::Toolkit::NP	The namespace for the NP Toolkit library.
sce::Toolkit::NP::AttachmentDetail	Contains details about an attachment.
sce::Toolkit::NP::AttachmentURL	Represents the URL of an attachment.
sce::Toolkit::NP::Auth	The namespace containing PlayStation™Network authentication functionality.
sce::Toolkit::NP::Auth::Interface	External interface to the PlayStation™Network authentication functionality.
sce::Toolkit::NP::CategoryBrowseParams	Contains information that is used to browse a category.
sce::Toolkit::NP::CategoryInfo	Contains information about a category on the PlayStation®Store.
sce::Toolkit::NP::CategoryInfoInputParams	Contains information that is used to retrieve information about a specific category that has been set up on the PlayStation®Store.
sce::Toolkit::NP::CategoryInfoSub	Contains information about a subcategory in the PlayStation®Store.
sce::Toolkit::NP::ChallengeConsumeRequest	Represents a request to consume a challenge.
sce::Toolkit::NP::ChallengeGetDataRequest	Represents a request to retrieve a challenge's binary data.
sce::Toolkit::NP::ChallengeGetItemListRequest	Represents a request to retrieve previous challenges.
sce::Toolkit::NP::ChallengeGetItemRequest	Represents a request to retrieve a single item.
sce::Toolkit::NP::ChallengeRecvDetails	Represents a received challenge.
sce::Toolkit::NP::ChallengeResponseRequest	Represents a request to notify a challenge.
sce::Toolkit::NP::Challenges	The namespace containing challenges functionality.
sce::Toolkit::NP::Challenges::Interface	The interface for accessing challenges.
sce::Toolkit::NP::ChallengeSendRequest	Represents a request to send a challenge.
sce::Toolkit::NP::CheckoutInputParams	Contains a list of SKUs to add to the checkout and a memory container.
sce::Toolkit::NP::Commerce	The namespace containing PlayStation™Network commerce functionality.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Item	Description
sce::Toolkit::NP::Commerce::Interface	External interface to the PlayStation™Network commerce functionality.
sce::Toolkit::NP::CommunicationId	Wraps up an NP Communication ID.
sce::Toolkit::NP::ConsumeEntitlementInputParams	Contains information that is used to consume a specified amount of a consumable service entitlement for a given user.
sce::Toolkit::NP::CreateNpSessionRequest	Provides information to be displayed on the Session server.
sce::Toolkit::NP::CreateSessionRequest	Provides information to be displayed on the matching server for a session.
sce::Toolkit::NP::DetailedProductInfoInputParams	Contains information that is used to retrieve detailed information on specific product.
sce::Toolkit::NP::DetailedProductInfoListInputParams	Contains information that is used to retrieve detailed information for a list of products.
sce::Toolkit::NP::DetailedProductInfoListInputParams::ProductId	Encapsulates a product ID.
sce::Toolkit::NP::DownloadListInputParams	Contains a list of SKUs to show in the download list to and a memory container.
sce::Toolkit::NP::Entitlement	Contains information about a service entitlement.
sce::Toolkit::NP::Event	Represents an event generated by a service.
sce::Toolkit::NP::FriendInfoRequest	Represents a request to retrieve a list of a user's friends.
sce::Toolkit::NP::FriendRankRequest	Represents a request to retrieve the ranks of a user's friends.
sce::Toolkit::NP::Friends	The namespace containing friends list functionality.
sce::Toolkit::NP::Friends::Interface	The interface for accessing the friends list.
sce::Toolkit::NP::GameCustomData	The namespace containing game custom data functionality.
sce::Toolkit::NP::GameCustomData::Interface	The game custom data interface class.
sce::Toolkit::NP::GameCustomDataGameDataRequest	Represents a request to get game custom data.
sce::Toolkit::NP::GameCustomDataItemsRequest	Represents a request to get game custom data items.
sce::Toolkit::NP::GameCustomDataMessageRequest	Represents a request to get a game custom data message.
sce::Toolkit::NP::GameCustomDataThumbnailRequest	Contains the request to get custom data thumbnail Image.
sce::Toolkit::NP::GameCustomDataUseFlagRequest	Represents a request to set the game custom data message usage flag.
sce::Toolkit::NP::GetEntitlementsInputParams	Contains information that is used get the service entitlements associated with a specified user.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Item	Description
sce::Toolkit::NP::IdDatabase	Manages the different IDs used inside the NP Toolkit library.
sce::Toolkit::NP::Interface	Contains the interface to the NP Toolkit library as a set of static methods.
sce::Toolkit::NP::InvitationDataRequest	Represents a request for invitation data.
sce::Toolkit::NP::InvitationInfoRequest	Represents a request for information about an invitation.
sce::Toolkit::NP::InvitationListRequest	Represents a request to retrieve an invitation list.
sce::Toolkit::NP::InviteJoinSessionRequest	Represents a request to join a session where the session is identified by an NP Session ID.
sce::Toolkit::NP::InviteMessage	Contains the components of an invite message.
sce::Toolkit::NP::JoinNpSessionRequest	Represents a request to join a session on the Session server.
sce::Toolkit::NP::JoinSessionRequest	The structure which needs to be passed in order to join a session.
sce::Toolkit::NP::KickMemberRequest	Represents a request to kick a member out of the room.
sce::Toolkit::NP::Matching	The namespace containing matching functionality.
sce::Toolkit::NP::Matching::Interface	The matching interface allows users to find other players and game sessions for online play.
sce::Toolkit::NP::MessageAttachment	Contains the components of a message.
sce::Toolkit::NP::MessageData	Contain the components of a message.
sce::Toolkit::NP::Messaging	The namespace containing messaging functionality.
sce::Toolkit::NP::Messaging::Interface	The static interface for sending messages.
sce::Toolkit::NP::ModifySessionAttributes	A structure used to specify attributes which needs to be modified.
sce::Toolkit::NP::ModifySessionRequest	A request structure used to specify how session information should be modified.
sce::Toolkit::NP::Near	The namespace containing the PlayStation™Network near service.
sce::Toolkit::NP::Near::Interface	The near interface contains a set of static methods for managing "near" actions.
sce::Toolkit::NP::NetInfo	The namespace containing network status functionality.
sce::Toolkit::NP::NetInfo::Interface	The interface for the network status.
sce::Toolkit::NP::NpSessionDetailedInformation	Represents detailed information about an NP Session.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Item	Description
sce::Toolkit::NP::NpSessionRequest	Represents a request to get Session server information.
sce::Toolkit::NP::Parameters	Contains the parameters for initializing the NP Toolkit library.
sce::Toolkit::NP::PostInvitationDataRequest	Represents a request to post invitation data.
sce::Toolkit::NP::Presence	The namespace containing presence functionality.
sce::Toolkit::NP::Presence::Interface	Allows a user's presence to be managed.
sce::Toolkit::NP::PresenceDetails	Holds the user's status string and data.
sce::Toolkit::NP::PresenceRequest	Represents a request to retrieve the presence information for a user's friend.
sce::Toolkit::NP::ProductBrowseParams	Contains the parameters needed to browse a product.
sce::Toolkit::NP::ProductListInputParams	Contains information that is used to retrieve a list of products from a specific category.
sce::Toolkit::NP::RangeOfRanksRequest	Represents a request to retrieve a range of ranks.
sce::Toolkit::NP::Ranking	The namespace containing ranking functionality.
sce::Toolkit::NP::Ranking::Interface	Ranking interface class.
sce::Toolkit::NP::RankingRequest	Represents the service label of the ranking service.
sce::Toolkit::NP::RegisterScoreRequest	Represents a request to register a user's score.
sce::Toolkit::NP::RegisterTrophyRequest	Represents a request to register a trophy package for a specific user.
sce::Toolkit::NP::Request	The base structure for a request, which contains the information required for all types of request.
sce::Toolkit::NP::RetrieveTrophyGameRequest	Represents a request to retrieve information about a games's trophy set.
sce::Toolkit::NP::RetrieveTrophyGroupRequest	Represents a request to retrieve trophy group information for a user.
sce::Toolkit::NP::RetrieveTrophyListRequest	Represents a request to retrieve a list of detailed trophy information for all the trophies in a games's trophy set.
sce::Toolkit::NP::RetrieveUserTrophyProgressRequest	Represents a request to retrieve a user's trophy progress.
sce::Toolkit::NP::SearchNpSessionRequest	Represents a request to search sessions on the Session server.
sce::Toolkit::NP::SearchSessionsRequest	A search descriptor to search for a session.
sce::Toolkit::NP::ServiceId	Wraps the string being used as a NP Service ID for commerce, ticketing, etc.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Item	Description
sce::Toolkit::NP::SessionInformation	Contains information about a session.
sce::Toolkit::NP::SessionMember	Provides information about the member in a session.
sce::Toolkit::NP::Sessions	The namespace containing session functionality on the PlayStation®4.
sce::Toolkit::NP::Sessions::Interface	The session interface allows users to find other players and game sessions for online play.
sce::Toolkit::NP::Sns	The namespace containing the PlayStation™Network SNS service.
sce::Toolkit::NP::Sns::Interface	The SNS interface allows users to post a message to Facebook.
sce::Toolkit::NP::SnsPostFacebook	Holds the necessary information needed to post a message to a Facebook user's wall.
sce::Toolkit::NP::Ticket	Represents a ticket.
sce::Toolkit::NP::Trophy	The namespace containing trophy functionality.
sce::Toolkit::NP::Trophy::Interface	Trophy interface class.
sce::Toolkit::NP::TSS	The namespace containing PlayStation™Network TSS (title small storage) functionality.
sce::Toolkit::NP::TSS::Interface	The external interface to the PlayStation™Network TSS (title small storage) functionality.
sce::Toolkit::NP::TssData	Represents TSS (title small storage) data.
sce::Toolkit::NP::TssGetStatusInputParams	Represents the input parameters used when getting the status of a TSS (title small storage) file from a specified slot.
sce::Toolkit::NP::TssInputParams	Represents the input parameters used when getting a TSS (title small storage) file from a specified slot.
sce::Toolkit::NP::TUS	The namespace containing PlayStation™Network TUS (title user storage) functionality.
sce::Toolkit::NP::TUS::Interface	The external interface to the PlayStation™Network TUS (title user storage) functionality.
sce::Toolkit::NP::TusData	Represents TUS (title user storage) data.
sce::Toolkit::NP::TusGetDataInputParams	Represents input parameters used when getting TUS (title user storage) data.
sce::Toolkit::NP::TusGetVarsInputParams	Represents the input parameters used when getting TUS (title user storage) variables.
sce::Toolkit::NP::TusSetDataInputParams	Represents input parameters used when setting TUS (title user storage) data.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Item	Description
sce::Toolkit::NP::TusSetVarsInputParams	Represents the input parameters used when setting TUS (title user storage) variables.
sce::Toolkit::NP::TusVariable	Represents a TUS (title user storage) variable.
sce::Toolkit::NP::UnlockTrophyRequest	Represents a request to unlock a trophy.
sce::Toolkit::NP::UpdateAttributeRequest	A request structure used to specify the session attributes to update.
sce::Toolkit::NP::UpdateNpSessionRequest	Represents a request to update information on the Session server.
sce::Toolkit::NP::UserProfile	The namespace containing PlayStation™Network user profile functionality.
sce::Toolkit::NP::UserProfile::Interface	The external interface to PlayStation™Network user profile functionality.
sce::Toolkit::NP::UserRankRequest	Represents a request to retrieve ranking information for a user.
sce::Toolkit::NP::Utilities	The namespace for utilities used by the NP Toolkit library.
sce::Toolkit::NP::Utilities::Future	A template implementation of the future class.
sce::Toolkit::NP::Utilities::FutureImpl	Represents a piece of data, for which an asynchronous reference is provided, that will be finalized at some point in the future.
sce::Toolkit::NP::VoucherInputParams	Contains details of how a voucher will be redeemed.
sce::Toolkit::NP::WordFilter	The namespace containing word filter functionality.
sce::Toolkit::NP::WordFilter::Interface	The external interface to the word filter functionality.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP

The namespace for the NP Toolkit library.

Definition

```
namespace NP {}
```

Description

The namespace for the NP Toolkit library.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::AttachmentDetail	Contains details about an attachment.
sce::Toolkit::NP::AttachmentURL	Represents the URL of an attachment.
sce::Toolkit::NP::Auth	The namespace containing PlayStation™Network authentication functionality.
sce::Toolkit::NP::CategoryBrowseParams	Contains information that is used to browse a category.
sce::Toolkit::NP::CategoryInfo	Contains information about a category on the PlayStation®Store.
sce::Toolkit::NP::CategoryInfoInputParams	Contains information that is used to retrieve information about a specific category that has been set up on the PlayStation®Store.
sce::Toolkit::NP::CategoryInfoSub	Contains information about a subcategory in the PlayStation®Store.
sce::Toolkit::NP::ChallengeConsumeRequest	Represents a request to consume a challenge.
sce::Toolkit::NP::ChallengeGetDataRequest	Represents a request to retrieve a challenge's binary data.
sce::Toolkit::NP::ChallengeGetItemListRequest	Represents a request to retrieve previous challenges.
sce::Toolkit::NP::ChallengeGetItemRequest	Represents a request to retrieve a single item.
sce::Toolkit::NP::ChallengeRecvDetails	Represents a received challenge.
sce::Toolkit::NP::ChallengeResponseRequest	Represents a request to notify a challenge.
sce::Toolkit::NP::Challenges	The namespace containing challenges functionality.
sce::Toolkit::NP::ChallengeSendRequest	Represents a request to send a challenge.
sce::Toolkit::NP::CheckoutInputParams	Contains a list of SKUs to add to the checkout and a memory container.
sce::Toolkit::NP::Commerce	The namespace containing PlayStation™Network commerce functionality.
sce::Toolkit::NP::CommunicationId	Wraps up an NP Communication ID.
sce::Toolkit::NP::ConsumeEntitlementInputParams	Contains information that is used to consume a specified amount of a consumable service entitlement for a given user.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Item	Description
sce::Toolkit::NP::CreateNpSessionRequest	Provides information to be displayed on the Session server.
sce::Toolkit::NP::CreateSessionRequest	Provides information to be displayed on the matching server for a session.
sce::Toolkit::NP::DetailedProductInfoInputParams	Contains information that is used to retrieve detailed information on specific product.
sce::Toolkit::NP::DetailedProductInfoListInputParams	Contains information that is used to retrieve detailed information for a list of products.
sce::Toolkit::NP::DownloadListInputParams	Contains a list of SKUs to show in the download list to and a memory container.
sce::Toolkit::NP::Entitlement	Contains information about a service entitlement.
sce::Toolkit::NP::Event	Represents an event generated by a service.
sce::Toolkit::NP::FriendInfoRequest	Represents a request to retrieve a list of a user's friends.
sce::Toolkit::NP::FriendRankRequest	Represents a request to retrieve the ranks of a user's friends.
sce::Toolkit::NP::Friends	The namespace containing friends list functionality.
sce::Toolkit::NP::GameCustomData	The namespace containing game custom data functionality.
sce::Toolkit::NP::GameCustomDataGameDataRequest	Represents a request to get game custom data.
sce::Toolkit::NP::GameCustomDataItemsRequest	Represents a request to get game custom data items.
sce::Toolkit::NP::GameCustomDataMessageRequest	Represents a request to get a game custom data message.
sce::Toolkit::NP::GameCustomDataThumbnailRequest	Contains the request to get custom data thumbnail Image.
sce::Toolkit::NP::GameCustomDataUseFlagRequest	Represents a request to set the game custom data message usage flag.
sce::Toolkit::NP::GetEntitlementsInputParams	Contains information that is used get the service entitlements associated with a specified user.
sce::Toolkit::NP::IdDatabase	Manages the different IDs used inside the NP Toolkit library.
sce::Toolkit::NP::Interface	Contains the interface to the NP Toolkit library as a set of static methods.
sce::Toolkit::NP::InvitationDataRequest	Represents a request for invitation data.
sce::Toolkit::NP::InvitationInfoRequest	Represents a request for information about an invitation.
sce::Toolkit::NP::InvitationListRequest	Represents a request to retrieve an invitation list.
sce::Toolkit::NP::InviteJoinSessionRequest	Represents a request to join a session where the session is identified by an NP Session ID.
sce::Toolkit::NP::InviteMessage	Contains the components of an invite message.
sce::Toolkit::NP::JoinNpSessionRequest	Represents a request to join a session on the Session server.
sce::Toolkit::NP::JoinSessionRequest	The structure which needs to be passed in order to join a session.
sce::Toolkit::NP::KickMemberRequest	Represents a request to kick a member out of the room.
sce::Toolkit::NP::Matching	The namespace containing matching functionality.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Item	Description
sce::Toolkit::NP::MessageAttachment	Contains the components of a message.
sce::Toolkit::NP::MessageData	Contain the components of a message.
sce::Toolkit::NP::Messaging	The namespace containing messaging functionality.
sce::Toolkit::NP::ModifySessionAttributes	A structure used to specify attributes which needs to be modified.
sce::Toolkit::NP::ModifySessionRequest	A request structure used to specify how session information should be modified.
sce::Toolkit::NP::Near	The namespace containing the PlayStation™Network near service.
sce::Toolkit::NP::NetInfo	The namespace containing network status functionality.
sce::Toolkit::NP::NpSessionDetailedInformation	Represents detailed information about an NP Session.
sce::Toolkit::NP::NpSessionRequest	Represents a request to get Session server information.
sce::Toolkit::NP::Parameters	Contains the parameters for initializing the NP Toolkit library.
sce::Toolkit::NP::PostInvitationDataRequest	Represents a request to post invitation data.
sce::Toolkit::NP::Presence	The namespace containing presence functionality.
sce::Toolkit::NP::PresenceDetails	Holds the user's status string and data.
sce::Toolkit::NP::PresenceRequest	Represents a request to retrieve the presence information for a user's friend.
sce::Toolkit::NP::ProductBrowseParams	Contains the parameters needed to browse a product.
sce::Toolkit::NP::ProductListInputParams	Contains information that is used to retrieve a list of products from a specific category.
sce::Toolkit::NP::RangeOfRanksRequest	Represents a request to retrieve a range of ranks.
sce::Toolkit::NP::Ranking	The namespace containing ranking functionality.
sce::Toolkit::NP::RankingRequest	Represents the service label of the ranking service.
sce::Toolkit::NP::RegisterScoreRequest	Represents a request to register a user's score.
sce::Toolkit::NP::RegisterTrophyRequest	Represents a request to register a trophy package for a specific user.
sce::Toolkit::NP::Request	The base structure for a request, which contains the information required for all types of request.
sce::Toolkit::NP::RetrieveTrophyGameRequest	Represents a request to retrieve information about a games's trophy set.
sce::Toolkit::NP::RetrieveTrophyGroupRequest	Represents a request to retrieve trophy group information for a user.
sce::Toolkit::NP::RetrieveTrophyListRequest	Represents a request to retrieve a list of detailed trophy information for all the trophies in a games's trophy set.
sce::Toolkit::NP::RetrieveUserTrophyProgressRequest	Represents a request to retrieve a user's trophy progress.
sce::Toolkit::NP::SearchNpSessionRequest	Represents a request to search sessions on the Session server.
sce::Toolkit::NP::SearchSessionsRequest	A search descriptor to search for a session.
sce::Toolkit::NP::ServiceId	Wraps the string being used as a NP Service ID for commerce, ticketing, etc.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Item	Description
sce::Toolkit::NP::SessionInformation	Contains information about a session.
sce::Toolkit::NP::SessionMember	Provides information about the member in a session.
sce::Toolkit::NP::Sessions	The namespace containing session functionality on the PlayStation®4.
sce::Toolkit::NP::Sns	The namespace containing the PlayStation™Network SNS service.
sce::Toolkit::NP::SnsPostFacebook	Holds the necessary information needed to post a message to a Facebook user's wall.
sce::Toolkit::NP::Ticket	Represents a ticket.
sce::Toolkit::NP::Trophy	The namespace containing trophy functionality.
sce::Toolkit::NP::TSS	The namespace containing PlayStation™Network TSS (title small storage) functionality.
sce::Toolkit::NP::TssData	Represents TSS (title small storage) data.
sce::Toolkit::NP::TssGetStatusInputParams	Represents the input parameters used when getting the status of a TSS (title small storage) file from a specified slot.
sce::Toolkit::NP::TssInputParams	Represents the input parameters used when getting a TSS (title small storage) file from a specified slot.
sce::Toolkit::NP::TUS	The namespace containing PlayStation™Network TUS (title user storage) functionality.
sce::Toolkit::NP::TusData	Represents TUS (title user storage) data.
sce::Toolkit::NP::TusGetDataInputParams	Represents input parameters used when getting TUS (title user storage) data.
sce::Toolkit::NP::TusGetVarsInputParams	Represents the input parameters used when getting TUS (title user storage) variables.
sce::Toolkit::NP::TusSetDataInputParams	Represents input parameters used when setting TUS (title user storage) data.
sce::Toolkit::NP::TusSetVarsInputParams	Represents the input parameters used when setting TUS (title user storage) variables.
sce::Toolkit::NP::TusVariable	Represents a TUS (title user storage) variable.
sce::Toolkit::NP::UnlockTrophyRequest	Represents a request to unlock a trophy.
sce::Toolkit::NP::UpdateAttributeRequest	A request structure used to specify the session attributes to update.
sce::Toolkit::NP::UpdateNpSessionRequest	Represents a request to update information on the Session server.
sce::Toolkit::NP::UserProfile	The namespace containing PlayStation™Network user profile functionality.
sce::Toolkit::NP::UserRankRequest	Represents a request to retrieve ranking information for a user.
sce::Toolkit::NP::Utilities	The namespace for utilities used by the NP Toolkit library.
sce::Toolkit::NP::VoucherInputParams	Contains details of how a voucher will be redeemed.
sce::Toolkit::NP::WordFilter	The namespace containing word filter functionality.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Type Definitions

AccessToken

Holds Facebook Access Token information.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::AccessToken {
            SceNpSnsFacebookLongAccessTokenResult result;
            String encodedToken;
            String permissions;
         } AccessToken;
      }
   }
}
```

Members

result	The access token result structure is passed to `sceNpSnsFbGetAccessToken()` and stores the result.
encodedToken	This string stores the encoded result necessary before appending to the Graph API path URL.
permissions	A string to store permissions for accessing a user's Facebook information.

Description

Holds Facebook Access Token information. This structure stores all the necessary variables for retrieving an access token from the Facebook Graph API. It holds additional information such as permissions that can be specified when requesting a token. The token received must be encoded before it can be appended to the Facebook Graph API URL.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

ActionLinkFb

Holds the necessary information needed to describe an action link to be posted to a Facebook user’s wall.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::ActionLinkFb {
            String name;
            String url;
         } ActionLinkFb;
      }
   }
}
```

Members

name	A string to hold the name of the action link that will be displayed as part of the stream story.
url	A string to hold the actual URL of the page that the user will be taken to upon clicking the name.

Description

Holds the necessary information needed to describe an action link to be posted to a Facebook user’s wall. An action link appears beside the “Like” and “Comment” options on a post.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

AvatarUrl

Contains the avatar URL.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef String AvatarUrl;
        }
    }
}
```

Description

Contains the avatar URL.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

BlockedList

Contains a list of blocked users.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< BlockedUser > BlockedList;
        }
    }
}
```

Description

Contains information on all users that are blocked by a user.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

BlockedUser

Contains information about a user who is blocked.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef NpUser BlockedUser;
        }
    }
}
```

Description

Contains information about a user who is blocked.

Notes

`Presence` information cannot be obtained for a blocked user.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

BlockedUsersInfoRequest

Represents a request to retrieve a list of users that the user has blocked.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::BlockedUsersInfoRequest {
                uint32_t flag;
            } BlockedUsersInfoRequest;
        }
    }
}
```

Members

flag	The specific flags for the request. Please refer to the `SCE_TOOLKIT_NP_FRIENDS_LIST*` flags.

Description

Represents a request to retrieve a list of users that the user has blocked. Information about each retrieved blocked user includes their NP ID.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

CategoryInfoSubList

Defines a list of subcategories in the PlayStation®Store.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::list< CategoryInfoSub > CategoryInfoSubList;
        }
    }
}
```

Description

Defines a list of subcategories in the PlayStation®Store.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ChallengeBinaryDataResult

Represents the result of a request to download the data attachment of a challenge.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::ChallengeBinaryDataResult {
            SceUInt64 inboxId;
            MessageAttachment attachment;
         } ChallengeBinaryDataResult;
      }
   }
}
```

Members

inboxId	The game custom data inbox ID.
attachment	The requested attachment.

Description

Represents the result of a request to download the data attachment of a challenge.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ChallengeRecipientList

Defines a list of users to send a challenge to.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef std::vector< SceNpOnlineId > ChallengeRecipientList;
      }
   }
}
```

Description

Defines a list of users to send a challenge to.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ChallengeStatus

Defines the possible status of a challenge.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef enum sce::Toolkit::NP::ChallengeStatus {
            Challenge = 1,
            ChallengeAccepted,
            ChallengeFailed,
            ChallengeSucceeded
         } ChallengeStatus;
      }
   }
}
```

Enumeration Values

Macro	Value	Description
`Challenge`	`1`	This is a challenge.
`ChallengeAccepted`	N/A	The challenge was accepted.
`ChallengeFailed`	N/A	The challenge was failed.
`ChallengeSucceeded`	N/A	The challenge was successfully completed.

Description

Defines the possible status of a challenge.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

CharPointerList

Defines a list of character pointers.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::list< char * > CharPointerList;
        }
    }
}
```

Description

Defines a list of character pointers.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

CommunicationIdList

A list of communication IDs.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::list< CommunicationId > CommunicationIdList;
        }
    }
}
```

Description

A list of communication IDs.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ConsumeChallengeResult

Represents the result of consuming a challenge data attachment.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::ConsumeChallengeResult {
            SceUInt64 inboxId;
         } ConsumeChallengeResult;
      }
   }
}
```

Members

inboxId	The game custom data inbox ID.

Description

Represents the result of consuming a challenge data attachment.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

CountryInfo

Contains a user's country information (country code and language).

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::CountryInfo {
            SceNpCountryCode countryCode;
            int language;
         } CountryInfo;
      }
   }
}
```

Members

countryCode	The user's country code.
language	The user's language. This is an `SCE_NP_LANG_XXX` value defined in the system utilities.

Description

Contains a user's country information (country code and language).

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

EntitlementList

Defines a list of entitlements.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef std::vector< Entitlement > EntitlementList;
      }
   }
}
```

Description

Defines a list of entitlements.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

EntitlementToConsume

Contains the details of an entitlement to consume.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::EntitlementToConsume {
                SceNpEntitlementId id;
                uint32_t consumedCount;
            } EntitlementToConsume;
        }
    }
}
```

Members

id	The ID of the entitlement to consume.
consumedCount	The amount to consume.

Description

Contains the details of an entitlement to consume.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Friend

Contains relevant friend information such as the friends NP ID.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef NpUser Friend;
        }
    }
}
```

Description

Contains relevant friend information such as the friends NP ID.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

FriendsList

Contains a list of friends.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< Friend > FriendsList;
        }
    }
}
```

Description

Contains information on all the friends of the user.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

FriendsRankInformation

Represents the ranks of a user's friends.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::FriendsRankInformation {
            SceNpScorePlayerRankData rankData
            [SCE_TOOLKIT_NP_MAX_FRIEND_LINE];
            SceNpScoreComment comment[SCE_TOOLKIT_NP_MAX_FRIEND_LINE];
            SceNpScoreGameInfo gameInfo[SCE_TOOLKIT_NP_MAX_FRIEND_LINE];
            SceNpScoreBoardId boardId;
            int numFriends;
            SceKernelSysClock updateTime;
         } FriendsRankInformation;
      }
   }
}
```

Members

rankData	An array of rank data for a user's friend list.
comment	An array of comments relating to the rank data.
gameInfo	An array of game information relating to the rank data.
boardId	The target board ID.
numFriends	The number of friends in the list.
updateTime	The timestamp.

Description

Represents the ranks of a user's friends. A `FriendsRankInformation` object is passed as an argument to `Ranking::Interface::displayFriendRank()`, and the user's friends' ranks are received via it.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

GameCustomDataGameData

Contains information about game custom data.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::GameCustomDataGameData {
            char expiredDate[SCE_TOOLKIT_NP_DATE_LENGTH+1];
            char dataName[SCE_TOOLKIT_NP_CHAR_LENGTH_128+1];
            char dataDescription[SCE_TOOLKIT_NP_CHAR_LENGTH_512+1];
            char reserved;
            SceToolkitNpAvailablePlatform availablePlatforms;
         } GameCustomDataGameData;
      }
   }
}
```

Members

expiredDate	The expiry date of the game custom data.
dataName	The name of the data.
dataDescription	The description of the data.
reserved	Reserved.
availablePlatforms	The platforms that the game custom data is available on.

Description

Contains information about game custom data.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

GameCustomDataItem

Contains information about a game custom data item.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::GameCustomDataItem {
                SceUInt64 itemId;
                SceNpOnlineId fromUser;
                char receivedDate[SCE_TOOLKIT_NP_DATE_LENGTH+1];
                char message[SCE_TOOLKIT_NP_CHAR_LENGTH_512+1];
                bool dataUsedFlag;
                char reserved;
                GameCustomDataGameData gameDataDetail;
                AttachmentDetail attachmentDetail;
            } GameCustomDataItem;
        }
    }
}
```

Members

itemId	The item ID.
fromUser	The sender's online ID.
receivedDate	The date that the item was received.
message	The message.
dataUsedFlag	The data used flag.
reserved	Reserved.
gameDataDetail	The game data object.
attachmentDetail	The custom data item's attachment details.

Description

Contains information about a game custom data item.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

GameCustomDataItemList

Defines a list of game custom data items.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< GameCustomDataItem > GameCustomDataItemList;
        }
    }
}
```

Description

Defines a list of game custom data items.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

GetInfoNpSessionRequest

Represents a request to get some information on the Session server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef NpSessionRequest GetInfoNpSessionRequest;
        }
    }
}
```

Description

Represents a request to get some information on the Session server.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

InGameDataMessage

Represents an in-game data message.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::InGameDataMessage {
                SceNpId npId;
                SceNpBasicInGameDataMessage message;
                SceNpPlatformType platformType;
            } InGameDataMessage;
        }
    }
}
```

Members

npId	The NP ID of the message recipient.
message	The in-game data message to be sent.
platformType	The platform the message is to be sent to.

Description

Represents an in-game data message. As well as the message, it includes the ID of the recipient and the platform they are on.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

InviteNpSessionRequest

Represents a request to invite a friend of a user to a session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef NpSessionRequest InviteNpSessionRequest;
        }
    }
}
```

Description

Represents a request to invite a friend of a user to a session.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

LeaveNpSessionRequest

Represents a request to leave a session on the Session server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef NpSessionRequest LeaveNpSessionRequest;
        }
    }
}
```

Description

Represents a request to leave a session on the Session server.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

LocalizedMetadata

Represents custom data and challenge localized data visible to the end user.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::LocalizedMetadata {
            char npLanguage
            [SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_NP_LANG_CODE_LEN+1];
            char name[SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_DATA_NAME_LEN+1];
            char description
            [SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_DATA_DESCRIPTION_LEN+1];
         } LocalizedMetadata;
      }
   }
}
```

Members

npLanguage	The language supported.
name	The localized data title.
description	The localized data description.

Description

Represents custom data and challenge localized data visible to the end user.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

LocalizedNpSessionName

Represents a localized session name.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::LocalizedNpSessionName {
            char npLanguage[SCE_TOOLKIT_NP_LANGUAGE_CODE_MAX_LEN+1];
            char sessionName[SCE_TOOLKIT_NP_SESSION_NAME_MAX_SIZE];
         } LocalizedNpSessionName;
      }
   }
}
```

Members

npLanguage	The language the name is in.
sessionName	The session name.

Description

Represents a localized session name.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

LocalizedNpSessionStatus

Represents localized session status.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::LocalizedNpSessionStatus {
            char npLanguage[SCE_TOOLKIT_NP_LANGUAGE_CODE_MAX_LEN+1];
            char sessionStatus[SCE_TOOLKIT_NP_SESSION_STATUS_MAX_SIZE];
         } LocalizedNpSessionStatus;
      }
   }
}
```

Members

npLanguage	The language the session status is in.
sessionStatus	The session status.

Description

Represents localized session status.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

MemberAddress

Provides address information about the member.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::MemberAddress {
            SceNetInAddr addr;
            SceUShort16 port;
            SceToolkitNpSessionSignalingFlag flag;
            SceChar8 padding;
         } MemberAddress;
      }
   }
}
```

Members

addr	The IP address.
port	The port number.
flag	The signaling flag.
padding	Padding, which aligns the structure to 4 bytes.

Description

Provides address information about the member.

Notes

To retrieve this information, the session has to be created with the `SCE_TOOLKIT_NP_CREATE_SIGNALING_MESH_SESSION` flag set.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

MultiMapCommIdServiceType

A map demonstrating linkage between the service type and the communication ID.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::multimap< CommunicationId, ServiceType >
            MultiMapCommIdServiceType;
        }
    }
}
```

Description

A map demonstrating linkage between the service type and the communication ID.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

MultiMapServiceIdServiceType

Defines a map demonstrating linkage between the service ID and service type.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::multimap< ServiceId, ServiceType >
            MultiMapServiceIdServiceType;
        }
    }
}
```

Description

Defines a map demonstrating linkage between the service ID and service type.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

MultiMapServiceLabelServiceType

Defines a map demonstrating linkage between the service label and the service type.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef std::multimap< SceNpServiceLabel, ServiceType >
         MultiMapServiceLabelServiceType;
      }
   }
}
```

Description

Defines a map demonstrating linkage between the service label and the service type.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NearDiscoveredGiftData

Represents gift data.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::NearDiscoveredGiftData {
                SceNearGiftDiscoveringId discoveringGiftId;
                SceUInt32 dataSize;
                const NearGiftData *pGiftData;
            } NearDiscoveredGiftData;
        }
    }
}
```

Members

discoveringGiftId	The discovering ID of the gift.
dataSize	The size of the gift data.
pGiftData	The buffer containing the gift data retrieved.

Description

Represents gift data. The struct is used by `Near::Interface::getGiftData()` and can also be used on received gifts.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NearDiscoveredGiftDetails

Represents information about a discovered gift.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::NearDiscoveredGiftDetails {
                SceNearDiscoveredGiftStatus giftStatus;
                SceNpId giftSender;
                SceNearGiftInfo giftInfo;
            } NearDiscoveredGiftDetails;
        }
    }
}
```

Members

giftStatus	The status of the gift.
giftSender	The `SceNpId` of the sender of the gift.
giftInfo	Information about the gift such as its name and description.

Description

Represents information about a discovered gift. This includes the status of the gift, who sent the gift, the gift's name and its description. The struct is used by `Near::Interface::getGiftDetails()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NearDiscoveredGiftImage

Represents a gift image.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::NearDiscoveredGiftImage {
            SceNearGiftDiscoveringId discoveringGiftId;
            SceUInt32 imageSize;
            const void *pImageBuffer;
         } NearDiscoveredGiftImage;
      }
   }
}
```

Members

discoveringGiftId	The discovering ID of the gift.
imageSize	The size of the gift image.
pImageBuffer	The buffer containing the image retrieved.

Description

Represents a gift image. The struct is used by `Near::Interface::getGiftImage()` and can also be used on discovered gifts and received gifts.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NearGiftData

Represents a gift.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::NearGiftData {
                NearGiftDataHeader header;
                SceUInt8 body[SCE_TOOLKIT_NP_MAX_GIFT_BODY_SIZE];
            } NearGiftData;
        }
    }
}
```

Members

header	The header of the gift.
body	The body of the gift.

Description

Represents a gift. It is split into 2 sections, which represent the gift's header and the gift itself. The header is 256 bytes while the body is 102144 bytes.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NearGiftDataHeader

Represents a gift's header information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::NearGiftDataHeader {
                SceUInt32 magicNumber;
                NearGiftType giftType;
                SceUInt32 giftId;
                SceUInt32 numberOfUsers;
                char npId[10][SCE_TOOLKIT_NP_MAX_ONLINEID];
                SceUInt8 reserved[70];
            } NearGiftDataHeader;
        }
    }
}
```

Members

magicNumber	The gift creation identifier, which identities that this gift was created using NP Toolkit.
giftType	The type of the gift.
giftId	The ID of the gift.
numberOfUsers	The number of users who has received this gift and then passed it on.
npId	An array of the 10 `SceNpId`s, which represents the last 10 users who have received this gift and then decided to pass it on.
reserved	For future expansion. Currently functions as padding.

Description

Represents a gift's header information. The first 256 bytes of a gift are reserved for the header. NP Toolkit uses this space to store some useful metadata about the gift such as its type, its ID and the last 10 users who received this gift and then passed it on. The magic number is used to identify that the gift was created using NP Toolkit as opposed to a gift that is not compatible with NP Toolkit's "near" service.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NearGiftInputParam

Represents the parameters that gift creation requires.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::NearGiftInputParam {
            const char *iconPath;
            const char *dataPath;
            const char *name;
            const char *description;
            NearGiftType giftType;
            SceNearGiftId giftId;
            SceNearGiftCondition giftCondition;
            SceUInt32 giftUnits;
         } NearGiftInputParam;
      }
   }
}
```

Members

iconPath	The path of the icon to be used for the gift.
dataPath	The path of the data to be used for the gift data.
name	The name of the gift. This is held in a string of up to `SCE_NEAR_GIFT_NAME_MAX_LENGTH` in length.
description	The description of the gift. This is held in a string of up to `SCE_NEAR_GIFT_DESCRIPTION_MAX_LENGTH` in length.
giftType	The type of the gift.
giftId	The gift ID.
giftCondition	The condition of the gift.
giftUnits	The number of units of this gift to be distributed before it is deleted from the server.

Description

Represents the parameters that gift creation requires. These are passed into `Near::Interface::createGift()`.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

NearGiftType

Defines the various types of gift.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef enum sce::Toolkit::NP::NearGiftType {
                nearGiftDefaultType = 0,
                nearGiftExchangeType,
                nearGiftAdvertisementType,
                nearGiftPowerCollectionType,
                nearGiftUnlockingType,
                nearGiftChallengeType
            } NearGiftType;
        }
    }
}
```

Enumeration Values

Macro	Value	Description
`nearGiftDefaultType`	0	The default type. Undefined. These can be normal item gifts.
`nearGiftExchangeType`	N/A	This type of gift requires that the user sends some item back to the sender before they are able to use the received gift.
`nearGiftAdvertisementType`	N/A	This type of gift just broadcasts some in-game information to nearby users.
`nearGiftPowerCollectionType`	N/A	This type of gift must be collected many times before some sort of in-game rewards are earned.
`nearGiftUnlockingType`	N/A	This type of gift just holds a key to already existing game content.
`nearGiftChallengeType`	N/A	This type of gift contains an invitation for nearby users.

Description

Defines the various types of gift. Gifts can be divided into different types by the application depending on their usage. These types does not affect usage of "near" service internally. The purpose is to give gift types so that the game application can determine the gift's behavior depending on the assigned type. Internally this could be used to expand the "near" service.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NearNeighbors

Represents nearby users information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::NearNeighbors {
                SceNpId neighbors[SCE_TOOLKIT_NP_MAX_NUM_NEIGHBORS];
                unsigned int arraySize;
            } NearNeighbors;
        }
    }
}
```

Members

neighbors	An array containing the `SceNpId`s of the returned nearby users.
arraySize	The number of returned nearby users.

Description

Represents nearby users information. The struct contains the number of nearby users retrieved and an array of the nearby users' `SceNpId`s.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

NearRelayGiftParam

Represents information used in relaying a gift.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::NearRelayGiftParam {
            SceNearGiftDiscoveringId discoveringGiftId;
            SceNearGiftCondition giftCondition;
            SceUInt32 giftUnits;
         } NearRelayGiftParam;
      }
   }
}
```

Members

discoveringGiftId	The discovering ID of the gift to be relayed.
giftCondition	The condition under which the gift can be relayed.
giftUnits	The units of the gift to be relayed.

Description

Represents information used in relaying a gift. The struct is used when calling `Near::Interface::relayGift`. After a gift has been received, an instance of this struct can be used to define the conditions necessary for the gift to be relayed.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NeighborType

Defines the types of nearby user that the application can retrieve.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef enum sce::Toolkit::NP::NeighborType {
                sceNpToolkitNPNeighborDefault = 0,
                sceNpToolkitNPNeighborRecent,
                sceNpToolkitNPNeighborNew
            } NeighborType;
        }
    }
}
```

Enumeration Values

Macro	Value	Description
`sceNpToolkitNPNeighborDefault`	0	The default type. Retrieves up to 100 nearby users, but does not take into consideration whether they are recent or not.
`sceNpToolkitNPNeighborRecent`	N/A	Retrieves nearby users who were discovered by recent updates, but includes nearby users who were discovered before as well.
`sceNpToolkitNPNeighborNew`	N/A	Retrieves nearby users who were discovered with the recent updates, but excludes nearby users who were discovered before.

Description

Defines the types of nearby users that the application can retrieve. Depending on the type specified, the “near” service will retrieve certain types of nearby users.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NetStateBasic

Contains basic network information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::NetStateBasic {
                int connectionStatus;
                char ipAddress[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
                SceNetCtlNatInfo natInfo;
            } NetStateBasic;
        }
    }
}
```

Members

connectionStatus	The connection status. This maps to `SCE_NET_CTL_STATE_XXX`.
ipAddress	The IP address of the network adapter.
natInfo	The NAT type.

Description

Contains basic network information. It includes only the most useful bits of information.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NetStateDetailed

Contains detailed network information.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::NetStateDetailed {
            int connectionStatus;
            SceNetCtlNatInfo natInfo;
            unsigned int device;
            SceNetEtherAddr ethernetAddress;
            unsigned char rssi;
            unsigned char channel;
            unsigned int mtu;
            unsigned int link;
            SceNetEtherAddr bssid;
            char ssid[SCE_NET_CTL_SSID_LEN];
            SceUInt8 padding[1];
            unsigned int wlanSecurity;
            unsigned int ipConfig;
            char dhcpHostname[SCE_NET_CTL_HOSTNAME_LEN];
            char pppoeAuthName[SCE_NET_CTL_AUTH_NAME_LEN];
            char ipAddress[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
            char netmask[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
            char defaultRoute[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
            char primaryDNS[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
            char secondaryDNS[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
            unsigned int httpProxyConfig;
            char httpProxyServer[SCE_NET_CTL_HOSTNAME_LEN];
            unsigned int httpProxyPort;
         } NetStateDetailed;
      }
   }
}
```

Members

connectionStatus	The connection status. This maps to `SCE_NET_CTL_STATE_XXX`.
natInfo	The NAT type.
device	The network device being used.
ethernetAddress	The MAC address.
rssi	The signal strength.
channel	The wireless channel used.
mtu	MTU.
link	The link connection state.
bssid	BSSID.
ssid	SSID.
padding	Padding.
wlanSecurity	Specifies whether wireless LAN is encrypted.
ipConfig	Specifies how the IP address is configured.
dhcpHostname	The DHCP hostname.
pppoeAuthName	The PPPOE authentication name.
ipAddress	The device’s IP address.
netmask	The device’s Net mask.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

defaultRoute	The default route IP address.
primaryDNS	The primary domain name server IP address.
secondaryDNS	The secondary domain name server IP address.
httpProxyConfig	The configuration of the proxy server.
httpProxyServer	The IP address of the proxy.
httpProxyPort	The proxy server port address.

Description

Contains detailed network information. Most of this information would only ever be useful during debugging.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NotifyChallengeResult

Represents the result of sending a challenge response.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::NotifyChallengeResult {
                SceUInt64 inboxId;
            } NotifyChallengeResult;
        }
    }
}
```

Members

inboxId	Game Custom Data Inbox ID of the challenge that was responded to.

Description

Represents the result of sending a challenge response.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NpSessionInformation

Represents information about an NP Session.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::NpSessionInformation {
            SceNpSessionId npSessionId;
            SceToolkitNpAvailablePlatform platform;
            int32_t errorCode;
         } NpSessionInformation;
      }
   }
}
```

Members

npSessionId	The session ID related to the Session server.
platform	The platform the session is on.
errorCode	An error code if this particular session failed to register on the Session server.

Description

Represents information about an NP Session.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

NpSessionInvitationInfo

Represents information about an invitation.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::NpSessionInvitationInfo {
            char receivedDate[SCE_TOOLKIT_NP_DATE_LENGTH+1];
            bool usedFlag;
            bool expired;
            char message[512];
            SceNpOnlineId fromUser;
            NpSessionDetailedInformation session;
            SceNpInvitationId invitationId;
            SceToolkitNpAvailablePlatform availablePlatforms;
         } NpSessionInvitationInfo;
      }
   }
}
```

Members

receivedDate	The date that the invitation was received.
usedFlag	A flag that specifies whether the invitation has been used.
expired	A flag that specifies whether the invitation has expired.
message	The message with the invitation.
fromUser	The user that the invitation is from.
session	Detailed information about the session.
invitationId	The invitation ID.
availablePlatforms	The platforms the session is available on.

Description

Represents information about an invitation.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

NpSessionInvitationInfoList

Represents a list of invitations.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef std::vector< NpSessionInvitationInfo >
         NpSessionInvitationInfoList;
      }
   }
}
```

Description

Represents a list of invitations.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NpSessionMember

Represents information about a member of an NP Session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::NpSessionMember {
                SceNpOnlineId onlineId;
                SceToolkitNpAvailablePlatform platform;
            } NpSessionMember;
        }
    }
}
```

Members

onlineId	The online ID of the member.
platform	The platform the user is currently playing on.

Description

Represents information about a member of an NP Session.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NpSessionMemberList

Represents a list of members in a session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< NpSessionMember > NpSessionMemberList;
        }
    }
}
```

Description

Represents a list of members in a session.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NpSessionsList

Represents a list of NP Sessions.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< NpSessionInformation > NpSessionsList;
        }
    }
}
```

Description

Represents a list of NP Sessions.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NpToolkitCallback

The type of function that should be passed as an event callback to the `Interface`.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef void (*NpToolkitCallback)(
                const Event & event
            );
        }
    }
}
```

Arguments

event	The `Event` the callback function has been called for.

Return Values

None

Description

The type of function that should be passed as a callback to the `Interface` for the return of event codes. These events can then be acted on by application code as and when needed. See the particular `Event` or service you require for further definition.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NpToolkitCallback2

The type of function that should be passed as an event callback to the `Interface`.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef void (*NpToolkitCallback2)(
                const Event & event,
                void *appData
            );
        }
    }
}
```

Arguments

event	The `Event` the callback function has been called for.
appData	A pointer to the application data which is returned when the event callback is called.

Return Values

None

Description

The type of function that should be passed as a callback to the `Interface` for the return of event codes. These events can then be acted on by application code as and when needed. See the particular `Event` or service you require for further definition.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

NpUser

Contains an NP user's profile information.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::NpUser {
            SceNpId npid;
            AvatarUrl avatarUrl;
            UserCountryInfo regionInfo;
            PresenceInfo presence;
         } NpUser;
      }
   }
}
```

Members

npid	The NP ID of the user.
avatarUrl	The avatar URL for the user.
regionInfo	The country information for the user.
presence	The presence information for the user.

Description

Contains an NP user's profile information.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

OStream

Defines an output stream.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::ostream OStream;
        }
    }
}
```

Description

Defines an output stream.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ParentalControlInfo

Contains information about parental control.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::ParentalControlInfo {
            int age;
            bool isContentRestricted;
            bool isChatRestricted;
            SceChar8 padding[2];
         } ParentalControlInfo;
      }
   }
}
```

Members

age	The user's age.
isContentRestricted	A flag that specifies whether to restrict content.
isChatRestricted	A flag that specifies whether to restrict chat.
padding	Padding.

Description

Contains information about parental control.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

PhotoFb

Holds a photo and the associated data, which describes the stream story, to be posted to a Facebook user's wall.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::PhotoFb {
            String url;
            String title;
            String caption;
            String description;
         } PhotoFb;
      }
   }
}
```

Members

url	A string to hold the URL of the photo, which is the focal point of the stream story.
title	A string to hold the accompanying title of the stream story.
caption	A string to hold the accompanying caption of the stream story.
description	A string to hold the accompanying description of the stream story.

Description

Holds a photo and the associated data, which describes the stream story, to be posted to a Facebook user's wall.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

PresenceGameTitleInfo

Contains presence information about the game title currently being played.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::PresenceGameTitleInfo {
            String npTitleId;
            String npTitleName;
            String gameStatus;
            char gameData[SCE_TOOLKIT_NP_IN_GAME_PRESENCE_DATA_SIZE_MAX];
         } PresenceGameTitleInfo;
      }
   }
}
```

Members

npTitleId	The ID of the game currently being played.
npTitleName	The name of the game currently being played.
gameStatus	The status string. This is set by the game currently being played.
gameData	Application specific data. This is set by the game currently being played and is only valid if the user is in same context.

Description

Contains presence information about the game title currently being played.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

PresenceInfo

Contains the primary presence information for a user.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::PresenceInfo {
                SceNpGamePresenceStatus onlineStatus;
                PresenceGameTitleInfo gameInfo;
                String platformType;
                uint32_t presenceType;
            } PresenceInfo;
        }
    }
}
```

Members

onlineStatus	The online status of the user.
gameInfo	The presence information about the game title currently being played.
platformType	The platform type.
presenceType	A flag that indicates the presence type. Please refer to the `SCE_TOOLKIT_NP_PRESENCE_TYPE_*` flags.

Description

Contains the primary presence information for a user.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ProductInfo

Contains information about a product in the PlayStation®Store.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::ProductInfo {
            uint32_t purchasabilityFlag;
            char productId[SCE_TOOLKIT_NP_COMMERCE_PRODUCT_ID_LEN];
            char productName[SCE_TOOLKIT_NP_COMMERCE_PRODUCT_NAME_LEN];
            char shortDescription
            [SCE_TOOLKIT_NP_COMMERCE_PRODUCT_SHORT_DESCRIPTION_LEN];
            char spName[SCE_TOOLKIT_NP_COMMERCE_SP_NAME_LEN];
            char imageUrl[SCE_TOOLKIT_NP_COMMERCE_URL_LEN];
            char price[SCE_TOOLKIT_NP_SKU_PRICE_LEN];
            char padding[6];
            SceRtcTick releaseDate;
         } ProductInfo;
      }
   }
}
```

Members

purchasabilityFlag	A flag that indicates whether the product can be purchased (SCE_TOOLKIT_NP_COMMERCE_PURCHASED_XXX).
productId	The product ID.
productName	The name of the product.
shortDescription	A short description of the product.
spName	The service provider name.
imageUrl	The product image URL.
price	The price of the product. This is formatted to include the currency code.
padding	Padding.
releaseDate	The product release date.

Description

Contains information about a product in the PlayStation®Store.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

ProductInfoDetailed

Contains detailed information about a product on the PlayStation®Store.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::ProductInfoDetailed {
                uint32_t purchasabilityFlag;
                char skuId[SCE_TOOLKIT_NP_COMMERCE_SKU_ID_LEN];
                char productId[SCE_TOOLKIT_NP_COMMERCE_PRODUCT_ID_LEN];
                char productName[SCE_TOOLKIT_NP_COMMERCE_PRODUCT_NAME_LEN];
                char shortDescription
                [SCE_TOOLKIT_NP_COMMERCE_PRODUCT_SHORT_DESCRIPTION_LEN];
                char longDescription
                [SCE_TOOLKIT_NP_COMMERCE_PRODUCT_LONG_DESCRIPTION_LEN];
                char legalDescription
                [SCE_TOOLKIT_NP_COMMERCE_PRODUCT_LEGAL_DESCRIPTION_LEN];
                char spName[SCE_TOOLKIT_NP_COMMERCE_SP_NAME_LEN];
                char imageUrl[SCE_TOOLKIT_NP_COMMERCE_URL_LEN];
                char price[SCE_TOOLKIT_NP_SKU_PRICE_LEN];
                char ratingSystemId
                [SCE_TOOLKIT_NP_COMMERCE_RATING_SYSTEM_ID_LEN];
                char ratingImageUrl[SCE_TOOLKIT_NP_COMMERCE_URL_LEN];
                SceChar8 padding[2];
                SceNpCommerce2ContentRatingDescriptorList ratingDescriptors;
                SceRtcTick releaseDate;
                uint64_t intPrice;
            } ProductInfoDetailed;
        }
    }
}
```

000004892117

Members

purchasabilityFlag	A flag that indicates whether the product can be purchased (`SCE_TOOLKIT_NP_COMMERCE_PURCHASED_XXX`).
skuId	The SKU ID.
productId	The product ID.
productName	The name of the product.
shortDescription	A short description of the product.
longDescription	A long description of the product.
legalDescription	The legal description for the product.
spName	The service provider name.
imageUrl	The product image URL.
price	The price of the product. This is formatted to include the currency code.
ratingSystemId	The ID of the rating system (for example: PEGI, ESRB).
ratingImageUrl	The URL of the rating icon.
padding	Padding.
ratingDescriptors	The list of rating descriptors.
releaseDate	The product release date.
intPrice	The integer representation of the price. This is not intended for user display.

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Description

Contains detailed information about a product on the PlayStation®Store.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ProductInfoDetailedList

Defines a list of detailed information about some products.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< ProductInfoDetailed > ProductInfoDetailedList;
        }
    }
}
```

Description

Defines a list of detailed information about some products.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ProductInfoList

Defines a list of product information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< ProductInfo > ProductInfoList;
        }
    }
}
```

Description

Defines a list of product information.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

PushNotification

Represents Push Notification data about a user.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::PushNotification {
            SceNpPeerAddress to;
            SceNpPeerAddress from;
            char *pData;
            size_t pDataLength;
         } PushNotification;
      }
   }
}
```

Members

to	The ID of the user.
from	The online ID (NP ID).
pData	The data associated with Push Notification.
pDataLength	The length of the data.

Description

Represents Push Notification data about a user.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

RankInformation

Represents a range of ranks for the purpose of displaying to users.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::RankInformation {
                SceNpScoreRankData rankData[SCE_TOOLKIT_NP_MAX_RANGE];
                SceNpScoreComment comment[SCE_TOOLKIT_NP_MAX_RANGE];
                SceNpScoreGameInfo gameInfo[SCE_TOOLKIT_NP_MAX_RANGE];
                SceRtcTick lastSortDate;
                SceNpScoreRankNumber totalRecord;
                SceNpScoreBoardId boardId;
                int range;
                int start;
                int rankReturned;
                SceUInt8 padding[4];
            } RankInformation;
        }
    }
}
```

Members

rankData	An array of rank data.
comment	An array of comments relating to the rank data.
gameInfo	An array of game information relating to the rank data.
lastSortDate	The time the server created the ranking data.
totalRecord	The total number of players registered in the target scoreboard.
boardId	The ID of the ranking board.
range	The length of the list.
start	The starting index for the list.
rankReturned	The rank returned.
padding	Padding.

Description

Represents a range of ranks for the purpose of displaying to users. A `RankInformation` object is passed as an argument to `Ranking::Interface::displayRangeOfRanks()`, and the range of ranks are received via it.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

ReceivedChallengeList

Defines a list of received challenges.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< ChallengeRecvDetails > ReceivedChallengeList;
        }
    }
}
```

Description

Defines a list of received challenges.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ReceivedInGameDataMessage

Represents a received in-game data message.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::ReceivedInGameDataMessage {
            unsigned int messageId;
            SceNpId from;
            SceNpBasicInGameDataMessage message;
         } ReceivedInGameDataMessage;
      }
   }
}
```

Members

messageId	The message ID of the message to be retrieved.
from	The NP ID of the message sender.
message	The received in-game data message.

Description

Represents a received in-game data message. As well as the message, it includes the ID of the message and and the message sender.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

RegisterScore

Represents a request to register a user’s score.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef RegisterScoreRequest RegisterScore;
        }
    }
}
```

Description

Represents a request to register a user’s score.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

RegisterSessionAttribute

A structure used to register session attributes.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::RegisterSessionAttribute {
                char attribute[SCE_TOOLKIT_NP_MAX_ATTRIBUTE_LENGTH+1];
                SceToolkitNpSessionAttributeType attributeType;
                SceToolkitNpSessionAttributeValueType valueType;
                SceToolkitNpSessionAttributeMaxSize maxSize;
                SceUInt8 padding;
            } RegisterSessionAttribute;
        }
    }
}
```

Members

attribute	The name of the session attribute.
attributeType	The type of the attribute. Use either `SCE_TOOLKIT_NP_SESSION_SEARCH_ATTRIBUTE`, `SCE_TOOLKIT_NP_SESSION_EXTERNAL_ATTRIBUTE`, `SCE_TOOLKIT_NP_SESSION_INTERNAL_ATTRIBUTE` or `SCE_TOOLKIT_NP_SESSION_MEMBER_ATTRIBUTE`.
valueType	The type of the value contained in the `SessionAttributeValue` object. Use `SCE_SESSION_ATTRIBUTE_VALUE_*` flags.
maxSize	The size of the binary data in the `SessionAttributeValue` object. Use `SCE_SESSION_ATTRIBUTE_MAX_SIZE_*` flags.
padding	Padding.

Description

A structure used to register session attributes.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

RegisterSessionAttributeList

Defines a list of session attributes which need to be registered.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< RegisterSessionAttribute >
            RegisterSessionAttributeList;
        }
    }
}
```

Description

Defines a list of session attributes which need to be registered.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

RetrieveChallenges

Represents a request to retrieve previous challenges.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef sce::Toolkit::NP::ChallengeGetItemListRequest
         RetrieveChallenges;
      }
   }
}
```

Description

Represents a request to retrieve previous challenges.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

SceNpCommerce2ContentRatingDescriptorList

Defines a list of commerce to content rating descriptors.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef std::list< SceNpCommerce2ContentRatingDescriptor >
         SceNpCommerce2ContentRatingDescriptorList;
      }
   }
}
```

Description

Defines a list of commerce to content rating descriptors.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SceNpEntitlementList

Defines a list of NP entitlements.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef std::vector< SceNpEntitlement > SceNpEntitlementList;
      }
   }
}
```

Description

Defines a list of NP entitlements.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SceNpGamePresenceStatus

Defines the possible status a game can have with regards to online presence.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef enum sce::Toolkit::NP::SceNpGamePresenceStatus {
            SCE_NP_GAME_PRESENCE_STATUS_OFFLINE,
            SCE_NP_GAME_PRESENCE_STATUS_ONLINE
         } SceNpGamePresenceStatus;
      }
   }
}
```

Enumeration Values

Macro	Description
SCE_NP_GAME_PRESENCE_STATUS_OFFLINE	The game is offline.
SCE_NP_GAME_PRESENCE_STATUS_ONLINE	The game is online.

Description

Defines the possible status a game can have with regards to online presence.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

SceNpIdList

Defines a list of NP IDs.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< SceNpId > SceNpIdList;
        }
    }
}
```

Description

Defines a list of NP IDs.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SceNpTusVariableList

Defines a list of TUS variables (output).

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< SceNpTusVariable > SceNpTusVariableList;
        }
    }
}
```

Description

Defines a list of TUS variables (output).

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ServiceIdList

Defines a list of service Ids.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::list< ServiceId > ServiceIdList;
        }
    }
}
```

Description

Defines a list of service Ids.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ServiceType

Defines the different services provided by the NP Toolkit library.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef enum sce::Toolkit::NP::ServiceType {
                core = 0,
                netInfo,
                profile,
                friends,
                messaging,
                ranking,
                presence,
                tus,
                tss,
                matching,
                sns,
                commerce,
                auth,
                trophy,
                near,
                wordFilter,
                webApi,
                sessions,
                gameCustomData,
                challenges,
                size
            } ServiceType;
        }
    }
}
```

000004892117

Enumeration Values

Macro	Value	Description
`core`	0	Core services provided by the NP Toolkit library, which are not related to a particular service class.
`netInfo`	N/A	A service providing information about the network connection.
`profile`	N/A	A service providing information about the current user's profile.
`friends`	N/A	A service managing friends lists and blocked lists etc.
`messaging`	N/A	A service sending messages to other PlayStation™Network users.
`ranking`	N/A	A service managing scoreboards.
`presence`	N/A	A service providing status updates on the user's PlayStation™Network profiles.
`tus`	N/A	A service for title user storage.
`tss`	N/A	A service for title small storage.
`matching`	N/A	A service providing matchmaking for online game play.
`sns`	N/A	A service providing access to social networking services.
`commerce`	N/A	A service providing in-game commerce functionality.

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Macro	Value	Description
`auth`	N/A	A service used to retrieve a ticket from the PlayStation™Network.
`trophy`	N/A	A service for managing trophies.
`near`	N/A	A service providing "near" functionality.
`wordFilter`	N/A	A service for censoring or sanitizing comments (or singular words).
`webApi`	N/A	A service managing web API calls.
`sessions`	N/A	A service providing sessions for invitation and session servers.
`gameCustomData`	N/A	A service providing game custom data messages to other PlayStation™Network users.
`challenges`	N/A	A service for managing challenges between users.
`size`	N/A	The number of services available.

Description

Defines the different services provided by the NP Toolkit library. It is used by messages to define which service the message came from or is going to. It is also used by the `NpToolkitController` to ensure messages reach the correct service and by the `ServiceFactory` to define which services are being referred to.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SessionAttribute

Contains the information on a session attribute.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::SessionAttribute {
                char attribute[SCE_TOOLKIT_NP_MAX_ATTRIBUTE_LENGTH+1];
                SceToolkitNpSessionAttributeType attributeType;
                SceNpMatching2Operator searchOperator;
                SceToolkitNpSessionAttributeMaxSize maxSize;
                SceToolkitNpSessionAttributeValueType attributeValueType;
                char m_reserved[3];
                SessionAttributeValue attributeValue;
            } SessionAttribute;
        }
    }
}
```

Members

attribute	The name of the session attribute.
attributeType	The type of the attribute.
searchOperator	The search operator if the session attribute type is `SCE_SESSION_SEARCH_ATTRIBUTE_*`. This is used when filtering the sessions.
maxSize	The size of the data in the `SessionAttributeValue` object.
attributeValueType	The type of value contained in the `SessionAttributeValue` object. Use `SCE_SESSION_ATTRIBUTE_VALUE_*` flags.
m_Reserved	Reserved.
attributeValue	The session attribute value.

Description

Contains the information on a session attribute.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SessionAttributeList

Holds an array of session attributes.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< SessionAttribute > SessionAttributeList;
        }
    }
}
```

Description

Holds an array of session attributes.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SessionAttributeValue

Holds the value of a session attribute.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef union sce::Toolkit::NP::SessionAttributeValue {
            SceUInt32 attributeIntValue;
            char attributeBinValue[SCE_TOOLKIT_NP_ATTRIBUTE_MAX_BIN_VALUE];
         } SessionAttributeValue;
      }
   }
}
```

Members

attributeIntValue	The integer value of a session attribute.
attributeBinValue	The binary data of a session attribute.

Description

Holds the value of a session attribute.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SessionEventId

Contains the room ID and the request ID for message event.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef union sce::Toolkit::NP::SessionEventId {
                SceNpMatching2RoomId roomId;
                SceNpMatching2RequestId requestId;
            } SessionEventId;
        }
    }
}
```

Members

roomId	The room ID for which the message was received. Retrieve this value if *msgFlag* is set to SCE_TOOLKIT_NP_ROOM_MESSAGE_RECEIVED.
requestId	The request ID for which the message was sent. Retrieve this value if *msgFlag* is set to SCE_TOOLKIT_NP_ROOM_MESSAGE_SENT.

Description

Contains the room ID and the request ID for message event.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SessionEventList

This structure contains information about session events.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< SceNpMatching2Event > SessionEventList;
        }
    }
}
```

Description

This structure contains information about session events.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SessionList

Contains information about sessions.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef std::vector< SessionInformation > SessionList;
      }
   }
}
```

Description

A list of sessions.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SessionMemberList

This structure contains information about session members.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::vector< SessionMember > SessionMemberList;
        }
    }
}
```

Description

This structure contains information about session members.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SessionMessageCallback

Session message callback.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef void (*SessionMessageCallback)(
            SceNpMatching2ContextId ctxId,
            SessionMessageEventType msgEvent,
            SceNpMatching2Event event,
            const void *data
         );
      }
   }
}
```

Arguments

ctxId	The matching context.
msgEvent	The message event type.
event	The matching 2 event
data	The data received.

Return Values

None

Description

Session message callback.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SessionMessageEventType

Contains information about a type of room message event.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::SessionMessageEventType {
            SceToolkitNpRoomMessageFlag msgFlag;
            SceNpMatching2RoomMemberId srcMemberId;
            SceChar8 padding[2];
            SessionEventId eventId;
         } SessionMessageEventType;
      }
   }
}
```

Members

msgFlag	A flag that specifies whether the room message was sent or received.
srcMemberId	The member ID of the room member whose message was received. This is available only when *msgFlag* is set to SCE_TOOLKIT_NP_ROOM_MESSAGE_RECEIVED.
padding	Padding.
eventId	The union for containing the room ID and the request ID.

Description

Contains information about a type of room message event.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

SessionRequestAttribute

A structure used in a search request or when setting session attributes.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::SessionRequestAttribute {
            char attribute[SCE_TOOLKIT_NP_MAX_ATTRIBUTE_LENGTH+1];
            SceNpMatching2Operator searchOperator;
            char padding[2];
            SessionAttributeValue attributeValue;
         } SessionRequestAttribute;
      }
   }
}
```

Members

attribute	The name of the session attribute.
searchOperator	The search operator if the session attribute type is `SCE_SESSION_SEARCH_ATTRIBUTE_*`. This is used when filtering the sessions.
padding	Padding.
attributeValue	The value of the session attribute.

Description

A structure used in a search request or when setting session attributes.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

SessionSlotsInfo

Holds information about the slots present in a session.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::SessionSlotsInfo {
            SceInt16 reservedSlots;
            SceInt16 openSlots;
         } SessionSlotsInfo;
      }
   }
}
```

Members

reservedSlots	The number of slots that are reserved for a friend in a session.
openSlots	The number of slots that are open to the public.

Description

Holds information about the slots present in a session.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

String

Defines a list of string.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef std::string String;
      }
   }
}
```

Description

Defines a list of string.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

TempRank

Holds the user's registered temporary rank.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::TempRank {
            SceNpScoreRankNumber tempRank;
         } TempRank;
      }
   }
}
```

Members

tempRank	The temporary rank.

Description

Holds the user's registered temporary rank. This structure is used when there are delays in registration, and a temporary rank is passed back from the server to the client.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

TrophyGameInfo

Represents the details of a game’s trophy set.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::TrophyGameInfo {
            SceNpTrophyGameDetails gameDetail;
            SceSize size;
            const void *iconData;
         } TrophyGameInfo;
      }
   }
}
```

Members

gameDetail	Contains information on the trophy set.
size	The size of the trophy set icon.
iconData	The trophy set icon in PNG format.

Description

Represents the details of a game’s trophy set. It contains a `SceNpTrophyGameDetails` object, which holds details on the number of trophies a game has, a description of the game and a title’s name. It also contains information on the size of the trophy set icon.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

TrophyGroupInfo

Represents information on a trophy group such as the ID, grade, name, description etc.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::TrophyGroupInfo {
            SceNpTrophyGroupDetails trophyGroupDetails;
            SceNpTrophyGroupData trophyGroupData;
            SceSize size;
            const void *iconData;
         } TrophyGroupInfo;
      }
   }
}
```

Members

trophyGroupDetails	Static trophy group information such as its ID, name, description etc.
trophyGroupData	Dynamic trophy group information such as the timestamp when the trophy group was unlocked.
size	The size of the trophy group icon.
iconData	A buffer which holds the trophy group icon's data in PNG format.

Description

Represents information on a trophy group such as the ID, grade, name, description etc. Also stored is the user's status for the trophy group such as how many trophies are unlocked. It is required by the `TrophyInterface::trophyRetrieveGroups()` function.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

TrophyInfo

Represents information on a trophy such as the ID, grade, name, description etc.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::TrophyInfo {
                SceSize size;
                SceNpTrophyDetails trophyDetails;
                SceNpTrophyData trophyData;
                void *iconData;
                SceChar8 padding[4];
            } TrophyInfo;
        }
    }
}
```

Members

size	The size of the trophy icon.
trophyDetails	Static trophy information such as its ID, name, description etc.
trophyData	Dynamic trophy information such as the timestamp when the trophy was unlocked.
iconData	A buffer which holds trophy icon's data in PNG format.
padding	Padding.

Description

Represents information on a trophy such as the ID, grade, name, description etc. Also stored is the user's status on each trophy in the trophy set such as whether it is unlocked or not. It is required by the `Trophy::Interface::retrieveTrophyList()` function.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

TusDataOutput

Represents TUS (title user storage) data output.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef struct sce::Toolkit::NP::TusDataOutput {
                TusData data;
                SceNpTusDataStatus status;
            } TusDataOutput;
        }
    }
}
```

Members

data	The TUS data.
status	The status of the data.

Description

Represents TUS (title user storage) data output.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

TusVariableList

Defines a list of TUS variables (input).

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            typedef std::list< TusVariable > TusVariableList;
        }
    }
}
```

Description

Defines a list of TUS variables (input).

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

UserCountryInfo

Contains a user’s country information (country code and language).

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::UserCountryInfo {
            char countryCode[SCE_TOOLKIT_NP_COUNTRY_CODE_LEN];
            char padding[2];
            int language;
         } UserCountryInfo;
      }
   }
}
```

Members

countryCode	The user’s country code.
padding	Padding.
language	The user’s language. This is an `SCE_NP_LANG_XXX` value defined in the system utilities.

Description

Contains a user’s country information (country code and language).

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

UserInfo

Represents information about a user.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::UserInfo {
            int userId;
            SceNpId npId;
            uint32_t state;
         } UserInfo;
      }
   }
}
```

Members

userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.
npId	Not used. Added only to provide interface parity with the PlayStation®4 platform.
state	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Description

Represents information about a user.

This structure exists only to provide interface parity with the PlayStation®4 platform.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

UserRankInformation

Represents ranking information for a user.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::UserRankInformation {
            SceNpScorePlayerRankData rankData;
            SceNpScoreComment comment;
            SceNpScoreGameInfo gameInfo;
            SceNpScoreBoardId boardId;
            SceChar8 padding[4];
         } UserRankInformation;
      }
   }
}
```

Members

rankData	The user's rank information.
comment	A comment retrieved relating to the rank data.
gameInfo	A piece of game information retrieved relating to the rank data.
boardId	The target board ID.
padding	Padding. Ensures bytes are aligned to an 8-byte boundary.

Description

Represents ranking information for a user. A `UserRankInformation` object is passed as an argument to `Ranking::Interface::displayUserRank()`, and the user's ranking information is received via it.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

WordFilterParam

Represents a comment to censor or sanitize using the word filter.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::WordFilterParam {
            char comment[SCE_NP_WORD_FILTER_CENSOR_COMMENT_MAXLEN];
            SceInt32 resolveRetry;
            SceUInt32 resolveTimeout;
            SceUInt32 connTimeout;
            SceUInt32 sendTimeout;
            SceUInt32 recvTimeout;
            bool isAsync;
            char padding[3];
         } WordFilterParam;
      }
   }
}
```

Members

comment	The comment to be checked by the word filter. The maximum length of the comment is defined by `SCE_NP_SCORE_CENSOR_COMMENT_MAXLEN`.
resolveRetry	An optional timeout setting, which specifies how many name resolution retries to make.
resolveTimeout	An optional timeout setting, which specifies the timeout (in microseconds) for name resolution attempts.
connTimeout	An optional timeout setting, which specifies the timeout (in microseconds) when connecting.
sendTimeout	An optional timeout setting, which specifies the timeout (in microseconds) when sending.
recvTimeout	An optional timeout setting, which specifies the timeout (in microseconds) when receiving.
isAsync	A flag that specifies whether to process the word filtering asynchronously or synchronously.
padding	Padding.

Description

Represents a comment to censor or sanitize using the word filter.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

WordFilterSanitized

Represents a comment that has been sanitized by the word filter.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         typedef struct sce::Toolkit::NP::WordFilterSanitized {
            char sanitizedComment
            [SCE_NP_WORD_FILTER_SANITIZE_COMMENT_MAXLEN];
            size_t size;
         } WordFilterSanitized;
      }
   }
}
```

Members

sanitizedComment	The sanitized comment. The maximum length of the sanitized comment is defined by `SCE_NP_SCORE_SANITIZE_COMMENT_MAXLEN`.
size	The size of the sanitized comment.

Description

Represents a comment that has been sanitized by the word filter.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::AttachmentDetail

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::AttachmentDetail

Contains details about an attachment.

Definition

```
#include <np_toolkit.h>
struct AttachmentDetail {};
```

Description

Contains details about an attachment.

Fields

Public Instance Fields

`SceToolkitNpGameCustomDataType` *type*	The game custom data type.
`AttachmentURL` *url*	The attachment URL.

Methods Summary

Methods	Description
`AttachmentDetail`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

AttachmentDetail

The default constructor.

Definition

```
#include <np_toolkit.h>
inline AttachmentDetail();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::AttachmentURL

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::AttachmentURL

Represents the URL of an attachment.

Definition

```
#include <np_toolkit.h>
struct AttachmentURL {};
```

Description

Represents the URL of an attachment.

Fields

Public Instance Fields

char *attachmentUrl [SCE_TOOLKIT_NP_ATTACHMENT_DATA_URL_LENGTH+1]*	The attachment URL for the game custom data.
char *reserved[2]*	Reserved.
bool *withItemId*	A flag that specifies whether the item ID should be appended.

Methods Summary

Methods	Description
AttachmentURL	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

AttachmentURL

The default constructor.

Definition

```
#include <np_toolkit.h>
inline AttachmentURL();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Auth

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Auth

The namespace containing PlayStation™Network authentication functionality.

Definition

```
namespace Auth {}
```

Description

The namespace containing PlayStation™Network authentication functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Auth::Interface	External interface to the PlayStation™Network authentication functionality.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Auth::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Auth::Interface

External interface to the PlayStation™Network authentication functionality.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

External interface to the PlayStation™Network authentication functionality. This class enables a ticket to be obtained, which can be passed to an external server in order to authenticate a PlayStation™Network user.

Methods Summary

Methods	Description
getCachedTicket	Retrieves a cached ticket if one is available and valid.
getTicket	Retrieves a ticket.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

getCachedTicket

Retrieves a cached ticket if one is available and valid.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Auth {
                class Interface {
                    static int getCachedTicket(
                        sce::Toolkit::NP::Utilities::Future< Ticket > *ticket,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

ticket	A pointer to the `Future` object that will hold the ticket.
async	A flag that indicates whether the function is to be called asynchronously. Defaults to true.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to the *`ticket`* parameter.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Retrieves a cached ticket if one is available and valid. This can be passed to an external server for authentication.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getTicket

Retrieves a ticket.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Auth {
                class Interface {
                    static int getTicket(
                        sce::Toolkit::NP::Utilities::Future< Ticket > *ticket
                    );
                }
            }
        }
    }
}
```

Arguments

ticket	A pointer to the `Future` object that will hold the ticket.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to the *ticket* parameter.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Retrieves a ticket. This can be passed to an external server for authentication.

Notes

This is an asynchronous function.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: CategoryBrowseParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::CategoryBrowseParams

Contains information that is used to browse a category.

Definition

```
#include <np_toolkit.h>
struct CategoryBrowseParams : public sce::Toolkit::NP::Request {};
```

Description

Contains information that is used to browse a category.

Fields

Public Instance Fields

char *categoryId [SCE_TOOLKIT_NP_COMMERCE_CATEGORY_ID_LEN]*	The ID of the category to launch.
bool *inGame*	Used on the PlayStation®3 platform only. Added here for parity.
void **memContainer*	Used on the PlayStation®3 platform only. Added here for parity. A value of `NULL` should be specified.
SceChar8 *padding[3]*	Padding of 3 bytes. Ensures alignment to a 4-byte boundary.
uint32_t *serviceLabel*	The PlayStation®4 service label.
int *userData*	Used on the PlayStation®3 platform only. Added here for parity.

Methods Summary

Methods	Description
CategoryBrowseParams	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

CategoryBrowseParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline CategoryBrowseParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::CategoryInfo

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::CategoryInfo

Contains information about a category on the PlayStation®Store.

Definition

```
#include <np_toolkit.h>
struct CategoryInfo {};
```

Description

Contains information about a category on the PlayStation®Store.

Fields

Public Instance Fields

`uint32_t` *`countOfProducts`*	The number of products in the category.
`uint32_t` *`countOfSubCategories`*	The number of immediate subcategories in the category.
`CategoryInfoSub` *`current`*	The currently selected subcategory.
`CategoryInfoSubList` *`subCategories`*	A list of subcategories in this category.

Methods Summary

Methods	Description
`CategoryInfo`	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

CategoryInfo

The default constructor.

Definition

```
#include <np_toolkit.h>
inline CategoryInfo();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: CategoryInfoInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::CategoryInfoInputParams

Contains information that is used to retrieve information about a specific category that has been set up on the PlayStation®Store.

Definition

```
#include <np_toolkit.h>
struct CategoryInfoInputParams : public sce::Toolkit::NP::Request {};
```

Description

Contains information that is used to retrieve information about a specific category that has been set up on the PlayStation®Store.

Fields

Public Instance Fields

`char categoryId [SCE_TOOLKIT_NP_COMMERCE_CATEGORY_ID_LEN]`	The ID of the category to obtain the information about. Leave blank to get information about the root category.
`uint32_t serviceLabel`	The PlayStation®4 service label.

Methods Summary

Methods	Description
`CategoryInfoInputParams`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

CategoryInfoInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline CategoryInfoInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::CategoryInfoSub

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::CategoryInfoSub

Contains information about a subcategory in the PlayStation®Store.

Definition

```
#include <np_toolkit.h>
struct CategoryInfoSub {};
```

Description

Contains information about a subcategory in the PlayStation®Store.

Fields

Public Instance Fields

`char categoryDescription [SCE_TOOLKIT_NP_COMMERCE_CATEGORY_DESCRIPTION_LEN]`	The detailed description of the subcategory.
`char categoryId [SCE_TOOLKIT_NP_COMMERCE_CATEGORY_ID_LEN]`	The ID of the subcategory.
`char categoryName [SCE_TOOLKIT_NP_COMMERCE_CATEGORY_NAME_LEN]`	The name of the subcategory.
`char imageUrl[SCE_TOOLKIT_NP_COMMERCE_URL_LEN]`	The image URL of the subcategory.

Methods Summary

Methods	Description
`CategoryInfoSub`	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

CategoryInfoSub

The default constructor.

Definition

```
#include <np_toolkit.h>
inline CategoryInfoSub();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
ChallengeConsumeRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ChallengeConsumeRequest

Represents a request to consume a challenge.

Definition

```
#include <np_toolkit.h>
struct ChallengeConsumeRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to consume a challenge. The challenge will be marked as used, and the binary data will no longer be available.

Fields

Public Instance Fields

`SceUInt64 inboxId`	The inbox ID of the message to consume.

Methods Summary

Methods	Description
`ChallengeConsumeRequest`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ChallengeConsumeRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ChallengeConsumeRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
ChallengeGetDataRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ChallengeGetDataRequest

Represents a request to retrieve a challenge's binary data.

Definition

```
#include <np_toolkit.h>
struct ChallengeGetDataRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to retrieve a challenge's binary data.

Fields

Public Instance Fields

`SceUInt64`	*`inboxId`*	The inbox ID of the message.

Methods Summary

Methods	Description
`ChallengeGetDataRequest`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ChallengeGetDataRequest

The default constructor.

Definition

```
        #include <np_toolkit.h>
        inline ChallengeGetDataRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: ChallengeGetItemListRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ChallengeGetItemListRequest

Represents a request to retrieve previous challenges.

Definition

```
#include <np_toolkit.h>
struct ChallengeGetItemListRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to retrieve previous challenges.

Fields

Public Instance Fields

`bool` *`filterUnusable`*	Whether to remove expired and used items.
`SceUInt8` *`numChallengesToGet`*	The number of previous challenges to retrieve.
`char` *`reserved`*	Reserved.
`SceUInt8` *`typeToGet`*	The type of challenges to retrieve.

Methods Summary

Methods	Description
`ChallengeGetItemListRequest`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ChallengeGetItemListRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ChallengeGetItemListRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: ChallengeGetItemRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ChallengeGetItemRequest

Represents a request to retrieve a single item.

Definition

```
#include <np_toolkit.h>
struct ChallengeGetItemRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to retrieve a single item.

Fields

Public Instance Fields

`SceUInt64`	*`inboxId`*	The Game Custom Data item ID.

Methods Summary

Methods	Description
`ChallengeGetItemRequest`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ChallengeGetItemRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ChallengeGetItemRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::ChallengeRecvDetails

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ChallengeRecvDetails

Represents a received challenge.

Definition

```
#include <np_toolkit.h>
struct ChallengeRecvDetails {};
```

Description

Represents a received challenge.

Fields

Public Instance Fields

char *description* *[SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_DATA_DESCRIPTION_LEN+1]*	The description of the challenge conditions.
SceUInt64 *expiry*	The time the challenge expires.
SceNpOnlineId *from*	The online ID of the user that sent this challenge.
SceUInt64 *inboxId*	The ID of the message in the user's inbox.
bool *isValid*	A flag that specifies whether the challenge is valid. A valid challenge is neither used or expired.
char *name* *[SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_DATA_NAME_LEN+1]*	The name of the challenge.
SceToolkitNpAvailablePlatform *platform*	The platforms this item is available for.
char *receivedDate[SCE_TOOLKIT_NP_DATE_LENGTH+1]*	The date the challenge was received.
ChallengeStatus *status*	The status of this challenge.
char *userMessage* *[SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_USER_MESSAGE_LEN+1]*	The personal message from the challenger.

Methods Summary

Methods	Description
ChallengeRecvDetails	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ChallengeRecvDetails

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ChallengeRecvDetails();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: ChallengeResponseRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ChallengeResponseRequest

Represents a request to notify a challenge.

Definition

```
#include <np_toolkit.h>
struct ChallengeResponseRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to send a response to a challenger. Use this to notify the sender of a challenge about the status of the challenge.

Fields

Public Instance Fields

`char *data`	The binary data.
`size_t dataSize`	The size of the binary data.
`char description [SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_DATA_DESCRIPTION_LEN+1]`	The description of the challenge conditions.
`char *imagePath`	The path to the image thumbnail.
`LocalizedMetadata *localizedMetadata`	The list of supported languages.
`size_t localizedMetadataNum`	The number of supported languages.
`char name [SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_DATA_NAME_LEN+1]`	The name of the challenge.
`ChallengeRecvDetails *originalChallenge`	The original challenge that was sent.
`SceToolkitNpAvailablePlatform platform`	The available platforms for this item.
`char reserved`	Reserved.
`ChallengeStatus status`	The status of the challenge.
`char userMessage [SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_USER_MESSAGE_LEN+1]`	The personal message from the challenger.

Methods Summary

Methods	Description
ChallengeResponseRequest	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ChallengeResponseRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ChallengeResponseRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Challenges

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Challenges

The namespace containing challenges functionality.

Definition

```
namespace Challenges {}
```

Description

The namespace containing challenges functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Challenges::Interface	The interface for accessing challenges.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Challenges::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Challenges::Interface

The interface for accessing challenges.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

This interface allows the game to send, retrieve and reply to challenges.

Methods Summary

Methods	Description
consumeItem	Consumes a challenge.
getChallengeData	Retrieves a challenge's data.
getItem	Retrieves the details of a single challenge item.
getItemList	Gets a list of challenges the user has that still have valid challenge data.
sendChallenge	Sends a challenge to a user/users.
sendResponse	Replies to a challenge.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

consumeItem

Consumes a challenge.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Challenges {
                class Interface {
                    static SceInt32 consumeItem(
                        const ChallengeConsumeRequest *consumeRequest,
                        Utilities::Future< ConsumeChallengeResult > *result,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

consumeRequest	The details of the request to consume a challenge.
result	A `Future` object, which indicates the result of the request.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was kicked off successfully.
SCE_TOOLKIT_NP_FAILED_ALLOCATE	The operation failed due to lack of memory.
SCE_TOOLKIT_NP_ERROR_INVALID_POINTER	The operation failed because an invalid pointer was passed to *result*.

Description

Consumes a challenge. This operation sets the data attachment of a challenge to a used state, which makes it unavailable.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getChallengeData

Retrieves a challenge's data.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Challenges {
                class Interface {
                    static SceInt32 getChallengeData(
                        const ChallengeGetDataRequest *dataRequest,
                        Utilities::Future< ChallengeBinaryDataResult > *result,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

dataRequest	The details of the request to retrieve a challenge's data.
result	A `Future` object, which indicates the result of the request.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_FAILED_ALLOCATE`	The operation failed due to lack of memory.
`SCE_TOOLKIT_NP_ERROR_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *`result`*.

Description

Retrieves a challenge's data.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getItem

Retrieves the details of a single challenge item.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Challenges {
                class Interface {
                    static SceInt32 getItem(
                        const ChallengeGetItemRequest *getItemRequest,
                        Utilities::Future< ChallengeRecvDetails > *result,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

getItemRequest	The details of the request to retrieve a challenge.
result	A `Future` object, which indicates the result of the request.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_FAILED_ALLOCATE`	The operation failed due to lack of memory.
`SCE_TOOLKIT_NP_ERROR_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *`result`*.
`SCE_TOOLKIT_NP_CHALLENGES_NOT_A_CHALLENGE`	The operation failed because the requested item is not a challenge related item.

Description

Retrieves the details of a single challenge item. This function will retrieve a challenge item depending on the details specified in *`getItemRequest`*.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getItemList

Gets a list of challenges the user has that still have valid challenge data.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Challenges {
                class Interface {
                    static SceInt32 getItemList(
                        const ChallengeGetItemListRequest *challengesToGet,
                        Utilities::Future< ReceivedChallengeList >
                        *challengeList,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

challengesToGet	The details of the request to obtain a list of challenges.
challengeList	A `Future` object, which receives any retrieved valid challenges.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_FAILED_ALLOCATE`	The operation failed due to lack of memory.

Description

Gets a list of challenges the user has that still have valid challenge data.

Notes

Because the Future object of this function uses STL attributes, a call to the default `new` may be made when the attributes are set.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sendChallenge

Sends a challenge to a user/users.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Challenges {
                class Interface {
                    static SceInt32 sendChallenge(
                        const ChallengeSendRequest *sendDetails,
                        Utilities::Future< int > *sendResult,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

sendDetails	The details of challenge to send.
sendResult	A `Future` object, which receives the status of the sent challenge.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_FAILED_ALLOCATE	The operation failed due to lack of memory.

Description

Sends a challenge to a user/users.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sendResponse

Replies to a challenge.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Challenges {
                class Interface {
                    static SceInt32 sendResponse(
                        const ChallengeResponseRequest *notifyDetails,
                        Utilities::Future< NotifyChallengeResult > *notifyResult,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

notifyDetails	The details of the reply to the challenge.
notifyResult	A `Future` object, which receives the status of the challenge reply.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_FAILED_ALLOCATE	The operation failed due to lack of memory.

Description

Replies to a challenge. This operation sends a message to the original sender of the challenge with the status of the current user's progress.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
ChallengeSendRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ChallengeSendRequest

Represents a request to send a challenge.

Definition

```
#include <np_toolkit.h>
struct ChallengeSendRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to send a challenge.

Fields

Public Instance Fields

`char *data`	The challenge arbitrary binary data attachment.
`SceSize dataSize`	The size of the challenge's binary data.
`char description [SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_DATA_DESCRIPTION_LEN+1]`	The description of the challenge conditions.
`SceUInt16 expiresMins`	The time until the challenge expires. This should be set 0 if there is to be no expiry time.
`char *imagePath`	The path to the image thumbnail.
`LocalizedMetadata *localizedMetadata`	A list of supported languages.
`size_t localizedMetadataNum`	The number of supported languages.
`char name [SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_DATA_NAME_LEN+1]`	The title of the challenge.
`SceToolkitNpAvailablePlatform platform`	The platform for this challenge.
`SceNpOnlineId *recipients`	A list of recipients.
`size_t recipientsNum`	The number of recipients.
`char reserved[3]`	Reserved.
`char userMessage [SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_USER_MESSAGE_LEN+1]`	The personal message from the challenger.

Methods Summary

Methods	Description
ChallengeSendRequest	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ChallengeSendRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ChallengeSendRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::CheckoutInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::CheckoutInputParams

Contains a list of SKUs to add to the checkout and a memory container.

Definition

```
#include <np_toolkit.h>
struct CheckoutInputParams : public sce::Toolkit::NP::Request {};
```

Description

Contains a list of SKUs to add to the checkout and a memory container.

Fields

Public Instance Fields

`void *memContainer`	Used on the PlayStation®3 platform only. Added here for parity. A value of `NULL` should be specified.
`uint32_t serviceLabel`	The PlayStation®4 service label.
`CharPointerList skuIds`	The list of SKU IDs.

Methods Summary

Methods	Description
`CheckoutInputParams`	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

CheckoutInputParams

The default constructor.

Definition

```
        #include <np_toolkit.h>
        inline CheckoutInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Commerce

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Commerce

The namespace containing PlayStation™Network commerce functionality.

Definition

```
namespace Commerce {}
```

Description

The namespace containing PlayStation™Network commerce functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Commerce::Interface	External interface to the PlayStation™Network commerce functionality.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Commerce::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Commerce::Interface

External interface to the PlayStation™Network commerce functionality.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

This class contains the set of static methods for utilizing PlayStation™Network commerce.

Methods Summary

Methods	Description
categoryBrowse	Launches the PlayStation®Store to a specified category where the user can browse and purchase.
categoryBrowse	Launches the PlayStation®Store to a specified category where the user can browse and purchase.
categoryBrowse	Launches the PlayStation®Store to a specified category where the user can browse and purchase.
checkout	Displays the checkout dialog.
checkout	Displays the checkout dialog.
consumeEntitlement	Consumes an entitlement.
consumeEntitlement	Consumes an entitlement.
createSession	Creates a commerce session.
displayDownloadList	Displays the download list dialog.
displayDownloadList	Displays the download list dialog.
getBgdlStatus	Gets the status of additional content and full game upgrades that is downloading in the background.
getCategoryInfo	Gets category information.
getCategoryInfo	Gets category information.
getDetailedProductInfo	Gets detailed information about a product in the PlayStation®Store.
getDetailedProductInfo	Gets detailed information about a product in the PlayStation®Store.
getEntitlementList	Gets a list of service entitlements.
getEntitlementList	Gets a list of service entitlements.
getProductList	Gets a list of products that are available in the PlayStation®Store.
getProductList	Gets a list of products that are available in the PlayStation®Store.
installContent	Installs additional content or full game upgrades that have been downloaded in the background.
productBrowse	Launches the PlayStation®Store to a specified product where the user can purchase it.
voucherCodeInput	Redeems a voucher code.
voucherCodeInput	Redeems a voucher code.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

categoryBrowse

Launches the PlayStation®Store to a specified category where the user can browse and purchase.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int categoryBrowse(
                        const CategoryBrowseParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

params	Specifies the PlayStation®Store category ID. To launch to the root category, do not set a category ID. This parameter also specifies which user will be browsing.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_ERROR_OFFLINE	The operation failed because the user is not connected to the network.
SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Launches the PlayStation®Store to a specified category where the user can browse and purchase. This causes the application to terminate. When the user has quit from the store, the application will restart automatically.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

categoryBrowse

Launches the PlayStation®Store to a specified category where the user can browse and purchase.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         namespace Commerce {
            class Interface {
               static int categoryBrowse(
                  const char *categoryId
               );
            }
         }
      }
   }
}
```

Arguments

categoryId	The PlayStation®Store category ID. Specify NULL to launch to the root category.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the user is not connected to the network.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Launches the PlayStation®Store to a specified category where the user can browse and purchase. This causes the application to terminate. When the user has quit from the store, the application will re-start automatically.

Notes

This is a synchronous function.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

categoryBrowse

Launches the PlayStation®Store to a specified category where the user can browse and purchase.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         namespace Commerce {
            class Interface {
               static int categoryBrowse(
                  const char *categoryId,
                  const int &userData
               );
            }
         }
      }
   }
}
```

Arguments

categoryId	The PlayStation®Store category ID. Specify `NULL` to launch to the root category.
userData	Not used. Added only to provide interface parity with the PlayStation®3 platform. Specify a value of 0.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Launches the PlayStation®Store to a specified category where the user can browse and purchase. This causes the application to terminate. When the user has quit from the store, the application will restart automatically.

Notes

This is a synchronous function.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

checkout

Displays the checkout dialog.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int checkout(
                        CheckoutInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

params	The checkout input parameters, which includes the list of SKUs to take to the checkout.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Displays the checkout dialog to the user where they can purchase a list of SKUs.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

checkout

Displays the checkout dialog.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int checkout(
                        CharPointerList &skuIds,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

skuIds	The list of SKUs to display.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Displays the checkout dialog to the user where they can purchase a list of SKUs.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

consumeEntitlement

Consumes an entitlement.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int consumeEntitlement(
                        const ConsumeEntitlementInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

params	The entitlement and the amount to consume.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Consumes a specified amount of a consumable service entitlement.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

consumeEntitlement

Consumes an entitlement.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int consumeEntitlement(
                        const SceNpEntitlementId &id,
                        const uint32_t &consumedCount,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

id	The ID of the entitlement to consume.
consumedCount	The amount to consume.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Consumes a specified amount of a consumable service entitlement.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

createSession

Creates a commerce session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int createSession(
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Creates a commerce session. When a commerce session has been created successfully, a `commerceSessionCreated` event will be sent to the NP Toolkit callback. This session is needed only before calling `getCategoryInfo()`, `getProductList()` or `getDetailedProductInfo()`.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

displayDownloadList

Displays the download list dialog.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int displayDownloadList(
                        DownloadListInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

params	The download list input parameters, which can include a list of SKUs to show.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.

Description

Displays the download list dialog. This enables the user to redownload from a list of purchased SKUs.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

displayDownloadList

Displays the download list dialog.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int displayDownloadList(
                        CharPointerList &skuIds,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

skuIds	The list of SKUs to display in the download list. Specify an empty list to see full list of items available.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.

Description

Displays the download list dialog to the user where they can re-download from a list of purchased SKUs.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getBgdlStatus

Gets the status of additional content and full game upgrades that is downloading in the background.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int getBgdlStatus(
                        sce::Toolkit::NP::Utilities::Future
                        < SceAppUtilBgdlStatus > *status,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

status	A `Future` object, which receives the status of content that has been downloaded in the background.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the `status` pointer was invalid.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Gets the status of additional content and full game upgrades that is downloading in the background.

Document serial number: 000004892117

000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getCategoryInfo

Gets category information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int getCategoryInfo(
                        sce::Toolkit::NP::Utilities::Future
                        < CategoryInfo > *info,
                        const CategoryInfoInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

info	A `Future` object, which receives the store category information retrieved from the PlayStation®Store.
params	The category that information is being requested for. To receive information for the root category, do not specify a Category ID.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ERROR_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_ERROR_INVALID_POINTER`	The operation failed because the *`info`* pointer was invalid.
`SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Gets category information that is available in the PlayStation®Store.

Notes

Because the Future object of this function uses STL attributes, a call to default new may be made when the attributes are set.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getCategoryInfo

Gets category information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int getCategoryInfo(
                        sce::Toolkit::NP::Utilities::Future
                        < CategoryInfo > *info,
                        const char *categoryId,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

info	A `Future` object, which receives the store category information retrieved from the PlayStation®Store.
categoryId	The category that information is being requested for. Specify `NULL` to receive information for the root category.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *`info`* pointer was invalid.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Gets category information that is available in the PlayStation®Store.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getDetailedProductInfo

Gets detailed information about a product in the PlayStation®Store.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int getDetailedProductInfo(
                        sce::Toolkit::NP::Utilities::Future
                        < ProductInfoDetailed > *info,
                        const DetailedProductInfoInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

info	A `Future` object, which receives the detailed product information.
params	The ID of the product that more details are required for.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_ERROR_OFFLINE	The operation failed because the user is not connected to the network.
SCE_TOOLKIT_NP_ERROR_INVALID_POINTER	The operation failed because the *info* pointer was invalid.
SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Gets detailed information about a product in the PlayStation®Store. This includes the product long description.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getDetailedProductInfo

Gets detailed information about a product in the PlayStation®Store.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         namespace Commerce {
            class Interface {
               static int getDetailedProductInfo(
                  sce::Toolkit::NP::Utilities::Future
                  < ProductInfoDetailed > *info,
                  const char *productId,
                  const char *categoryId,
                  bool async = true
               );
            }
         }
      }
   }
}
```

Arguments

info	A `Future` object, which receives the detailed product information.
productId	The ID of the product.
categoryId	The ID of the category that the product is in. Specify `NULL` if the product is in the root category.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *`info`* pointer was invalid.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Gets detailed information about a product in the PlayStation®Store. This includes the product long description.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getEntitlementList

Gets a list of service entitlements.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int getEntitlementList(
                        sce::Toolkit::NP::Utilities::Future
                        < EntitlementList > *list,
                        const GetEntitlementsInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

list	A Future object, which receives the entitlement list.
params	The user and service label parameters.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Gets a list of service entitlements that are associated with the current PlayStation™Network user.

Notes

Because the Future object of this function uses STL attributes, a call to default new may be made when the attributes are set.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getEntitlementList

Gets a list of service entitlements.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int getEntitlementList(
                        sce::Toolkit::NP::Utilities::Future
                        < SceNpEntitlementList > *list
                    );
                }
            }
        }
    }
}
```

Arguments

list	A `Future` object, which receives the entitlement list.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.

Description

Gets a list of service entitlements that are associated with the current PlayStation™Network user.

Notes

This is an asynchronous function.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getProductList

Gets a list of products that are available in the PlayStation®Store.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int getProductList(
                        sce::Toolkit::NP::Utilities::Future
                        < ProductInfoList > *productList,
                        const ProductListInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

productList	A `Future` object, which receives the list of products that are available in the PlayStation®Store.
params	The ID of category that the products are in. Leave category ID blank to retrieve from the root category.
async	Specifies whether the function will be called synchronously or asynchronously.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ERROR_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_ERROR_INVALID_POINTER`	The operation failed because the *productList* pointer was invalid.
`SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Gets a list of products that are available in the PlayStation®Store.

Notes

Because the Future object of this function uses STL attributes, a call to default new may be made when the attributes are set.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getProductList

Gets a list of products that are available in the PlayStation®Store.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int getProductList(
                        sce::Toolkit::NP::Utilities::Future
                        < ProductInfoList > *productList,
                        const char *categoryId,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

productList	A `Future` object, which receives the list of products that are available in the PlayStation®Store.
categoryId	The ID of category that the products are in. Specify `NULL` if the products are in the root category.
async	Specifies whether the function will be called synchronously or asynchronously

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *productList* pointer was invalid.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Gets a list of products that are available in the PlayStation®Store.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

installContent

Installs additional content or full game upgrades that have been downloaded in the background.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int installContent(
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the user is not connected to the network.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Installs additional content or full game upgrades that have been downloaded in the background. A reboot is not required for installation to complete.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

productBrowse

Launches the PlayStation®Store to a specified product where the user can purchase it.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         namespace Commerce {
            class Interface {
               static int productBrowse(
                  const ProductBrowseParams &params,
                  bool async = true
               );
            }
         }
      }
   }
}
```

Arguments

params	The product browse parameters
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Launches the PlayStation®Store to a specified product where the user can purchase it. This can be either an in-game store overlay or the launch of the store after the termination of the application. In the latter case, the application will re-start automatically.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

voucherCodeInput

Redeems a voucher code.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int voucherCodeInput(
                        const VoucherInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

params	Not used. Added only to provide interface parity with the PlayStation®3 platform.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Redeems a voucher code. The function displays the voucher code input screen so a voucher/promotional code can be redeemed.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

voucherCodeInput

Redeems a voucher code.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Commerce {
                class Interface {
                    static int voucherCodeInput(
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the user is not connected to the network.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Redeems a voucher code. The function displays the voucher code input screen so a voucher/promotional code can be redeemed.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::CommunicationId

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::CommunicationId

Wraps up an NP Communication ID.

Definition

```
#include <np_toolkit.h>
class CommunicationId {};
```

Description

Wraps up an NP Communication ID. This protects against cases where the structure for these IDs varies on different platforms.

Methods Summary

Methods	Description
`CommunicationId`	A constructor that takes the components of the NP Communication ID as `string` objects.
`CommunicationId`	A constructor that takes the components of the NP Communication ID as their underlying objects.
`CommunicationId`	The default constructor for the `CommunicationId` class.
`getId`	Gets the NP Communication ID as a pointer to the wrapped data.
`getPass`	Gets the passphrase for the NP Communication ID.
`getSig`	Gets the signature for the NP Communication ID.
`operator<`	Less than comparison operator needed for some storage types.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

CommunicationId

A constructor that takes the components of the NP Communication ID as `string` objects.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         class CommunicationId {
            inline CommunicationId(
               const String &id,
               const String &pass,
               const String &sig
            );
         }
      }
   }
}
```

Arguments

id	The string representing the NP Communication ID.
pass	The string representing the passphrase for the NP Communication ID.
sig	The signature for the NP Communication ID.

Return Values

None

Description

A constructor that takes the components of the NP Communication ID as `string` objects. It copies the first 9 characters of the ID string into the data string. If the string isn't 9 characters long, it will create an empty data structure.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

CommunicationId

A constructor that takes the components of the NP Communication ID as their underlying objects.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         class CommunicationId {
            inline CommunicationId(
               const SceNpCommunicationId &id,
               const SceNpCommunicationPassphrase &pp,
               const SceNpCommunicationSignature &sig
            );
         }
      }
   }
}
```

Arguments

id	The NP Communication ID.
pp	The passphrase for the NP Communication ID.
sig	The signature for the NP Communication ID.

Return Values

None

Description

A constructor that takes the components of the NP Communication ID as their underlying objects. It copies contents directly from the parameters passed in.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

CommunicationId

The default constructor for the CommunicationId class.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class CommunicationId {
                inline CommunicationId();
            }
        }
    }
}
```

Arguments

None

Return Values

None

Description

The default constructor for the CommunicationId class. This just initializes the components of the NP Communication ID to 0 using `memset()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Operator Methods

operator<

Less than comparison operator needed for some storage types.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class CommunicationId {
                inline bool operator<(
                    const CommunicationId &rhs
                ) const;
            }
        }
    }
}
```

Arguments

rhs	The NP Communication ID to compare the stored ID against.

Return Values

If the supplied NP Communication ID is greater than the stored ID, a value of true is returned. A value of false is returned if this is not the case.

Description

Less than comparison operator needed for some storage types.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

getId

Gets the NP Communication ID as a pointer to the wrapped data.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         class CommunicationId {
            inline const SceNpCommunicationId *getId() const;
         }
      }
   }
}
```

Arguments

None

Return Values

A pointer to the NP Communication ID.

Description

Gets the NP Communication ID as a pointer to the wrapped data.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getPass

Gets the passphrase for the NP Communication ID.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class CommunicationId {
                inline const SceNpCommunicationPassphrase *getPass() const;
            }
        }
    }
}
```

Arguments

None

Return Values

The passphrase for the NP Communication ID.

Description

Gets the passphrase for the NP Communication ID.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getSig

Gets the signature for the NP Communication ID.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class CommunicationId {
                inline const SceNpCommunicationSignature *getSig() const;
            }
        }
    }
}
```

Arguments

None

Return Values

The signature for the NP Communication ID.

Description

Gets the signature for the NP Communication ID.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
ConsumeEntitlementInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ConsumeEntitlementInputParams

Contains information that is used to consume a specified amount of a consumable service entitlement for a given user.

Definition

```
#include <np_toolkit.h>
struct ConsumeEntitlementInputParams : public sce::Toolkit::NP::Request {};
```

Description

Contains information that is used to consume a specified amount of a consumable service entitlement for a given user.

Fields

Public Instance Fields

`uint32_t` *consumedCount*	The amount to consume.
`char` *entitlementId* *[SCE_TOOLKIT_NP_COMMERCE_ENTITLEMENT_ID_LEN]*	The service entitlement ID.
`uint32_t` *serviceLabel*	The PlayStation®4 service label.

Methods Summary

Methods	Description
ConsumeEntitlementInputParams	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ConsumeEntitlementInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ConsumeEntitlementInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
CreateNpSessionRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::CreateNpSessionRequest

Provides information to be displayed on the Session server.

Definition

```
#include <np_toolkit.h>
struct CreateNpSessionRequest : public sce::Toolkit::NP::Request {};
```

Description

Provides information to be displayed on the Session server.

Fields

Public Instance Fields

`SceToolkitNpAvailablePlatform` *`availablePlatforms`*	The platforms that the session is available on.
`LocalizedNpSessionName` *`*localizedSessionNames`*	The localized session names.
`LocalizedNpSessionStatus` *`*localizedSessionStatuses`*	The localized session statuses.
`SceToolkitNpSessionLockFlag` *`lockFlag`*	A flag that specifies whether the session is locked.
`int32_t` *`maxSlots`*	The maximum number of slots available for the session.
`SceToolkitNpSessionCreateFlag` *`migrationFlag`*	A flag that specifies whether the session is “owner-bind” or “owner-migration”. By default, the flag is set to “owner-bind”. Use `SCE_TOOLKIT_NP_CREATE_HOST_MIGRATION_SESSION` to specify “owner-migration”.
`uint32_t` *`numlocalizedSessionName`*	The number of the localized session names.
`uint32_t` *`numlocalizedSessionStatus`*	The number of localized session statuses.
`char` *`*sessionChangeableData`*	The Changeable Data associated with the session.
`uint32_t` *`sessionChangeableDataSize`*	The size of the Changeable Data.
`char` *`*sessionData`*	The binary data, which can be up to 1MiB in size.
`uint32_t` *`sessionDataSize`*	The size of the session data.
`char` *`sessionImgPath [SCE_TOOLKIT_NP_SESSION_IMAGE_PATH_MAX_SIZE]`*	The path of the image to upload to the Session server.
`char` *`sessionName [SCE_TOOLKIT_NP_SESSION_NAME_MAX_SIZE]`*	The session name.
`char` *`sessionStatus [SCE_TOOLKIT_NP_SESSION_STATUS_MAX_SIZE]`*	The status string which will be registered with the Session server.
`SceToolkitNpSessionTypeFlag` *`sessionTypeFlag`*	A flag that specifies whether the session is private or public.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Methods Summary

Methods	Description
CreateNpSessionRequest	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

CreateNpSessionRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline CreateNpSessionRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::CreateSessionRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::CreateSessionRequest

Provides information to be displayed on the matching server for a session.

Definition

```
#include <np_toolkit.h>
struct CreateSessionRequest : public sce::Toolkit::NP::Request {};
```

Description

Provides information to be displayed on the matching server for a session.

Fields

Public Instance Fields

`SceToolkitNpAvailablePlatform` *`availablePlatforms`*	The platforms that the session is available on.
`SceToolkitNpSessionCreateFlag` *`createSessionFlag`*	The flags to specify type of session to create. Use OR if multiple flags are required.
`int32_t` *`maxSlots`*	The maximum numbers of slots in a session.
`int32_t` *`numSessionAttributes`*	The number of session attributes.
`uint8_t` *`padding[2]`*	Padding.
`SceNpMatching2ServerId` *`serverId`*	The server ID. Used to specify if there is a specific server on which to create the session. If the value is reset an appropriate server is selected by the NP Toolkit library.
`SessionRequestAttribute` *`*sessionAttributes`*	A pointer to session attributes which needs to be set in the session.
`char` *`sessionImgPath [SCE_TOOLKIT_NP_SESSION_IMAGE_PATH_MAX_SIZE]`*	The path of the image to be uploaded to the Session server.
`char` *`sessionName [SCE_TOOLKIT_NP_SESSION_NAME_MAX_SIZE]`*	The session name.
`char` *`sessionPassword [SCE_NP_MATCHING2_SESSION_PASSWORD_SIZE]`*	The password for the session. This member is only relevant when the `SCE_TOOLKIT_NP_CREATE_PASSWORD_SESSION` flag is set.
`char` *`sessionStatus [SCE_TOOLKIT_NP_SESSION_STATUS_MAX_SIZE]`*	The status string, which will be registered with the Session server during registration.
`SceToolkitNpSessionTypeFlag` *`sessionTypeFlag`*	The flag to specify whether the session is private or public. If no flag is specified, then the *`slotsInformation`* member needs to be provided.
`SessionSlotsInfo` *`slotsInformation`*	The number of private and public players in a session. This information need not be set if *`sessionTypeFlag`* is set.
`SceNpMatching2WorldId` *`worldId`*	The world ID. Used to specify if there is a specific world in which to create the session. If the value is reset the session will be created in a random world.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Methods Summary

Methods	Description
CreateSessionRequest	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

CreateSessionRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline CreateSessionRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: DetailedProductInfoInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::DetailedProductInfoInputParams

Contains information that is used to retrieve detailed information on specific product.

Definition

```
#include <np_toolkit.h>
struct DetailedProductInfoInputParams : public sce::Toolkit::NP::Request {};
```

Description

Contains information that is used to retrieve detailed information on specific product.

Fields

Public Instance Fields

char *categoryId* *[SCE_TOOLKIT_NP_COMMERCE_CATEGORY_ID_LEN]*	The category that the product is in. This is not required on the PlayStation®4.
char *productId* *[SCE_TOOLKIT_NP_COMMERCE_PRODUCT_ID_LEN]*	The product ID.
uint32_t *serviceLabel*	The PlayStation®4 service label.

Methods Summary

Methods	Description
DetailedProductInfoInputParams	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

DetailedProductInfoInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline DetailedProductInfoInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::DetailedProductInfoListInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP:: DetailedProductInfoListInputParams

Contains information that is used to retrieve detailed information for a list of products.

Definition

```
#include <np_toolkit.h>
struct DetailedProductInfoListInputParams : public sce::Toolkit::NP::Request
{};
```

Description

Contains information that is used to retrieve detailed information for a list of products.

Fields

Public Instance Fields

ProductIdList *productIds*	The list of product IDs to retrieve the detailed information for.
uint32_t *serviceLabel*	The PlayStation®4 service label.

Methods Summary

Methods	Description
DetailedProductInfoListInputParams	The default constructor.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::DetailedProductInfoListInputParams::ProductId	Encapsulates a product ID.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Type Definitions

ProductIdList

Defines a list of product IDs.

Definition

```
#include <np_toolkit.h>
typedef std::vector< ProductId > ProductIdList;
```

Description

Defines a list of product IDs.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

DetailedProductInfoListInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline DetailedProductInfoListInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
DetailedProductInfoListInputParams::
ProductId

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP:: DetailedProductInfoListInputParams::ProductId

Encapsulates a product ID.

Definition

```
#include <np_toolkit.h>
struct ProductId {};
```

Description

Encapsulates a product ID.

Fields

Public Instance Fields

char *id[SCE_TOOLKIT_NP_COMMERCE_PRODUCT_ID_LEN]*	The product ID.
char *reserved[16]*	Reserved.

Methods Summary

Methods	Description
ProductId	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ProductId

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ProductId();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: DownloadListInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::DownloadListInputParams

Contains a list of SKUs to show in the download list to and a memory container.

Definition

```
#include <np_toolkit.h>
struct DownloadListInputParams : public sce::Toolkit::NP::Request {};
```

Description

Contains a list of SKUs to show in the download list to and a memory container.

Fields

Public Instance Fields

void *memContainer*	Used on the PlayStation®3 platform only. Added here for parity. A value of `NULL` should be specified.
uint32_t *serviceLabel*	The PlayStation®4 service label.
CharPointerList *skuIds*	The list of SKU IDs.

Methods Summary

Methods	Description
DownloadListInputParams	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

DownloadListInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline DownloadListInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Entitlement

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Entitlement

Contains information about a service entitlement.

Definition

```
#include <np_toolkit.h>
struct Entitlement {};
```

Description

Contains information about a service entitlement.

Fields

Public Instance Fields

`uint32_t` *`consumedCount`*	The amount of times a consumable service entitlement has been consumed.
`SceRtcTick` *`createdDate`*	The date when the user initially got the service entitlement.
`char` *`entitlementId [SCE_TOOLKIT_NP_COMMERCE_ENTITLEMENT_ID_LEN]`*	The service entitlement ID.
`SceRtcTick` *`expireDate`*	The date when the service entitlement expires.
`char` *`padding`*`[4]`	Padding.
`int32_t` *`remainingCount`*	The remaining uses for a consumable service entitlement. This may be a negative value.
`uint32_t` *`type`*	The type of service entitlement. These are defined by the `SCE_TOOLKIT_NP_COMMERCE_ENTITLEMENT_TYPE_XXX` flags.

Methods Summary

Methods	Description
`Entitlement`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

Entitlement

The default constructor.

Definition

```
#include <np_toolkit.h>
inline Entitlement();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Event

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Event

Represents an event generated by a service.

Definition

```
#include <np_toolkit.h>
struct Event {};
```

Description

Event objects are created by the service and then passed through the `NpToolkitController`. From the `NpToolkitController` they are propagated to the application via a registered callback. This gives the external application important information on which events have been generated.

Fields

Public Instance Fields

UserEvent *event*	The type of the event.
char *m_reserved[4]*	Reserved.
PushNotification *notificationInfo*	Additional data associated with `blockListUpdated` and `friendsListUpdated` events.
int *returnCode*	The return code of the event if applicable.
ServiceType *service*	The service this event was generated by.
sce::Toolkit::NP::Utilities::AdditionalInfo *webRequestInfo*	Provides additional information about a WebAPI request.

Methods Summary

Methods	Description
Event	The default constructor for the Event struct.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Type Definitions

UserEvent

Defines the different types of event that can be generated by the services.

Definition

```
#include <np_toolkit.h>
typedef enum sce::Toolkit::NP::Event::UserEvent {
   unknown = -1,
   enetDown = 0,
   enetUp,
   loggedIn,
   loggedOut,
   netInfoGotBandwidth,
   netInfoGotBasic,
   netInfoGotDetailed,
   netInfoError,
   netInfoDialogComplete,
   profileGotOnlineId,
   profileGotNpId,
   profileGotOnlineName,
   profileGotAvatarUrl,
   profileGotMyLanguages,
   profileGotCachedUserInfo,
   profileGotCountryInfo,
   profileGotParentalInfo,
   profileGotPlatform,
   profileError,
   friendsGotFriendsList,
   friendsNoFriends,
   friendsListReady,
   friendsListUpdated,
   friendsPresenceUpdated,
   friendsGuiExited,
   blockListReady,
   blockListUpdated,
   blockListGotInformation,
   blockListNoUser,
   presenceSet,
   presenceSetFailed,
   presenceGotInformation,
   presenceGetFailed,
   presenceGameStatusUpdated,
   presenceGameDataUpdated,
   presenceGameTitleInfoUpdated,
   rankingRangeRetrieved,
   rankingRangeRetrievedFail,
   rankingFriendsRetrieved,
   rankingFriendsRetrievedFail,
   rankingFriendsRetrievedFailNoFriends,
   rankingMatchingBoardFound,
   rankingHighScore,
   rankingServerError,
   rankingCommunityError,
   rankingScoreRegistered,
   rankingScoreRegisteredFail,
```

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

```
        rankingScoreRegisteredFailNotBest,
        rankingUserRankRetrieved,
        rankingUserRankRetrievedFail,
        rankingMemoryError,
        matchingSessionCreated,
        matchingSessionJoined,
        matchingSessionSearchCompleted,
        matchingSessionError,
        matchingSessionLeft,
        matchingSessionModified,
        matchingSessionUpdate,
        matchingSessionMessageSentError,
        npSessionCreateResult,
        npSessionJoinResult,
        npSessionError,
        npSessionLeaveResult,
        npSessionModified,
        npSessionUpdateResult,
        npSessionGetInfoResult,
        npSessionGetSessionDataResult,
        npSessionGetChangeableSessionDataResult,
        npSessionSearchResult,
        npSessionInviteNotification,
        npSessionInviteGetInfoResult,
        npSessionInviteGetInfoListResult,
        npSessionInviteGetDataResult,
        npSessionInvitePostInvitationResult,
        npSessionInviteSetDataUsedResult,
        gameCustomDataItemListResult,
        gameCustomDataGameDataResult,
        gameCustomDataMessageResult,
        gameCustomDataSetUseFlagResult,
        gameCustomDataGameThumbnailResult,
        gameCustomDataNotification,
        snsMessagePosted,
        snsDialogStarted,
        snsDialogFinished,
        snsError,
        commerceSessionCreated,
        commerceSessionAborted,
        commerceGotCategoryInfo,
        commerceGotProductList,
        commerceGotDetailedProductInfo,
        commerceProductBrowseStarted,
        commerceProductBrowseSuccess,
        commerceProductBrowseAborted,
        commerceProductBrowseFinished,
        commerceNoEntitlements,
        commerceGotEntitlementList,
        commerceConsumedEntitlement,
        commerceCheckoutStarted,
        commerceCheckoutFinished,
        commerceDownloadListStarted,
        commerceDownloadListFinished,
        commerceInstallStarted,
        commerceInstallFinished,
        commerceGotBgdlStatus,
        commerceError,
        messageSent,
        messageError,
        messageRetrieved,
```

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

```
            messageDialogTerminated,
            messageInGameDataReceived,
            messageInGameDataRetrievalDone,
            authGotTicket,
            authGotCachedTicket,
            authNewTicket,
            authError,
            trophyNotInit,
            trophyContextFail,
            trophyNotEnoughSpace,
            trophyListRetrievalSuccess,
            trophyListRetrievalFail,
            trophyPlatinumUnlocked,
            trophyUnlockSuccess,
            trophyUnlockFail,
            trophyAlreadyUnlocked,
            trophyAlreadyRegistered,
            trophyInvalidID,
            trophySetSetupSuccess,
            trophySetSetupAborted,
            trophySetSetupCancelled,
            trophySetSetupFail,
            trophyListAlreadyRetrieving,
            trophyGroupInfoRetrievalSuccess,
            trophyGroupInfoRetrievalFail,
            trophyGameInfoRetrievalSuccess,
            trophyGameInfoRetrievalFail,
            trophySetupAbortSuccess,
            trophySetupAbortFail,
            trophyBusy,
            trophyTerminated,
            trophyProgressSuccess,
            trophyProgressFail,
            trophyCacheReady,
            nearInitFailed,
            nearInitSuccess,
            nearGetMyStatusFailed,
            nearGetMyStatusSuccess,
            nearGetNeighborFailed,
            nearGetNeighborSuccess,
            nearGetGiftInfoFailed,
            nearGetGiftInfoSuccess,
            nearGetGiftImageFailed,
            nearGetGiftImageSuccess,
            nearGetGiftDataFailed,
            nearGetGiftDataSuccess,
            nearCreateRegisterGiftFailed,
            nearCreateRegisterGiftSuccess,
            nearRelayGiftFailed,
            nearRelayGiftSuccess,
            tssGotData,
            tssGotDataFromSlot,
            tssGotDataStatus,
            tssNoData,
            tssError,
            tusDataSet,
            tusDataReceived,
            tusVariablesSet,
            tusVariablesReceived,
            tusCrossSaveDataSet,
            tusCrossSaveDataReceived,
```

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

```
        tusError,
        wordFilterInitSuccess,
        wordFilterInitFailed,
        wordFilterComplete,
        challengesRetrieveListResult,
        challengesRetrieveResponsesResult,
        challengesRetrieveChallengesResult,
        challengesSendChallengeResult,
        challengesNotifyChallengeResult,
        challengesConsumeResult,
        challengesInspectItemResult,
        challengesDataRetrieveResult,
        serviceTerminate,
        serviceError
    } UserEvent;
```

Enumeration Values

Macro	Value	Description
`unknown`	`-1`	An unknown event.
`enetDown`	`0`	An event from the NetCtl service generated when the connection layer has gone down.
`enetUp`	N/A	An event from the NetCtl service generated when a connection has been established.
`loggedIn`	N/A	An event from the NetCtl service generated when a connection to the PlayStation™Network has been established.
`loggedOut`	N/A	An event from the NetCtl service generated when a connection to the PlayStation™Network has been lost.
`netInfoGotBandwidth`	N/A	An event generated when the results from the bandwidth test have been obtained.
`netInfoGotBasic`	N/A	An event generated when basic network information has been obtained.
`netInfoGotDetailed`	N/A	An event generated when detailed network information has been obtained.
`netInfoError`	N/A	An event generated when an error occurs in the net information service.
`netInfoDialogComplete`	N/A	An event generated when the log in dialog has been completed.
`profileGotOnlineId`	N/A	An event generated when the user's online ID has been retrieved.
`profileGotNpId`	N/A	An event generated when the user's NP ID has been retrieved.
`profileGotOnlineName`	N/A	An event generated when the user's online name has been retrieved.
`profileGotAvatarUrl`	N/A	An event generated when the user's avatar URL has been retrieved.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Macro	Value	Description
`profileGotMyLanguages`	N/A	An event generated when the user's PlayStation™Network languages have been retrieved.
`profileGotCachedUserInfo`	N/A	An event generated when a user's cached information has been retrieved.
`profileGotCountryInfo`	N/A	An event generated when the user's country details have been retrieved.
`profileGotParentalInfo`	N/A	An event generated when the user's parental control settings have been retrieved.
`profileGotPlatform`	N/A	An event generated when the current platform have been retrieved.
`profileError`	N/A	An event generated when if an error occurred while retrieving user profile information.
`friendsGotFriendsList`	N/A	An event generated when the user's friends list has been retrieved.
`friendsNoFriends`	N/A	An event generated if the user has no friends.
`friendsListReady`	N/A	An event generated when the user's friends list has been populated.
`friendsListUpdated`	N/A	An event generated when the user's friends list has been updated (a friend has been added/removed).
`friendsPresenceUpdated`	N/A	An event generated when the user's friends list has been updated with new presence information.
`friendsGuiExited`	N/A	An event generated when the GUI component for friends exits.
`blockListReady`	N/A	An event generated when the user's block list has been populated.
`blockListUpdated`	N/A	An event generated when the user's block list has been updated (a friend has been added/removed).
`blockListGotInformation`	N/A	An event generated when the user's block list request has been processed and it contains user/s.
`blockListNoUser`	N/A	An event generated when the user's block list contain no users.
`presenceSet`	N/A	An event generated when presence has been set successfully.
`presenceSetFailed`	N/A	An event generated when presence has not been set successfully.
`presenceGotInformation`	N/A	An event generated when the presence of a user's friend has been retrieved.
`presenceGetFailed`	N/A	An event generated when an attempt to get the presence of a user's friend failed.
`presenceGameStatusUpdated`	N/A	An event generated when the user's friends list has been updated with new presence information concerning game status.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Macro	Value	Description
presenceGameDataUpdated	N/A	An event generated when the user's friends list has been updated with new presence information concerning game data.
presenceGameTitleInfoUpdated	N/A	An event generated when the user's friends list has been updated with new presence information concerning title information.
rankingRangeRetrieved	N/A	An event generated when a range of ranks from a scoreboard has been retrieved.
rankingRangeRetrievedFail	N/A	An event generated when a range of ranks from a scoreboard failed to be retrieved.
rankingFriendsRetrieved	N/A	An event generated when the ranks and scores belonging a user's friend have been retrieved.
rankingFriendsRetrievedFail	N/A	An event generated when the rank of a user's friend could not be retrieved.
rankingFriendsRetrievedFailNoFriends	N/A	An event generated when a user has no friends so ranking service cannot perform a request.
rankingMatchingBoardFound	N/A	An event generated when a board matching the arguments was found.
rankingHighScore	N/A	An event generated when a higher score was already recorded in the cache.
rankingServerError	N/A	An event generated when an error occurred with the ranking server.
rankingCommunityError	N/A	An event generated when an error occurred communicating with the ranking server.
rankingScoreRegistered	N/A	An event generated when a ranking score has been registered.
rankingScoreRegisteredFail	N/A	An event generated when an attempt to register a ranking score has failed.
rankingScoreRegisteredFailNotBest	N/A	An event generated when an attempt to register a ranking score has failed.
rankingUserRankRetrieved	N/A	An event generated when a user's rank has been retrieved.
rankingUserRankRetrievedFail	N/A	An event generated when a user's rank failed to be retrieved.
rankingMemoryError	N/A	An event generated when the ranking service cannot allocate anymore memory for the cache.
matchingSessionCreated	N/A	An event generated when session creation has been completed.
matchingSessionJoined	N/A	An event generated when the join session process has been completed.
matchingSessionSearchCompleted	N/A	An event generated when the search process has been completed.
matchingSessionError	N/A	An event generated when there was error performing the current process.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Macro	Value	Description
`matchingSessionLeft`	N/A	An event generated when the user has left the current session.
`matchingSessionModified`	N/A	An event generated when the session has been modified.
`matchingSessionUpdate`	N/A	An event generated when the session has been updated.
`matchingSessionMessageSentError`	N/A	An event generated when the session message has been sent.
`npSessionCreateResult`	N/A	An event generated when the Np session creation process has been completed.
`npSessionJoinResult`	N/A	An event generated when the join Np session process has been completed.
`npSessionError`	N/A	An event generated when there was error performing the current Np session process.
`npSessionLeaveResult`	N/A	An event generated when the user has left the current Np session.
`npSessionModified`	N/A	An event generated when the Np session has been modified.
`npSessionUpdateResult`	N/A	An event generated when the Np session has been updated.
`npSessionGetInfoResult`	N/A	An event generated when the Np session information has been retrieved.
`npSessionGetSessionDataResult`	N/A	An event generated when the Np session data associated with the session has been retrieved.
`npSessionGetChangeableSessionDataResult`	N/A	An event generated when the Np changeable session data has been retrieved.
`npSessionSearchResult`	N/A	An event generated when the Np session search request has been completed.
`npSessionInviteNotification`	N/A	An event generated when the Np session push notification is received.
`npSessionInviteGetInfoResult`	N/A	An event generated when the Np invitation info has been retrieved.
`npSessionInviteGetInfoListResult`	N/A	An event generated when the list of Np invitation info has been retrieved.
`npSessionInviteGetDataResult`	N/A	An event generated when the Np invitation data has been retrieved.
`npSessionInvitePostInvitationResult`	N/A	An event generated when the PostInvitation has completed.
`npSessionInviteSetDataUsedResult`	N/A	An event generated when the invitation used Flag has been set.
`gameCustomDataItemListResult`	N/A	An event generated when a request to retrieve a game custom data item has been completed.
`gameCustomDataGameDataResult`	N/A	An event generated when a request to retrieve game data has been completed.
`gameCustomDataMessageResult`	N/A	An event generated when a request to retrieve a game custom data message has been completed.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Macro	Value	Description
`gameCustomDataSetUseFlagResult`	N/A	An event generated when a request to set the game custom data use flag has been completed.
`gameCustomDataGameThumbnailResult`	N/A	An event generated when a request to retrieve a thumbnail image attached to a received game custom data message has been completed.
`gameCustomDataNotification`	N/A	An event generated when a push notification related to the game custom data is received.
`snsMessagePosted`	N/A	An event generated when a message was successfully posted to Facebook.
`snsDialogStarted`	N/A	An event generated when the Facebook common dialog has been started.
`snsDialogFinished`	N/A	An event generated when the Facebook common dialog has ended.
`snsError`	N/A	An event generated when an sns specific error has occurred.
`commerceSessionCreated`	N/A	An event generated when a commerce session has successfully been created.
`commerceSessionAborted`	N/A	An event generated when the creation of commerce session has been aborted.
`commerceGotCategoryInfo`	N/A	An event generated when some category information has been retrieved from the store.
`commerceGotProductList`	N/A	An event generated when a list of products that are available has been retrieved from the store.
`commerceGotDetailedProductInfo`	N/A	An event generated when some detailed product information has been retrieved from the store.
`commerceProductBrowseStarted`	N/A	An event generated when product overlay has started.
`commerceProductBrowseSuccess`	N/A	An event generated when a product browse was completed successfully, and the user purchased the product.
`commerceProductBrowseAborted`	N/A	An event generated when a product browse was aborted by the user (the user pressed back).
`commerceProductBrowseFinished`	N/A	An event generated when a product browse has finished and it is now safe to free memory.
`commerceNoEntitlements`	N/A	An event generated when no entitlements are available for the current user.
`commerceGotEntitlementList`	N/A	An event generated when the list of entitlements has been received for the current user.
`commerceConsumedEntitlement`	N/A	An event generated when the user has successfully consumed an entitlement.
`commerceCheckoutStarted`	N/A	An event generated when a store checkout overlay has started.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Macro	Value	Description
`commerceCheckoutFinished`	N/A	An event generated when a store checkout overlay has finished.
`commerceDownloadListStarted`	N/A	An event generated when a download list overlay has started.
`commerceDownloadListFinished`	N/A	An event generated when a download list overlay has finished.
`commerceInstallStarted`	N/A	An event generated when a content install overlay has started.
`commerceInstallFinished`	N/A	An event generated when a content install overlay has finished.
`commerceGotBgdlStatus`	N/A	An event generated when the background download status has been retrieved.
`commerceError`	N/A	An event generated when a commerce error has occurred.
`messageSent`	N/A	An event generated when a message has been sent.
`messageError`	N/A	An event generated when a message failed to be received or sent.
`messageRetrieved`	N/A	An event generated when a message attachment has been retrieved.
`messageDialogTerminated`	N/A	An event generated when a message dialog box is terminated.
`messageInGameDataReceived`	N/A	An event generated when an in-game data message has been received.
`messageInGameDataRetrievalDone`	N/A	An event generated when an in-game data message has been retrieved.
`authGotTicket`	N/A	An event generated when a ticket from the PlayStation™Network has been received.
`authGotCachedTicket`	N/A	An event generated when a cached ticket from the PlayStation™Network has been received.
`authNewTicket`	N/A	An event generated when a new ticket is available.
`authError`	N/A	An event generated when an authentication error occurred.
`trophyNotInit`	N/A	An event generated when the trophy service is not initialized.
`trophyContextFail`	N/A	An event generated when there is a context error for the trophy service.
`trophyNotEnoughSpace`	N/A	An event generated when there is not enough space returned from trying to register trophy set.
`trophyListRetrievalSuccess`	N/A	An event generated when a trophy list was retrieved successfully.
`trophyListRetrievalFail`	N/A	An event generated when a trophy list failed to be retrieved.
`trophyPlatinumUnlocked`	N/A	An event generated when a platinum trophy is unlocked.
`trophyUnlockSuccess`	N/A	An event generated when a trophy was unlocked successfully.
`trophyUnlockFail`	N/A	An event generated when a trophy could not be unlocked.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Macro	Value	Description
`trophyAlreadyUnlocked`	N/A	An event generated when a trophy is already unlocked.
`trophyAlreadyRegistered`	N/A	An event generated when a trophy set is already registered.
`trophyInvalidID`	N/A	An event generated when an invalid trophy ID was passed to the trophy service.
`trophySetSetupSuccess`	N/A	An event generated when a trophy set setup was successful.
`trophySetSetupAborted`	N/A	An event generated when a trophy set setup was aborted.
`trophySetSetupCancelled`	N/A	An event generated when a trophy set setup was cancelled by the user.
`trophySetSetupFail`	N/A	An event generated when a trophy set setup failed.
`trophyListAlreadyRetrieving`	N/A	An event generated when the trophy service is already in the process of trying to retrieve another list.
`trophyGroupInfoRetrievalSuccess`	N/A	An event generated when group information retrieval was successful.
`trophyGroupInfoRetrievalFail`	N/A	An event generated when group information retrieval failed.
`trophyGameInfoRetrievalSuccess`	N/A	An event generated when trophy game information retrieval was successful.
`trophyGameInfoRetrievalFail`	N/A	An event generated when trophy game information retrieval failed.
`trophySetupAbortSuccess`	N/A	An event generated when the trophy service setup was aborted successful.
`trophySetupAbortFail`	N/A	An event generated when the trophy service setup failed to be aborted.
`trophyBusy`	N/A	An event generated when the trophy service is busy and therefore cannot process a request.
`trophyTerminated`	N/A	An event generated when the trophy service has been terminated.
`trophyProgressSuccess`	N/A	An event generated when the progress of the user has been retrieved.
`trophyProgressFail`	N/A	An event generated when the progress of the user failed to be retrieved.
`trophyCacheReady`	N/A	An event generated when the trophy cache is ready to use.
`nearInitFailed`	N/A	An event generated when the “near” service initialization failed.
`nearInitSuccess`	N/A	An event generated when the “near” service initialization succeeded.
`nearGetMyStatusFailed`	N/A	An event generated when retrieval of the user’s “near” status succeeded.
`nearGetMyStatusSuccess`	N/A	An event generated when retrieval of the user’s “near” status failed.
`nearGetNeighborFailed`	N/A	An event generated when retrieval of the user’s nearby users failed.
`nearGetNeighborSuccess`	N/A	An event generated when retrieval of the user’s nearby users succeeded.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Macro	Value	Description
`nearGetGiftInfoFailed`	N/A	An event generated when retrieval of a gift's details failed.
`nearGetGiftInfoSuccess`	N/A	An event generated when retrieval of a gift's details succeeded.
`nearGetGiftImageFailed`	N/A	An event generated when retrieval of a gift's image failed.
`nearGetGiftImageSuccess`	N/A	An event generated when retrieval of a gift's image succeeded.
`nearGetGiftDataFailed`	N/A	An event generated when retrieval of a gift's data failed.
`nearGetGiftDataSuccess`	N/A	An event generated when retrieval of a gift's data succeeded.
`nearCreateRegisterGiftFailed`	N/A	An event generated when registration of a gift failed.
`nearCreateRegisterGiftSuccess`	N/A	An event generated when registration of a gift succeeded.
`nearRelayGiftFailed`	N/A	An event generated when relay of a gift failed.
`nearRelayGiftSuccess`	N/A	An event generated when relay of a gift succeeded.
`tssGotData`	N/A	An event generated when data has been retrieved from a `TSS` (title small storage) server.
`tssGotDataFromSlot`	N/A	An event generated when data has been retrieved from a specified slot on a `TSS` (title small storage) server.
`tssGotDataStatus`	N/A	An event generated when the data status has been retrieved from a specified slot on a `TSS` (title small storage) server.
`tssNoData`	N/A	An event generated when no data is found on a `TSS` (title small storage) server.
`tssError`	N/A	An event generated when an error occurs while working with a `TSS` (title small storage) server.
`tusDataSet`	N/A	An event generated when data is uploaded to a `TUS` (title user storage) server.
`tusDataReceived`	N/A	An event generated when data is retrieved from a `TUS` (title user storage) server.
`tusVariablesSet`	N/A	An event generated when variables are set on a `TUS` (title user storage) server.
`tusVariablesReceived`	N/A	An event generated when variables are retrieved from a `TUS` (title user storage) server.
`tusCrossSaveDataSet`	N/A	An event generated when data is uploaded to a `TUS` (title user storage) server for cross-saves.
`tusCrossSaveDataReceived`	N/A	An event generated when data is retrieved from a `TUS` (title user storage) server for cross-saves.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Macro	Value	Description
`tusError`	N/A	An event generated when an error occurs while working with a TUS (title user storage) server.
`wordFilterInitSuccess`	N/A	An event generated when a word filter was initialized successfully.
`wordFilterInitFailed`	N/A	An event generated when a word filter failed to initialize.
`wordFilterComplete`	N/A	An event generated when a word filter operation has been completed.
`challengesRetrieveListResult`	N/A	An event generated when a request to retrieve items from the user's inbox has completed.
`challengesRetrieveResponsesResult`	N/A	An event generated when a request to retrieve responses from the user's inbox has completed.
`challengesRetrieveChallengesResult`	N/A	An event generated when a request to retrieve challenges from the user's inbox has completed.
`challengesSendChallengeResult`	N/A	An event generated when a request to send a challenge has been completed.
`challengesNotifyChallengeResult`	N/A	An event generated when a request to reply to a challenge has been completed.
`challengesConsumeResult`	N/A	An event generated when a request to consume a message has been completed.
`challengesInspectItemResult`	N/A	An event generated when a request to inspect a game custom data item has been completed.
`challengesDataRetrieveResult`	N/A	An event generated when a request to retrieve challenge data has been completed.
`serviceTerminate`	N/A	The event message received by the application when a request to terminate a service is processed.
`serviceError`	N/A	A generic error event related to the service specified in the service parameter of `Event` structure.

Description

Defines the different types of event that can be generated by the services.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

Event

The default constructor for the Event struct.

Definition

```
#include <np_toolkit.h>
inline Event();
```

Arguments

None

Return Values

None

Description

The default constructor for the Event struct.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SCE CONFIDENTIAL

sce::Toolkit::NP::FriendInfoRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::FriendInfoRequest

Represents a request to retrieve a list of a user's friends.

Definition

```
#include <np_toolkit.h>
struct FriendInfoRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to retrieve a list of a user's friends. Information about each retrieved friend includes their NP ID.

Fields

Public Instance Fields

`uint32_t` *`flag`*	The specific flags for the request. Please refer to the `SCE_TOOLKIT_NP_FRIENDS_LIST*` flags. The specific flags for the request. Refer to `SCE_TOOLKIT_NP_FRIENDS_LIST*`.
`SceInt` *`limit`*	The number of friends to be requested in a single call. If this is set to 0, then all friends are retrieved and *`offset`* is ignored as well.
`SceUInt32` *`offset`*	The offset into the user's friends list at which to start retrieving friends.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::FriendRankRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::FriendRankRequest

Represents a request to retrieve the ranks of a user's friends.

Definition

```
#include <np_toolkit.h>
struct FriendRankRequest : public sce::Toolkit::NP::RankingRequest {};
```

Description

Represents a request to retrieve the ranks of a user's friends.

Fields

Public Instance Fields

`SceNpScoreBoardId` *`boardId`*	The ID of the board that the ranks of a user's friends are to be retrieved from.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Friends

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Friends

The namespace containing friends list functionality.

Definition

```
namespace Friends {}
```

Description

The namespace containing friends list functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Friends::Interface	The interface for accessing the friends list.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Friends::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Friends::Interface

The interface for accessing the friends list.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

This interface allows the game to retrieve the current user's list of friends from the PlayStation™Network.

Methods Summary

Methods	Description
getBlockedUsersList	Retrieves the user's block list.
getFriendslist	Retrieves the user's friends list.
getFriendslist	Retrieves the user's friends list.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

getBlockedUsersList

Retrieves the user's block list.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Friends {
                class Interface {
                    static int getBlockedUsersList(
                        sce::Toolkit::NP::Utilities::Future
                        < BlockedList > *blockedlist,
                        const BlockedUsersInfoRequest *request,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

blockedlist	A pointer to the vector that contains all of the user's block list.
request	The information about the user's block list information is required.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Retrieves the user's block list.

If the process is asynchronous the application is notified by a `blockListGotInformation` or `blockListNoUser` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method. On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

Because the Future object of this function uses STL attributes, a call to the default `new` may be made when the attributes are set.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getFriendslist

Retrieves the user's friends list.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Friends {
                class Interface {
                    static int getFriendslist(
                        sce::Toolkit::NP::Utilities::Future
                        < FriendsList > *friendslist,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

friendslist	A pointer to the vector that contains all of the user's friends.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Retrieves the user's friends list. The results also contain presence information about each friend.

If the process is asynchronous the application is notified by a `friendsGotFriendsList` or `friendsNoFriends` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method. On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

This function has been provided for backwards compatibility. This function will return results for the initial user and return a complete list of friends. It is recommended to use the function with the `sce::Toolkit::NP::FriendInfoRequest` input parameter.

Because the Future object of this function uses STL attributes, a call to the default `new` may be made when the attributes are set.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getFriendslist

Retrieves the user’s friends list.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Friends {
                class Interface {
                    static int getFriendslist(
                        sce::Toolkit::NP::Utilities::Future
                        < FriendsList > *friendslist,
                        const FriendInfoRequest *request,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

friendslist	A pointer to the vector that contains all of the user’s friends.
request	The information about the user’s friends list that is required.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Retrieves the user’s friends list. The results also contain presence information about each friend.

If the process is asynchronous the application is notified by a `friendsGotFriendsList` or `friendsNoFriends` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method. On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

Because the Future object of this function uses STL attributes, a call to the default `new` may be made when the attributes are set.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::GameCustomData

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::GameCustomData

The namespace containing game custom data functionality.

Definition

```
namespace GameCustomData {}
```

Description

The namespace containing game custom data functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::GameCustomData::Interface	The game custom data interface class.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::GameCustomData:: Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::GameCustomData::Interface

The game custom data interface class.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

The game custom data interface class.

Methods Summary

Methods	Description
getGameData	Gets the attached data of a game custom data message.
getItemList	Gets a list of game custom data items.
getMessage	Gets a game custom data message.
getThumbnail	Gets the thumbnail image attached to a received game custom data message.
setMessageUseFlag	Sets the use flag for a game custom data message.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

getGameData

Gets the attached data of a game custom data message.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace GameCustomData {
                class Interface {
                    static int32_t getGameData(
                        const GameCustomDataGameDataRequest *request,
                        sce::Toolkit::NP::Utilities::Future< MessageAttachment >
                        *msgAttch,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

request	The details about the request to get the attached data of a game custom data message.
msgAttch	A `Future` object, which receives the attached data.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_ERROR_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *msgAttch*.

Description

Gets the attached data of a game custom data message.

When run in non-blocking mode, the function returns an appropriate error code when it not able to kick off the request. On successful completion of the process, the application is notified by a `gameCustomDataGameDataResult` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getItemList

Gets a list of game custom data items.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace GameCustomData {
                class Interface {
                    static int32_t getItemList(
                        const GameCustomDataItemsRequest *request,
                        sce::Toolkit::NP::Utilities::Future
                        < GameCustomDataItemList > *itemList,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

request	The details about the request to get a list of game custom data items.
itemList	A `Future` object, which receives the list of game custom data items.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_ERROR_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *itemList*.

Description

Gets a list of game custom data items.

When run in non-blocking mode, the function returns an appropriate error code when it not able to kick off the request. On successful completion of the process, the application is notified by a `gameCustomDataItemListResult` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Notes

Because the Future object of this function uses STL attributes, a call to the default `new` may be made when the attributes are set.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getMessage

Gets a game custom data message.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace GameCustomData {
                class Interface {
                    static int32_t getMessage(
                        const GameCustomDataMessageRequest *request,
                        sce::Toolkit::NP::Utilities::Future< GameCustomDataItem >
                        *msgAttch,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

request	The details about the request to get a game custom data message.
msgAttch	A `Future` object, which receives the message.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was kicked off successfully.
SCE_TOOLKIT_NP_ERROR_INVALID_POINTER	The operation failed because an invalid pointer was passed to *msgAttch*.

Description

Gets a game custom data message.

When run in non-blocking mode, the function returns an appropriate error code when it not able to kick off the request. On successful completion of the process, the application is notified by a `gameCustomDataMessageResult` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getThumbnail

Gets the thumbnail image attached to a received game custom data message.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace GameCustomData {
                class Interface {
                    static int32_t getThumbnail(
                        const GameCustomDataThumbnailRequest *request,
                        sce::Toolkit::NP::Utilities::Future< MessageAttachment >
                        *msgAttch,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

request	The details about the request to retrieve the thumbnail image.
msgAttch	A `Future` object, which receives the attached data.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_ERROR_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *attch*.

Description

Gets the thumbnail image attached to a received game custom data message.

When run in non-blocking mode, the function returns an appropriate error code when it not able to kick off the request. On successful completion of the process, the application is notified by a `gameCustomDataGameThumbnailResult` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setMessageUseFlag

Sets the use flag for a game custom data message.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace GameCustomData {
                class Interface {
                    static int32_t setMessageUseFlag(
                        const GameCustomDataUseFlagRequest *request,
                        sce::Toolkit::NP::Utilities::Future< int >
                        *processResult,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

request	The details about the request to set the use flag for a game custom data message.
processResult	A `Future` object, which indicates the result of the request.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_ERROR_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *processResult*.

Description

Sets the use flag for a game custom data message.

When run in non-blocking mode, the function returns an appropriate error code when it not able to kick off the request. On successful completion of the process, the application is notified by a `gameCustomDataSetUseFlagResult` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

SCE CONFIDENTIAL

sce::Toolkit::NP::
GameCustomDataGameDataRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP:: GameCustomDataGameDataRequest

Represents a request to get game custom data.

Definition

```
#include <np_toolkit.h>
struct GameCustomDataGameDataRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to get game custom data.

Fields

Public Instance Fields

`SceUInt64` *`itemId`*	The item ID of the game custom data.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: GameCustomDataItemsRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::GameCustomDataItemsRequest

Represents a request to get game custom data items.

Definition

```
#include <np_toolkit.h>
struct GameCustomDataItemsRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to get game custom data items.

Fields

Public Instance Fields

`SceInt` *`limit`*	The maximum length of the game custom data to obtain in one request.
`SceUInt32` *`offset`*	The paging offset.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: GameCustomDataMessageRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP:: GameCustomDataMessageRequest

Represents a request to get a game custom data message.

Definition

```
#include <np_toolkit.h>
struct GameCustomDataMessageRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to get a game custom data message.

Fields

Public Instance Fields

`SceUInt64` *`itemId`*	The item ID of the game custom data.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: GameCustomDataThumbnailRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP:: GameCustomDataThumbnailRequest

Contains the request to get custom data thumbnail Image.

Definition

```
#include <np_toolkit.h>
struct GameCustomDataThumbnailRequest : public sce::Toolkit::NP::Request {};
```

Description

Contains the request to get custom data thumbnail Image.

Fields

Public Instance Fields

`SceUInt64` *`itemId`*	The item ID of the game custom data.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
GameCustomDataUseFlagRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::GameCustomDataUseFlagRequest

Represents a request to set the game custom data message usage flag.

Definition

```
#include <np_toolkit.h>
struct GameCustomDataUseFlagRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to set the game custom data message usage flag.

Fields

Public Instance Fields

`SceUInt64` *`itemId`*	The item ID of the game custom data.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: GetEntitlementsInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::GetEntitlementsInputParams

Contains information that is used get the service entitlements associated with a specified user.

Definition

```
#include <np_toolkit.h>
struct GetEntitlementsInputParams : public sce::Toolkit::NP::Request {};
```

Description

Contains information that is used get the service entitlements associated with a specified user.

Fields

Public Instance Fields

`uint32_t` *`serviceLabel`*	The PlayStation®4 service label.

Methods Summary

Methods	Description
`GetEntitlementsInputParams`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

GetEntitlementsInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline GetEntitlementsInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::IdDatabase

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::IdDatabase

Manages the different IDs used inside the NP Toolkit library.

Definition

```
#include <np_toolkit.h>
class IdDatabase {};
```

Description

Manages the different IDs used inside the NP Toolkit library. This class will store multiple NP Communication IDs and multiple NP Service IDs to help represent the use of multiple services within a single application. For most cases, using one NP Communication ID and/or one NP Service ID is usual. The use of multiple NP Communication IDs tends to be for a single specific use such as regional matching.

Methods Summary

Methods	Description
`addCommsId`	Adds an NP Communication ID for a particular service type.
`addServiceId`	Adds an NP Service ID for a particular service type.
`addServiceLabel`	Adds an NP Service Label for a particular service type.
`getCommsIdsForService`	Gets a list of all NP Communication IDs used for a particular service.
`getNpServiceLabel`	Gets the NP Service Label associated with a specific service.
`getPrimaryCommsId`	Gets the main NP Communication ID assumed to be used by all services.
`getPrimaryServiceId`	Gets the NP Service ID used by the application.
`getServiceIdsForService`	Gets a list of all NP Service IDs used for a particular service.
`IdDatabase`	A constructor for the `IdDatabase` class, which takes a `CommunicationId` object as its parameter.
`~IdDatabase`	The default destructor for the `IdDatabase` class.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

IdDatabase

A constructor for the IdDatabase class, which takes a CommunicationId object as its parameter.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class IdDatabase {
                IdDatabase(
                    const CommunicationId &initialId
                );
            }
        }
    }
}
```

Arguments

initialId	A CommunicationId object.

Return Values

None

Description

A constructor for the IdDatabase class, which takes a CommunicationId object as its parameter. This is because any application using the PlayStation™Network will be using an NP Communication ID. In the future, it may be necessary to provide a Service ID equivalent constructor for games that only use ticketing or commerce.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

~IdDatabase

The default destructor for the IdDatabase class.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class IdDatabase {
                ~IdDatabase();
            }
        }
    }
}
```

Arguments

None

Return Values

None

Description

The default destructor for the IdDatabase class. Uses the default destructor of its members.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

addCommsId

Adds an NP Communication ID for a particular service type.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class IdDatabase {
                void addCommsId(
                    const CommunicationId &commsId,
                    ServiceType service
                );
            }
        }
    }
}
```

Arguments

commsId	The CommunicationId object to add.
service	The service to set it against.

Return Values

None

Description

Adds an NP Communication ID for a particular service type.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

addServiceId

Adds an NP Service ID for a particular service type.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class IdDatabase {
                void addServiceId(
                    const ServiceId servId,
                    ServiceType service
                );
            }
        }
    }
}
```

Arguments

servId	The `ServiceId` object to add.
service	The service to set it against.

Return Values

None

Description

Adds an NP Service ID for a particular service type.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

addServiceLabel

Adds an NP Service Label for a particular service type.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class IdDatabase {
                void addServiceLabel(
                    const SceNpServiceLabel &titleId,
                    ServiceType service
                );
            }
        }
    }
}
```

Arguments

titleId	The `SceNpServiceLabel` object to add.
service	The service to set it against.

Return Values

None

Description

Adds an NP Service Label for a particular service type.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getCommsIdsForService

Gets a list of all NP Communication IDs used for a particular service.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class IdDatabase {
                const CommunicationIdList getCommsIdsForService(
                    ServiceType service
                );
            }
        }
    }
}
```

Arguments

service	The service to request the IDs for.

Return Values

A list of all the NP Communication IDs used for a particular service.

Description

Gets a list of all NP Communication IDs used for a particular service.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getNpServiceLabel

Gets the NP Service Label associated with a specific service.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class IdDatabase {
                const SceNpServiceLabel getNpServiceLabel(
                    ServiceType service
                );
            }
        }
    }
}
```

Arguments

service	The service to get the label for.

Return Values

The NP Service Label used by the service.

Description

Gets the NP Service Label associated with a specific service.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getPrimaryCommsId

Gets the main NP Communication ID assumed to be used by all services.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         class IdDatabase {
            const CommunicationId getPrimaryCommsId();
         }
      }
   }
}
```

Arguments

None

Return Values

The NP Communication ID.

Description

Gets the main NP Communication ID assumed to be used by all services.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getPrimaryServiceId

Gets the NP Service ID used by the application.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class IdDatabase {
                const ServiceId getPrimaryServiceId();
            }
        }
    }
}
```

Arguments

None

Return Values

The NP Service ID used by the application.

Description

Gets the NP Service ID used by the application.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getServiceIdsForService

Gets a list of all NP Service IDs used for a particular service.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class IdDatabase {
                const ServiceIdList getServiceIdsForService(
                    ServiceType service
                );
            }
        }
    }
}
```

Arguments

service	The service to request the IDs for.

Return Values

A list of all the NP Service IDs used for a particular service.

Description

Gets a list of all NP Service IDs used for a particular service.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Interface

Contains the interface to the NP Toolkit library as a set of static methods.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

Contains the interface to the NP Toolkit library as a set of static methods. The interface wraps up the `Thread` and the `NpToolkitController` objects thereby giving the user a simple interface.

Methods Summary

Methods	Description
`init`	Initializes the NP Toolkit library.
`init`	Initializes the NP Toolkit library.
`init`	Initializes the NP Toolkit library.
`registerEventCallback`	Registers the event callback function for the NP Toolkit library. This callback returns `Event` messages to the application.
`registerEventCallback`	Registers the event callback function for the NP Toolkit library. This callback returns `Event` messages to the application.
`registerNpCommsId`	Registers an NP Communication ID against `Toolkit::NP`.
`registerServiceId`	Registers an NP Service ID against the application.
`terminate`	Terminates the NP Toolkit library.
`terminateService`	Terminates a service within the NP Toolkit library.
`unregisterEventCallback`	Unregisters the event callback function.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

init

Initializes the NP Toolkit library.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Interface {
                static int init(
                    NpToolkitCallback callback,
                    CommunicationId &idIn,
                    OStream &out = std::cout,
                    OStream &err = std::cerr
                );
            }
        }
    }
}
```

Arguments

callback	The NpToolkitCallback function pointer. Events are returned to the initializing application on this.
idIn	The CommunicationId object representing the NP Communication ID and pass-phrases provided by SCE.
out	Optional. The output stream for the standard out messages from the NP Toolkit library. Defaults to `cout`.
err	Optional. The output stream for the standard error messages from the NP Toolkit library. Defaults to `cerr`.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ALREADY_INITIALISED`	The operation failed because the NP Toolkit library has already been initialized.
`SCE_TOOLKIT_NP_INIT_START_THREAD`	The operation failed because an error occurred while trying to start a thread.

Description

Initializes the NP Toolkit library and starts the `Thread` class running. This method must be executed to start the NP Toolkit library or none of its functionality will be available. This method will block while it initializes its thread and will return an error if it is unable to do so.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

init

Initializes the NP Toolkit library.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Interface {
                static int init(
                    NpToolkitCallback2 callback,
                    CommunicationId &idIn,
                    void *appData,
                    OStream &out = std::cout,
                    OStream &err = std::cerr
                );
            }
        }
    }
}
```

Arguments

callback	The NpToolkitCallback2 function pointer. Events are returned to the initializing application on this.
idIn	The CommunicationId object representing the NP Communication ID and pass-phrases provided by SCE.
appData	The pointer to the application data which is returned when the event callback is called.
out	Optional. The output stream for the standard out messages from the NP Toolkit library. Defaults to `Cout`.
err	Optional. The output stream for the standard error messages from the NP Toolkit library. Defaults to `Cerr`.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ALREADY_INITIALISED`	The operation failed because the NP Toolkit library has already been initialized.
`SCE_TOOLKIT_NP_INIT_START_THREAD`	The operation failed because an error occurred while trying to start a thread.

Description

Initializes the NP Toolkit library and starts the `Thread` class running. This method must be executed to start the NP Toolkit library or none of its functionality will be available. This method will block while it initializes its thread and will return an error if it is unable to do so.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

init

Initializes the NP Toolkit library.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Interface {
                static int init(
                    Parameters &params,
                    bool initAppUtils = false
                );
            }
        }
    }
}
```

Arguments

params	A structure containing all the parameters for the library.
initAppUtils	A flag that specifies whether to ensure that the Application Utility Library has been initialized. Defaults to false.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ALREADY_INITIALISED`	The operation failed because the NP Toolkit library has already been initialized.
`SCE_TOOLKIT_NP_INIT_START_THREAD`	The operation failed because an error occurred while trying to start a thread.

Description

Initializes the NP Toolkit library and starts the `Thread` class running. This method must be executed to start the NP Toolkit library or none of the NP Toolkit functionality will be available. This method will block while it initializes its thread and will return an error if it is unable to do so.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

registerEventCallback

Registers the event callback function for the NP Toolkit library. This callback returns `Event` messages to the application.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Interface {
                static int registerEventCallback(
                    NpToolkitCallback func
                );
            }
        }
    }
}
```

Arguments

func	The callback function.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the pointer passed to *func* was invalid.

Description

Registers the event callback function for the NP Toolkit library, which notifies the application that an event has occurred. When the callback function is called it receives references to small `Event` objects, which allow for simple information to be passed back with the event. See the definition of `Event` for more information.

Notes

An event callback is needed in the `init()` so this function is only needed if the callback needs to change during execution.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

registerEventCallback

Registers the event callback function for the NP Toolkit library. This callback returns `Event` messages to the application.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Interface {
                static int registerEventCallback(
                    NpToolkitCallback2 func,
                    void *appData
                );
            }
        }
    }
}
```

Arguments

func	The callback function.
appData	A pointer to the application data which is returned when the event callback is called.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the pointer passed to *func* was invalid.

Description

Registers the event callback function for the NP Toolkit library, which notifies the application that an event has occurred. When the callback function is called it receives references to small `Event` objects, which allow for simple information to be passed back with the event. See the definition of `Event` for more information.

Notes

An event callback is needed in the `init()` so this function is only needed if the callback needs to change during execution.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

registerNpCommsId

Registers an NP Communication ID against Toolkit::NP.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Interface {
                static int registerNpCommsId(
                    const CommunicationId &id,
                    ServiceType service
                );
            }
        }
    }
}
```

Arguments

id	The NP Communication ID to set.
service	The type of service. Specify this if you need to use to a specific communication ID for a specific service.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful, and the Toolkit::NP thread has been correctly registered with an NP Communication ID.
`errors`	The operation failed to register the ID; errors have been returned.

Description

Registers an NP Communication ID against Toolkit::NP.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

registerServiceId

Registers an NP Service ID against the application.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Interface {
                static int registerServiceId(
                    const ServiceId &id,
                    ServiceType service
                );
            }
        }
    }
}
```

Arguments

id	The NP Service ID to register.
service	The type of service. Specify this if you need to use to a specific service ID for a specific service.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful, and the `Toolkit::NP` thread has been correctly registered with an NP Service ID.
`errors`	The operation failed to register the ID; Errors have been returned.

Description

Registers an NP Service ID against the application.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

terminate

Terminates the NP Toolkit library.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Interface {
                static int terminate();
            }
        }
    }
}
```

Arguments

None

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful and the Toolkit::NP thread has been correctly terminated.
errors	The operation failed to terminate the thread; Errors have been returned.

Description

Terminates all library functionality and returns all resources currently in use. The method will block while it terminates any running threads. It will then clean up the associated synchronization resources used. It will also free up any memory being used by NP Toolkit library at this point. The call will interrupt the event queues so no further work waiting on the queues will be processed, and the thread will quit without performing anymore processing. If the NP Toolkit library is not running, then this method does nothing.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

terminateService

Terminates a service within the NP Toolkit library.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Interface {
                static int terminateService(
                    ServiceType serviceType
                );
            }
        }
    }
}
```

Arguments

serviceType	The service to terminate.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Terminates all library functionality of the specified service and returns all resources currently in use. This is an asynchronous function and will kick off a process to terminate a service. The result of that termination process is returned through an event callback.

Notes

Not all services can be terminated. The following services can be terminated:

- Matching
- Ranking
- Trophy

On terminating certain services, you might have to initialize the service again before using its functionality unless the service does not have an initialization function specified for it.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

unregisterEventCallback

Unregisters the event callback function.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Interface {
                static int unregisterEventCallback();
            }
        }
    }
}
```

Arguments

None

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful, and the `Toolkit::NP` thread has correctly unregistered a callback.
`errors`	The operation failed to unregister the callback; errors have been returned.

Description

Unregisters the event callback function.

Notes

An event callback is needed in the arguments to `Interface`, so this function is only needed if the callback needs to change during execution.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::InvitationDataRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::InvitationDataRequest

Represents a request for invitation data.

Definition

```
#include <np_toolkit.h>
struct InvitationDataRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request for invitation data.

Fields

Public Instance Fields

`SceNpInvitationId` *`invitationId`*	The invitation ID.

Methods Summary

Methods	Description
`InvitationDataRequest`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

InvitationDataRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline InvitationDataRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::InvitationInfoRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::InvitationInfoRequest

Represents a request for information about an invitation.

Definition

```
#include <np_toolkit.h>
struct InvitationInfoRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request for information about an invitation.

Fields

Public Instance Fields

`SceNpInvitationId` *`invitationId`*	The invitation ID.

Methods Summary

Methods	Description
`InvitationInfoRequest`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

InvitationInfoRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline InvitationInfoRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::InvitationListRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::InvitationListRequest

Represents a request to retrieve an invitation list.

Definition

```
#include <np_toolkit.h>
struct InvitationListRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to retrieve an invitation list.

Fields

Public Instance Fields

`SceNpOnlineId` *`onlineId`*	The online ID of the current user associated with userID.

Methods Summary

Methods	Description
`InvitationListRequest`	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

InvitationListRequest

The default constructor.

Definition

```
        #include <np_toolkit.h>
        inline InvitationListRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: InviteJoinSessionRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::InviteJoinSessionRequest

Represents a request to join a session where the session is identified by an NP Session ID.

Definition

```
#include <np_toolkit.h>
struct InviteJoinSessionRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to join a session where the session is identified by an NP Session ID.

Fields

Public Instance Fields

`SceNpSessionId` *`npSessionId`*	The session ID related to the Session server.

Methods Summary

Methods	Description
InviteJoinSessionRequest	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

InviteJoinSessionRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline InviteJoinSessionRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::InviteMessage

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::InviteMessage

Contains the components of an invite message.

Definition

```
#include <np_toolkit.h>
struct InviteMessage {};
```

Description

Contains the components of an invite message.

Fields

Public Instance Fields

String *body*	The body text of the message.
SceToolkitNpDialogType *dialogFlag*	A flag that indicates whether the recipient list is editable by the user.
SceUInt16 *expireMinutes*	The amount of time until the expiration of the message in minutes from now. This parameter is optional for the application data attached message. A value of 0 means that the message does not expire.
String *iconPath*	The path to the icon which needs to be displayed in the message.
SceNpOnlineId **npIds*	A list of NP IDs.
size_t *npIdsCount*	The number of IDs in the *npIds* list. If *npIds* is NULL, specify the maximum number of NP IDs which can be added by the user.

Methods Summary

Methods	Description
InviteMessage	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

InviteMessage

The default constructor.

Definition

```
#include <np_toolkit.h>
inline InviteMessage();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: JoinNpSessionRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::JoinNpSessionRequest

Represents a request to join a session on the Session server.

Definition

```
#include <np_toolkit.h>
struct JoinNpSessionRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to join a session on the Session server.

Fields

Public Instance Fields

`char invitationParam[8192]`	The invitation parameter.
`bool invite`	A flag that indicates whether the join request is from an invitation.
`char m_reserved[3]`	Reserved.
`SceNpSessionId npSessionId`	The session ID related to the Session server.

Methods Summary

Methods	Description
JoinNpSessionRequest	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

JoinNpSessionRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline JoinNpSessionRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::JoinSessionRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::JoinSessionRequest

The structure which needs to be passed in order to join a session.

Definition

```
#include <np_toolkit.h>
struct JoinSessionRequest : public sce::Toolkit::NP::Request {};
```

Description

The structure which needs to be passed in order to join a session.

Fields

Public Instance Fields

SessionRequestAttribute *memberAttributes	A pointer to the member attributes that needs to be set in the session.
int32_t numSessionAttributes	The number of session attributes.
SessionInformation *sessionInformation	The information about the session that the sender of the request wishes to join.
SceNpMatching2SessionPassword *sessionPassword	The password for the session if it is password protected.

Methods Summary

Methods	Description
JoinSessionRequest	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

JoinSessionRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline JoinSessionRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::KickMemberRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::KickMemberRequest

Represents a request to kick a member out of the room.

Definition

```
#include <np_toolkit.h>
struct KickMemberRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to kick a member out of the room.

Fields

Public Instance Fields

`SceToolkitNpKickMemberFlag` *`flag`*	Specifies whether the member is allowed to rejoin or not after being kicked out (please use `SCE_TOOLKIT_NP_KICK_MEMBER_FLAG_*` flags).
`SessionMember` *`member`*	The room member to kick out.
`char` *`reserved[7]`*	Reserved.

Methods Summary

Methods	Description
`KickMemberRequest`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

KickMemberRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline KickMemberRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Matching

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Matching

The namespace containing matching functionality.

Definition

```
namespace Matching {}
```

Description

The namespace containing matching functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Matching::Interface	The matching interface allows users to find other players and game sessions for online play.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Matching::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Matching::Interface

The matching interface allows users to find other players and game sessions for online play.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

The matching interface allows users to find other players and game sessions for online play.

Methods Summary

Methods	Description
createSession	Creates a session room.
inviteToSession	Sends a session invite to a friend of the user.
joinInvitedSession	Joins the user to a session that they were invited to.
joinSession	Joins a specific session.
kickMember	Kicks a room member out of a room.
leaveSession	Leaves a currently joined/created session.
modifySession	Modifies a specific session.
quickSession	Searches for a session and join the first one available.
registerRoomMessageCallback	Registers a callback which will be called for room messages.
registerSessionAttributes	Registers attributes used by the session.
searchSessions	Searches for a session.
sendRoomMessage	Sends room/chat messages to the room members.
updateSession	Updates the current session information.
updateSessionAttribute	Updates the session attributes of the current session.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

createSession

Creates a session room.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int createSession(
                        const CreateSessionRequest *sessionRequest,
                        sce::Toolkit::NP::Utilities::Future
                        < SessionInformation > *sessionInformation,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

sessionRequest	A structure that describes the session details. This will be assigned to a room on successful creation.
sessionInformation	Output. Receives the session information upon successful creation of the session.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.
`SCE_TOOLKIT_NP_MATCHING_SERVICE_BUSY`	The operation failed because the matching service is busy processing a previous request.
`SCE_TOOLKIT_NP_MATCHING_SESSION_ALREADY_ACTIVE`	The operation failed because the user is already in a session. They must leave a session in order to join or create a new session.
`other`	An NP Library Error Code.

Description

Creates a session room on the NP matching server. The session creation process is kicked off on a different thread.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Non-Blocking (*async* = true) If the process was not able to kick off, the function returns an appropriate error code. On completion of the process the application is notified using a `matchingSessionCreated` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking (*async* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve session information using the `get()` method of the `Future` object.

Notes

If this function is called from the main thread, it should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

See Also

`searchSessions()`, `joinSession()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

inviteToSession

Sends a session invite to a friend of the user.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int inviteToSession(
                        const SessionInformation *currentSession,
                        const InviteMessage *msg
                    );
                }
            }
        }
    }
}
```

Arguments

currentSession	A pointer to the session the user currently is in.
msg	The user specified message.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was kicked off successfully.
SCE_TOOLKIT_NP_MATCHING_SESSION_DOES_NOT_EXIST	The operation failed because the session that an invite was sent for is invalid.

Description

Sends a session invite to a friend of the user.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

joinInvitedSession

Joins the user to a session that they were invited to.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int joinInvitedSession(
                        const MessageAttachment *attachment,
                        sce::Toolkit::NP::Utilities::Future
                        < SessionInformation > *sessionInformation
                    );
                }
            }
        }
    }
}
```

Arguments

attachment	A pointer to the message attachment which was retrieved using `MessagingServiceInterface::retrieveMessageAttachment()`.
sessionInformation	Output. Receives the session information when a session is successfully joined.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was kicked off successfully.
SCE_TOOLKIT_NP_MESSAGE_ATTACHMENT_INVALID	The operation failed because the message attachment type is not supported by the NP Toolkit library.

Description

Joins the user to a session that they were invited to. When the operation has completed an event callback will be generated, and the application can retrieve the result using the `get()` method of the `Future` object.

Notes

This function is asynchronous.

The `Future` object should be valid until the callback of the event is processed.

See Also

`inviteToSession()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

joinSession

Joins a specific session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int joinSession(
                        const JoinSessionRequest *sessionJoinRequest,
                        sce::Toolkit::NP::Utilities::Future
                        < SessionInformation > *sessionInformation,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

sessionJoinRequest	A structure which contains information about session to be joined.
sessionInformation	Output. Receives the session information when a session is successfully joined.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.
SCE_TOOLKIT_NP_MATCHING_SERVICE_BUSY	The operation failed because the matching service is busy processing a previous request.
SCE_TOOLKIT_NP_MATCHING_SESSION_ALREADY_ACTIVE	The operation failed because the user is already in a session. They must leave a session in order to join or create a new session.
other	An NP Library Error Code.

Description

Joins a session on the NP matching server. This function kicks off a process that requests to join a session on a different thread.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On completion of the process the application is notified by a `matchingSessionJoined` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking Process (*`async`* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

See Also

`searchSessions()`, `createSession()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

kickMember

Kicks a room member out of a room.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int kickMember(
                        KickMemberRequest *kickRequest,
                        sce::Toolkit::NP::Utilities::Future< int >
                        *processResult,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

kickRequest	The details about a request to kick a member out of a room.
processResult	Output. Receives the result of this process when the session has been successfully updated.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Kicks a room member out of a room. Only a room owner can kick a member out of a room.

See Also

searchSessions(), createSession(), joinSession()

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

leaveSession

Leaves a currently joined/created session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int leaveSession(
                        const SessionInformation *leavingSession,
                        sce::Toolkit::NP::Utilities::Future
                        < int > *processResult,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

leavingSession	A pointer to the session the user wants to leave. Set to `NULL` if the session is not known.
processResult	Output. Receives the result of this process when the session has been successfully left.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.
`SCE_TOOLKIT_NP_MATCHING_SERVICE_BUSY`	The operation failed because the matching service is busy processing a previous request.
`SCE_TOOLKIT_NP_MATCHING_SESSION_DOES_NOT_EXIST`	The operation failed because the session which the user is trying to leave does not exist.
`other`	An NP Library Error Code.

Description

Leaves a currently joined/created session. This request is processed on a different thread.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On successful completion of the process the application is notified by a `matchingSessionLeft` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking Process (*async* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed. Even when an application receives a `matchingSessionError` `Event`, the session will still be destroyed and therefore no longer be valid.

See Also

`searchSessions()`, `createSession()`, `joinSession()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

modifySession

Modifies a specific session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int modifySession(
                        const ModifySessionRequest *sessionDesc,
                        sce::Toolkit::NP::Utilities::Future
                        < int > *processResult,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

sessionDesc	The session attributes to modify.
processResult	Output. Receives the result of this process when the session has been successfully modified.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Modifies session(room) information on the NP matching server. This function kicks off a process that requests to modify a session on a different thread.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On completion of the process the application is notified by a matchingSessionModified Event. The application can then verify whether there was an error during the process or whether the process was successful by using the hasError() or hasResult() method of the Future object. If an error has occurred, the application can get the error code using the getError() method.

Blocking Process (*async* = false) The function blocks until a result is returned. If the function is successful, it returns SCE_TOOLKIT_NP_SUCCESS; otherwise an appropriate error code is returned.

On successful completion of the operation, the application should update the session information by calling updateSession() and specifying matchingSessionModified for the *userEvent* parameter of the function.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Notes

Only the session owner can modify the session information with the exception of `SCE_TOOLKIT_NP_SESSION_MEMBER_ATTRIBUTE`. If any other session member tries to modify an attribute of a session other than `SCE_TOOLKIT_NP_SESSION_MEMBER_ATTRIBUTE`, the function will immediately return an error code. If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

See Also

`searchSessions()`, `createSession()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

quickSession

Searches for a session and join the first one available.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int quickSession(
                        const SearchSessionsRequest *searchCriteria,
                        sce::Toolkit::NP::Utilities::Future
                        < SessionInformation > *sessionInformation,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

searchCriteria	Specifies the type of a session to look for. Set to `NULL` if not looking for a specific session.
sessionInformation	Output. Receives the session information when a session is successfully joined.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.
`SCE_TOOLKIT_NP_MATCHING_SERVICE_BUSY`	The operation failed because the matching service is busy processing a previous request.
`SCE_TOOLKIT_NP_MATCHING_SESSION_ALREADY_ACTIVE`	The operation failed because the user is already in a session. They must leave a session in order to join or create a new session.
`other`	An NP Library Error Code.

Description

This function searches for a session and joins the first one available on the NP matching server. This function kicks off a process that searches for a session on a NP Toolkit library thread.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On successful completion of the process the application is notified by a `matchingSessionJoined` event. The application can then verify whether there was an error during

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking Process (*async* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

See Also

`searchSessions()`, `createSession()`, `joinSession()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

registerRoomMessageCallback

Registers a callback which will be called for room messages.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         namespace Matching {
            class Interface {
               static int registerRoomMessageCallback(
                  SessionMessageCallback cbfunc
               );
            }
         }
      }
   }
}
```

Arguments

cbfunc	The room message callback function to register.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Registers a callback which will be called for room messages.

See Also

inviteToSession()

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

registerSessionAttributes

Registers attributes used by the session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int registerSessionAttributes(
                        const RegisterSessionAttribute *sessionAttributes,
                        int numParameters
                    );
                }
            }
        }
    }
}
```

Arguments

sessionAttributes	A pointer to the memory address which contains information about the attributes to register.
numParameters	The number of attributes to be registered.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_MATCHING_INVALID_PARAMETERS	The operation failed because information about the attributes was missing.
SCE_TOOLKIT_NP_MATCHING_FAIL_TO_REGISTER_PARAMETERS	The operation failed because the application tried to register more attributes than it can. For more information on this please refer to the *NP Toolkit Library Overview*.

Description

Registers the session attributes which are going to be used by the application during session creation or searching. For correct working of session attributes and filtering, the application needs to register the session attributes when it initializes. The application should not re-register attributes once specified. Once registered, the attributes are maintained throughout the life cycle of the application.

Notes

If a session attribute which the application failed to register is referenced during creation, searching or modifying a session, then `SessionInformation` will not contain any information on the attribute.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

searchSessions

Searches for a session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int searchSessions(
                        const SearchSessionsRequest *searchRequest,
                        sce::Toolkit::NP::Utilities::Future
                        < SessionList > *sessionList,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

searchRequest	Specifies the type of a session to look for.
sessionList	Output. Receives the results of the search.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.
SCE_TOOLKIT_NP_MATCHING_SERVICE_BUSY	The operation failed because the matching service is busy processing a previous request.
other	An NP Library Error Code.

Description

Searches for a session on the NP matching server. On calling, this function kicks off a process that searches for a session on a NP Toolkit library thread.

Non-Blocking Process (*async* = true) If the process was not able to kick off, the function returns an appropriate error code. On completion of the process, the application is notified by a `matchingSessionSearchCompleted` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking Process (*async* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Notes

If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

See Also

`createSession()`, `joinSession()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sendRoomMessage

Sends room/chat messages to the room members.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int sendRoomMessage(
                        SceToolkitNpRoomMessageFlag msgFlag,
                        String msg,
                        SceNpMatching2RoomMemberId *member,
                        SceUInt32 numMembers
                    );
                }
            }
        }
    }
}
```

Arguments

msgFlag	The room message type.
msg	The message data.
member	The list of members to send the message to. To broadcast to all members, specify an empty list.
numMembers	The number of members in the list.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Sends room/chat messages to the room members. When the operation has been completed, the callback registered with `registerRoomMessageCallback()` is called with the appropriate NP matching2 event and room message event (corresponding to `SCE_TOOLKIT_NP_MATCHING_INVALID_ROOM_MESSAGE`).

See Also

`inviteToSession()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

updateSession

Updates the current session information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int updateSession(
                        SessionInformation *currentSession,
                        SceNpMatching2Event *event,
                        Event::UserEvent userEvent
                    );
                }
            }
        }
    }
}
```

Arguments

currentSession	Input/Output. The current session information structure to be updated.
event	Output. Receives the NP Matching2 event that caused the session to be updated.
userEvent	Input. The event for which the session update is called. This could be either a `matchingSessionUpdate` or `matchingSessionModified` Event.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_MATCHING_SESSION_KICKEDOUT	The operation failed because the user has been kicked out of the matching session.
SCE_TOOLKIT_NP_MATCHING_SERVICE_BUSY	The operation failed because the matching service is busy processing a previous request.
SCE_TOOLKIT_NP_MATCHING_SESSION_DOES_NOT_EXIST	The operation failed because the session the user was trying to update does not exist.
SCE_TOOLKIT_NP_MATCHING_SESSION_ROOM_DESTROYED	The operation failed because the session the user was in has been destroyed.

Description

Updates the current session information. This function should be called when an application receives a `matchingSessionUpdate` or `matchingSessionModified` Event.

Notes

On receiving an error, the application should clear the current session.

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

This operation should be called from the same thread on which the callback was received (the NP Toolkit thread).

See Also

searchSessions(), createSession(), joinSession()

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

updateSessionAttribute

Updates the session attributes of the current session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Matching {
                class Interface {
                    static int updateSessionAttribute(
                        UpdateAttributeRequest *attributeRequest,
                        sce::Toolkit::NP::Utilities::Future
                        < int > *processResult,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

attributeRequest	Input. The session attribute to be updated.
processResult	Output. Receives the result of this process when the session has been successfully updated.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Updates the session attributes of the current session.

See Also

searchSessions(), createSession(), joinSession()

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::MessageAttachment

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::MessageAttachment

Contains the components of a message.

Definition

```
#include <np_toolkit.h>
class MessageAttachment {};
```

Description

Contains the components of a message. These can be left empty, which will cause the user to be prompted to manually fill in the data using the System Software.

Methods Summary

Methods	Description
getAttachmentData	Gets the attachment data contained within the MessageAttachment object.
getAttachmentSize	Gets the size of the attachment data.
getCommId	Gets the NP Communication ID associated with the attachment.
MessageAttachment	The default constructor.
~MessageAttachment	The destructor for the MessageAttachment class.
setAttachmentData	Sets the attachment data for the MessageAttachment object.
setCommunicationID	Sets the NP Communication ID for the MessageAttachment object.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

MessageAttachment

The default constructor.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class MessageAttachment {
                MessageAttachment();
            }
        }
    }
}
```

Arguments

None

Return Values

None

Description

The default constructor. This does not allocate any memory on the heap.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~MessageAttachment

The destructor for the MessageAttachment class.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class MessageAttachment {
                ~MessageAttachment();
            }
        }
    }
}
```

Arguments

None

Return Values

None

Description

The destructor for the MessageAttachment class. This clear the internal buffer and de-allocates any memory that was allocated to the heap.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

getAttachmentData

Gets the attachment data contained within the MessageAttachment object.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         class MessageAttachment {
            inline SceChar8 *getAttachmentData() const;
         }
      }
   }
}
```

Arguments

None

Return Values

A pointer to the attachment data buffer.

Description

Gets the attachment data contained within the MessageAttachment object.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getAttachmentSize

Gets the size of the attachment data.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class MessageAttachment {
                inline size_t getAttachmentSize() const;
            }
        }
    }
}
```

Arguments

None

Return Values

The size of the allocated attachment data buffer.

Description

Gets the size of the attachment data.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getCommId

Gets the NP Communication ID associated with the attachment.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class MessageAttachment {
                inline SceNpCommunicationId &getCommId();
            }
        }
    }
}
```

Arguments

None

Return Values

The NP Communication ID.

Description

Gets the NP Communication ID associated with the attachment.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setAttachmentData

Sets the attachment data for the `MessageAttachment` object.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class MessageAttachment {
                int setAttachmentData(
                    SceChar8 *data,
                    size_t dataSize
                );
            }
        }
    }
}
```

Arguments

data	The data to be copied into the buffer.
dataSize	The size of the data array to be copied in bytes.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_ARGUMENT`	The operation failed because *dataSize* was not a valid integer between 1-`SCE_NP_BASIC_MAX_MESSAGE_ATTACHMENT_SIZE`.
`SCE_TOOLKIT_NP_FAILED_ALLOCATE`	The operation failed because the required memory could not be allocated on the heap.

Description

Sets the attachment data for the `MessageAttachment` object. Use this method for setting the buffer inside the object. The data given is copied into an internal buffer which is kept on the heap.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setCommunicationID

Sets the NP Communication ID for the MessageAttachment object.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class MessageAttachment {
                void setCommunicationID(
                    SceNpCommunicationId &commsID
                );
            }
        }
    }
}
```

Arguments

commsID	The NP Communication ID.

Return Values

None

Description

Sets the NP Communication ID for the MessageAttachment object.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::MessageData

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::MessageData

Contain the components of a message.

Definition

```
#include <np_toolkit.h>
struct MessageData {};
```

Description

Contains the components of a message.

Fields

Public Instance Fields

`SceChar8 *attachment`	The buffer containing the actual data. This should be set using `MessageAttachment::setData()`.
`size_t attachmentSize`	The number of bytes within the buffer that are being used.
`AttachmentURL attachmentURL`	The details about the attachment URL, which will only be set when sending a custom data message.
`SceToolkitNpAvailablePlatform availablePlatforms`	The platforms that this message will be available on.
`String body`	The body text of the message.
`String dataDescription`	The data description.
`String dataName`	The data name.
`SceToolkitNpDialogType dialogFlag`	A flag that indicates whether the recipient list is editable by the user.
`SceUInt16 expireMinutes`	The amount of time until the expiration of the message in minutes from now. This parameter is optional for the application data attached message. A value of 0 means that the message does not expire.
`String iconPath`	The supported file types are PNG or JPEG, and the maximum file size of the icon is defined by `SCE_NP_MESSAGE_DIALOG_MAX_INDEX_ICON_SIZE`. The path to an icon, which needs to be displayed in the message.
`SceNpOnlineId *npIds`	A list of NP IDs.
`size_t npIdsCount`	The number of IDs in the *npIds* list. If *npIds* is `NULL`, specify the maximum number of NP IDs which can be added by the user.
`SceNpSessionId npSessionId`	The session ID related to the session server if the message is an invite message.

Methods Summary

Methods	Description
MessageData	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

MessageData

The default constructor.

Definition

```
#include <np_toolkit.h>
inline MessageData();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Messaging

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Messaging

The namespace containing messaging functionality.

Definition

```
namespace Messaging {}
```

Description

The namespace containing messaging functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Messaging::Interface	The static interface for sending messages.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Messaging::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Messaging::Interface

The static interface for sending messages.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

The static interface for sending messages, which forwards commands on to the NP Toolkit thread. The NP Toolkit then attempts to fulfill the requests, and success or failure is notified by an event callback.

These functions have a corresponding set of messages that are passed to event callbacks to indicate asynchronous errors or events.

Events:

`MESSAGE_ATTACHMENT_RECEIVED`	A message with a data attachment was received.
`MESSAGE_INVITE_RECEIVED`	A message with an invite was received.
`MESSAGE_XMB_INVITE_ACCEPTED`	An invite was accepted via the system menu.

Results:

`MESSAGE_ATTACHMENT_OPENED`	A message with a data attachment was opened, and the sysutil GUI is now closed.
`MESSAGE_INVITE_ACCEPTED`	An invite was accepted via the NP Toolkit, and the sysutil GUI is now closed.
`MESSAGE_SENT`	A message was sent successfully.

These `Event` messages can contain error codes.

Methods Summary

Methods	Description
`displayReceivedMessages`	Used to display receive message data using the System Software GUI.
`retrieveInGameMessage`	Retrieves an in-game data message.
`retrieveMessageAttachment`	Retrieves the attached data of an `SceAppUtilAppEventParam` message.
`retrieveMessageAttachmentFromId`	Retrieves the attached data of a message using a message ID.
`sendInGameMessage`	Sends an in-game data message to other PlayStation™Network users.
`sendMessage`	Sends a message to a remote user.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

displayReceivedMessages

Used to display receive message data using the System Software GUI.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Messaging {
                class Interface {
                    static int32_t displayReceivedMessages(
                        SceToolkitNpMessageType messageType
                    );
                }
            }
        }
    }
}
```

Arguments

messageType	Specifies whether show invites or data messages to the user.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.

Description

Upon calling this function the System GUI is launched allowing the user to select from their inbox all messages they have received that relate to this NP Communication ID. The messages are filtered by whether they are invite or data messages.

Notes

This method is asynchronous.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

retrieveInGameMessage

Retrieves an in-game data message.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Messaging {
                class Interface {
                    static int32_t retrieveInGameMessage(
                        unsigned int messageId,
                        sce::Toolkit::NP::Utilities::Future
                        < ReceivedInGameDataMessage > *inGameDataMessage
                    );
                }
            }
        }
    }
}
```

Arguments

messageId	The ID of the message.
inGameDataMessage	A `Future` object, which receives the in-game data message.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_MESSAGE_INVALID_ID`	The operation failed because the message ID was not valid.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *inGameDataMessage* parameter.
`SCE_TOOLKIT_NP_FAILED_ALLOCATE`	The operation failed because there is not enough memory to process the request.
`messageInGameDataRetrievalDone`	This event is passed to an event callback when this operation is processed. Please check the *returnCode* member of the `Event` object to see if the operation succeeded or failed.

Description

Retrieves an in-game data message. The NP `Toolkit` library receives in-game data messages and caches them internally. The cache can hold up to 5 messages. When an in-game data message is received, the NP `Toolkit` library will pass a `messageInGameDataReceived` event back to the application. The *returnCode* member of the `Event` object contains the message ID, and this should be used when calling this function to retrieve the message. Due to the relatively small size of the cache, it is recommended that this function is called as soon as the `messageInGameDataReceived` event is received.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Notes

This method is asynchronous.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

retrieveMessageAttachment

Retrieves the attached data of an SceAppUtilAppEventParam message.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Messaging {
                class Interface {
                    static int32_t retrieveMessageAttachment(
                        const SceAppUtilAppEventParam *eventParam,
                        sce::Toolkit::NP::Utilities::Future
                        < MessageAttachment > *attch
                    );
                }
            }
        }
    }
}
```

Arguments

eventParam	An event parameter.
attch	A `Future` object, which receives the attached data.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *attch*.
`messageRetrieved`	This event is passed to an event callback to signify the operation has completed successfully.
`messageError`	This event is passed to an event callback to signify that an error occurred during the operation.

Description

Retrieves the attached data of an `SceAppUtilAppEventParam` message. This function should be called when an `SceAppUtilAppEventParam` message is triggered. When the operation has completed, an event callback will be generated, and the `Future` object passed to *attch* needs to be polled to obtain the attached data.

Notes

This method is asynchronous.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

retrieveMessageAttachmentFromId

Retrieves the attached data of a message using a message ID.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Messaging {
                class Interface {
                    static int32_t retrieveMessageAttachmentFromId(
                        const SceNpMessageId *msgId,
                        sce::Toolkit::NP::Utilities::Future
                        < MessageAttachment > *attch,
                        SceToolkitNpMessageType msgType
                    );
                }
            }
        }
    }
}
```

Arguments

msgId	The ID of the message.
attch	A `Future` object, which receives the attached data.
msgType	Specifies whether the message contains invite data or custom data.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *attch*.
`messageRetrieved`	This event is passed to an event callback to signify the operation has completed successfully.
`messageError`	This event is passed to an event callback to signify that an error occurred during the operation.

Description

Retrieves the attached data of a message using a message ID. When the operation has completed an event callback will be generated, and the `Future` object passed to *attch* needs to be polled to obtain the attached data.

Notes

This method is asynchronous.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sendInGameMessage

Sends an in-game data message to other PlayStation™Network users.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Messaging {
                class Interface {
                    static int32_t sendInGameMessage(
                        const InGameDataMessage *inGameMessage
                    );
                }
            }
        }
    }
}
```

Arguments

inGameMessage	The in-game data message containing the NP ID of the recipient as well as the message.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was kicked off successfully.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because an invalid pointer was passed to the *inGameMessage* parameter.
SCE_TOOLKIT_NP_FAILED_ALLOCATE	The operation failed because there is not enough memory to process the request.
messageSent	This event is passed to an event callback to signify the operation has completed successfully.
messageError	This event is passed to an event callback to signify that an error occurred during the operation.

Description

Sends an in-game data message to other PlayStation™Network users. It is possible to specify which platform and user the message will be sent to. Unlike other methods of sending a message, this method does not utilize the system's GUI for the user to input their message in; instead it has no GUI. Messages are only sent to other users who are both online and in the same game context at the moment of sending. If this is not the case, the message will not be received.

Notes

This method is asynchronous.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sendMessage

Sends a message to a remote user.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         namespace Messaging {
            class Interface {
               static int32_t sendMessage(
                  const MessageData *msg,
                  SceToolkitNpMessageType messageType,
                  bool gui = true
               );
            }
         }
      }
   }
}
```

Arguments

msg	The message details including recipients, subject and body text.
messageType	Specifies whether this is an invite or data message.
gui	A flag that specifies whether to use the System GUI or not. Defaults to true.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation completed successfully.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because no message data was sent.
`SCE_TOOLKIT_NP_INVALID_ARGUMENT`	The operation failed because the message body was too long.
`SCE_TOOLKIT_NP_MESSAGE_ATTACHMENT_INVALID`	The operation failed because the attachments where either `NULL` or incorrect. For example, a string could have been too long.
`SCE_TOOLKIT_NP_FAILED_ALLOCATE`	The operation failed because memory could not be allocated on the heap.
`other`	error code A NP Library/NP LIbrary `Toolkit` Error Code

Description

This function will cause the NP Toolkit thread to send a message to a remote user on the PlayStation™Network. The function supports both custom data attachment messages and cross-title invites which are the primary uses of the messaging system. Using the GUI is optional so long as all the arguments for the message are present. If anything is missing in the parameters and the GUI is enabled, the user will be prompted via System Software to enter the data manually.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Notes

This method is asynchronous. When the system GUI is used on successful termination of dialog box, a `messageSent` event is returned through the registered callback, and the *`returnCode`* member of the `Event` object is set to `SceNpMessageDialogResultUserAction`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
ModifySessionAttributes

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ModifySessionAttributes

A structure used to specify attributes which needs to be modified.

Definition

```
#include <np_toolkit.h>
struct ModifySessionAttributes : public sce::Toolkit::NP::Request {};
```

Description

A structure used to specify attributes which needs to be modified.

Fields

Public Instance Fields

`char` *`attribute [SCE_TOOLKIT_NP_MAX_ATTRIBUTE_LENGTH]`*	The name of the session attribute.
`SessionAttributeValue` *`attributeValue`*	The value of the session attribute.

Methods Summary

Methods	Description
`ModifySessionAttributes`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ModifySessionAttributes

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ModifySessionAttributes();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
ModifySessionRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ModifySessionRequest

A request structure used to specify how session information should be modified.

Definition

```
#include <np_toolkit.h>
struct ModifySessionRequest : public sce::Toolkit::NP::Request {};
```

Description

A request structure used to specify how session information should be modified.

Fields

Public Instance Fields

`SceToolkitNpSessionAttributeType` *`attributeType`*	The type of attribute.
`SceNpMatching2FlagAttr` *`flagAttr`*	Specifies the room flag attribute which needs to be set.
`SceNpMatching2FlagAttr` *`flagFilter`*	Specifies the room flag filter which needs to be set.
`int32_t` *`numAttributes`*	The number of attributes to be modified.
`SceUInt8` *`padding[3]`*	Padding.
`ModifySessionAttributes` *`*sessionAttributes`*	The attributes to be modified. Only one type of attribute (`SCE_TOOLKIT_NP_SESSION_*_ATTRIBUTE`) can be specified.

Methods Summary

Methods	Description
`ModifySessionRequest`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ModifySessionRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ModifySessionRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Near

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Near

The namespace containing the PlayStation™Network near service.

Definition

```
namespace Near {}
```

Description

The namespace containing the PlayStation™Network near service.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Near::Interface	The near interface contains a set of static methods for managing "near" actions.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Near::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Near::Interface

The near interface contains a set of static methods for managing “near” actions.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

The near interface contains a set of static methods for managing “near” actions. This includes creation and retrieval of “near” gifts, retrieving the user’s “near” status, retrieving the user’s nearby users and launching the “near” application.

Methods Summary

Methods	Description
`compareGiftId`	Compares a specified gift ID against the gift ID of a gift referenced by a `SceNearGiftDiscoveringId`.
`createGift`	Creates and registers a gift to be distributed to other nearby users.
`getGiftData`	Retrieves gift’s data body.
`getGiftDetails`	Retrieves information about a gift such as the status of the gift, who sent the gift, the gift’s name and its description.
`getGiftImage`	Retrieves a gift’s image.
`getMyStatus`	Retrieves current user’s “near” status.
`getNeighbor`	Retrieves the current user’s nearby users.
`initNear`	Initializes the “near” service.
`launchNearApp`	Launches the “near” application.
`relayGift`	Relays a gift ready for redistribution.
`termNear`	Terminates the “near” service.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

compareGiftId

Compares a specified gift ID against the gift ID of a gift referenced by a `SceNearGiftDiscoveringId`.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         namespace Near {
            class Interface {
               static int compareGiftId(
                  const SceUInt32 giftId,
                  const SceNearGiftDiscoveringId discoveringGiftId
               );
            }
         }
      }
   }
}
```

Arguments

giftId	The gift ID to check against a gift ID held internally in a header.
discoveringGiftId	The `SceNearGiftDiscoveringId` of the gift whose internal gift ID is to be checked against the *giftId* parameter.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_GIFT_NOT_MATCH`	The operation failed because the Ids did not match.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because NP Toolkit could not allocate the required memory.
`SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED`	The operation failed because the “near” service has not been initialized yet.

Description

Compares a specified gift ID against the gift ID of a gift referenced by a `SceNearGiftDiscoveringId`. The header of the gift referenced by the `SceNearGiftDiscoveringId` is checked to see if the gift ID within it matches *giftId*.

Notes

This function is not thread safe.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

For information on additional error messages, please see documentation on the following functions: `sceNearOpenReceivedGiftData()`,`sceNearReadReceivedGiftData()` and `sceNearCloseReceivedGiftData()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

createGift

Creates and registers a gift to be distributed to other nearby users.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Near {
                class Interface {
                    static int createGift(
                        const NearGiftInputParam *input,
                        const bool isAsync = true
                    );
                }
            }
        }
    }
}
```

Arguments

input	The necessary information for creating and registering the gift.
isAsync	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because NP Toolkit could not allocate the required memory.
`SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED`	The operation failed because the “near” service has not been initialized yet.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer (a `NULL` pointer for example) was passed as an argument.
`SCE_NP_MANAGER_ERROR_NOT_SIGNIN`	The operation failed because the user is not online. Note that other errors from `sceNpManagerGetNpId()` may also be returned.
`Event::nearCreateRegisterGiftSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the gift creation and registration succeeded.
`Event::nearCreateRegisterGiftFailed`	In the case of an asynchronous operation, this event will be passed to an event callback when the gift creation and registration failed.

Description

Creates and registers a gift to be distributed to other nearby users. When a gift is created using NP Toolkit, it uses the first 256 bytes of the gift’s body as a header. It also inserts into the header a gift ID

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

and the online name of up to 10 users who have received and passed the gift on using `relayGift()`
Please see `sce::Toolkit::NP::NearGiftDataHeader` for more information.

Because NP Toolkit gifts use a header, gifts created outside of NP Toolkit will not be compatible with the NP Toolkit “near” service interface; internally NP Toolkit processes the gift’s header. For example `relayGift()` and `compareGiftId()` will not work on gifts created outside of NP Toolkit.

Notes

For information on additional error messages, please see documentation on the following functions: `sceIoOpen()`, `sceIoRead()`, and `sceNearSetGift()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getGiftData

Retrieves gift's data body.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Near {
                class Interface {
                    static int getGiftData(
                        sce::Toolkit::NP::Utilities::Future
                        < NearDiscoveredGiftData > *giftData,
                        const SceNearGiftDiscoveringId discoveringGiftId,
                        const bool isAsync = true
                    );
                }
            }
        }
    }
}
```

Arguments

giftData	A `Future` object with a `sce::Toolkit::NP::NearDiscoveredGiftData` data type, which receives the specified gift's data body.
discoveringGiftId	The discovering ID of the gift whose data that needs to be returned.
isAsync	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because NP Toolkit could not allocate the required memory.
`SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED`	The operation failed because the "near" service has not been initialized yet.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer (a `NULL` pointer for example) was passed as an argument.
`Event::nearGetGiftDataSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the gift data retrieval has succeeded.
`Event::nearGetGiftDataFailed`	In the case of an asynchronous operation, this event will be passed to an event callback when the gift data retrieval has failed.

Description

Retrieves gift's data body. This includes the data body size and the address to the data buffer.

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

NP Toolkit uses a data buffer memory area internally to store the data buffer retrieved from the "near" utility. The data buffer content is not copied over to the application side. Instead, the memory address of the data memory area is given to the application via the *`pGiftData`* member of the `NearDiscoveredGiftData` struct. It is important not to modify or delete the memory area referred to by *`pGiftData`*. If a local copy is required, the memory should be copied to the application's local memory area after this function has returned.

Notes

This function is not thread safe.

For information on additional error messages, please see documentation on the following functions: `sceNearOpenReceivedGiftData()`, `sceNearReadReceivedGiftData()` and `sceNearCloseReceivedGiftData()`.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getGiftDetails

Retrieves information about a gift such as the status of the gift, who sent the gift, the gift's name and its description.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Near {
                class Interface {
                    static int getGiftDetails(
                        sce::Toolkit::NP::Utilities::Future
                        < NearDiscoveredGiftDetails > *giftDetails,
                        const SceNearGiftDiscoveringId giftDiscoveringId,
                        const bool isAsync = true
                    );
                }
            }
        }
    }
}
```

Arguments

giftDetails	A `Future` object with a `sce::Toolkit::NP::NearDiscoveredGiftDetails` data type, which receives the information on the specified gift.
giftDiscoveringId	The discovering ID of the gift whose data that needs to be returned.
isAsync	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because NP Toolkit could not allocate the required memory.
`SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED`	The operation failed because the "near" service has not been initialized yet.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer (a `NULL` pointer for example) was passed as an argument.
`Event::nearGetGiftInfoSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the gift status retrieval succeeded.
`Event::nearGetGiftImageFailed`	In the case of an asynchronous operation, this event will be passed to an event callback when the gift status retrieval failed.

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Description

Retrieves information about a gift such as the status of the gift, who sent the gift, the gift's name and its description.

Notes

For information on additional error messages, please see documentation on the following functions: `sceNearGetDiscoveredGiftSender()`, `sceNearGetDiscoveredGiftInfo()` and `sceNearGetDiscoveredGiftStatus()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getGiftImage

Retrieves a gift's image.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Near {
                class Interface {
                    static int getGiftImage(
                        sce::Toolkit::NP::Utilities::Future
                        < NearDiscoveredGiftImage > *giftImage,
                        const SceNearGiftDiscoveringId discoveringGiftId,
                        const bool isAsync = true
                    );
                }
            }
        }
    }
}
```

Arguments

giftImage	A `Future` object with a `sce::Toolkit::NP::NearDiscoveredGiftImage` data type, which receives the specified gift's image.
discoveringGiftId	The discovering ID of the gift whose data that needs to be returned.
isAsync	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because NP Toolkit could not allocate the required memory.
`SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED`	The operation failed because the "near" service has not been initialized yet.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer (a `NULL` pointer for example) was passed as an argument.
`Event::nearGetGiftImageSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the gift image retrieval has succeeded.
`Event::nearGetGiftImageFailed`	In the case of an asynchronous operation, this event will be passed to an event callback when the gift image retrieval has failed.

Description

Retrieves a gift's image. This includes the image size and address to the image buffer.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

NP Toolkit uses an image buffer memory area internally to store the data buffer retrieved from the "near" utility. The image buffer content is not copied over to the application side. Instead, the memory address of the image memory area is given to the application via the *`pImageBuffer`* member of the `NearDiscoveredGiftImage` struct. It is important not to modify or delete the memory area referred to by *`pImageBuffer`*. If a local copy is required, the memory should be copied to the application's local memory area after this function has returned.

Notes

This function is not thread safe.

For information on additional error messages, please see documentation on the following functions: `sceNearOpenDiscoveredGiftImage()`, `sceNearReadDiscoveredGiftImage()`, and `sceNearCloseDiscoveredGiftImage()`.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getMyStatus

Retrieves current user's "near" status.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Near {
                class Interface {
                    static int getMyStatus(
                        sce::Toolkit::NP::Utilities::Future
                        < SceNearMyStatus > *myStatus,
                        const bool isAsync = true
                    );
                }
            }
        }
    }
}
```

Arguments

myStatus	A Future object with a sce::Toolkit::NP::SceNearMyStatus data type, which receives the current user's "near" status.
isAsync	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_OUT_OF_MEMORY	The operation failed because NP Toolkit could not allocate the required memory.
SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED	The operation failed because the "near" service has not been initialized yet.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because an invalid pointer (a NULL pointer for example) was passed as an argument.
Event::nearGetMyStatusSuccess	In the case of an asynchronous operation, this event will be passed to an event callback when status retrieval succeeded.
Event::nearGetMyStatusFailed	In the case of an asynchronous operation, this event will be passed to an event callback when status retrieval failed.

Description

Retrieves current user's "near" status. This includes the distance covered, the number of gifts discovered, the number of titles discovered and the number of users discovered.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Notes

For information on additional error messages, please see documentation on the following functions: `sceNearGetNeighbors()`, `sceNearGetLastNeighborFoundDateTime()`, `sceNearGetRecentNeighbors()`, and `sceNearGetNewNeighbors()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getNeighbor

Retrieves the current user's nearby users.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Near {
                class Interface {
                    static int getNeighbor(
                        sce::Toolkit::NP::Utilities::Future
                        < NearNeighbors > *neighbors,
                        const NeighborType type,
                        const bool isAsync = true
                    );
                }
            }
        }
    }
}
```

Arguments

neighbors	A `Future` object with a `sce::Toolkit::NP::NearNeighbors` data type, which receives a list of up to 100 nearby users' `SceNpIds`.
type	The type of nearby users to be retrieved.
isAsync	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because NP Toolkit could not allocate the required memory.
`SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED`	The operation failed because the "near" service has not been initialized yet.
`Event::nearGetNeighborFailed`	In the case of an asynchronous operation, this event will be passed to an event callback when nearby user retrieval failed.
`Event::nearGetNeighborSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when nearby user retrieval succeeded.

Description

Retrieves the current user's nearby users. There are three types of nearby user what can be retrieved. The default type includes all the nearby users that the current user has discovered. The second type, which is `sceNpToolkitNPNeighborRecent`, only includes nearby users discovered since the last update. The third type, which is `sceNpToolkitNPNeighborRecent`, only includes newly discovered users from the last update. Up to 100 users can be retrieved.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Notes

For additional error messages returned, see the references to the following functions: `sceNearGetNeighbors()`,`sceNearGetLastNeighborFoundDateTime()`, `sceNearGetRecentNeighbors()` and `sceNearGetNewNeighbors()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

initNear

Initializes the “near” service.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Near {
                class Interface {
                    static int initNear(
                        const unsigned int memorySize,
                        const unsigned int version,
                        const bool isAsync = true
                    );
                }
            }
        }
    }
}
```

Arguments

memorySize	The size of the “near” utility work memory area. This argument should be a value of `SCE_NEAR_UTIL_DEFAULT_WORKMEMORY_SIZE` or larger, and it will default to `SCE_NEAR_UTIL_DEFAULT_WORKMEMORY_SIZE` if a value less than `SCE_NEAR_UTIL_DEFAULT_WORKMEMORY_SIZE` is supplied.
version	The version of the data exchanged via `SceNpCommunicationId`.
isAsync	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_NEAR_ALREADY_INITED`	The operation failed because the “near” service has already been initialized.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because NP Toolkit could not allocate the required memory.
`Event::nearInitFailed`	In async mode this will be generated when initialization fails
`Event::nearInitSuccess`	In async mode this will be generated when initialization succeeds

Description

Initializes the “near” service. Unlike other NP Toolkit services, this function needs to be called before using the “near” service. This is because an application is required to specify the amount of the memory that the “near” service will allocate internally for gift creation and retrieval before using the service.

Internally, this function calls `sceNearInitialize()`, so “work” memory will be allocated as documented for `sceNearInitialize()`. In addition to “work” memory allocated for the “near” utility, a memory area of `SCE_NEAR_GIFT_IMAGE_MAX_SIZE` will be allocated for images, and a

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

memory area of SCE_NEAR_GIFT_DATA_MAX_SIZE will be allocated for gift data. Image and gift data processing takes place in these memory areas.

Notes

For information on additional error messages, please see documentation on sceNearInitialize().

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

launchNearApp

Launches the “near” application.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Near {
                class Interface {
                    static int launchNearApp(
                        SceNearAppAction updateType,
                        SceNearGiftDiscoveringId giftId
                    );
                }
            }
        }
    }
}
```

Arguments

updateType	The type of launch. Please refer to documentation on `SceNearAppAction` for the possible options.
giftId	If the *updateType* is `SCE_NEAR_APP_ACTION_TAKE_GIFT`, the `SceNearGiftDiscoveringId` of the gift the user can download should be specified. For any other type of update, 0 should be passed in.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED`	The operation failed because the “near” service has not been initialized yet.

Description

Launches the “near” application. Calling this function will suspend the game application. Depending on the type of launch, the user will arrive in a specific area of the “near” application. For example, the “Update” area where user can sync with the server or to download a discovered gift.

Notes

For information on additional error messages, please see documentation on the following functions: `sceNearLaunchNearAppForUpdate()` and `sceNearLaunchNearAppForDownload()`.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

relayGift

Relays a gift ready for redistribution.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Near {
                class Interface {
                    static int relayGift(
                        const NearRelayGiftParam *giftData,
                        const bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

giftData	A NearRelayGiftParam object containing the gift condition, unit and `SceNearGiftDiscoveringId` that needs to be relayed.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_GIFT_NOT_MATCH`	The operation failed because the gift ID does not match.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because NP Toolkit could not allocate the required memory.
`SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED`	The operation failed because the "near" service has not been initialized yet.
`SCE_NP_MANAGER_ERROR_NOT_SIGNIN`	The operation failed because the user is not online. Note that other errors from `sceNpManagerGetNpId()` may also be returned.
`Event::nearRelayGiftSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the gift relay has succeeded.
`Event::nearRelayGiftFailed`	In the case of an asynchronous operation, this event will be passed to an event callback when the gift relay has failed.

Description

Relays a gift ready for redistribution. This function repackages "received" gifts and registers them for distribution again. Before repackaging, the "received" gift's header is modified by inserting the current user's online name into it.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Notes

This function is not thread safe.

For information on additional error messages, please see documentation on the following functions: `sceNearOpenReceivedGiftData()`, `sceNearReadReceivedGiftData()`, `sceNearCloseReceivedGiftData()` and `sceNearSetGift()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

termNear

Terminates the “near” service.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Near {
                class Interface {
                    static int termNear();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED	The “near” service has already been terminated.

Description

Terminates the “near” service. Work memory allocated during `initNear()` will be freed.

Notes

For information on additional error messages, please see documentation on the `sceNearInitialize()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::NetInfo

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::NetInfo

The namespace containing network status functionality.

Definition

```
namespace NetInfo {}
```

Description

The namespace containing network status functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::NetInfo::Interface	The interface for the network status.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::NetInfo::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::NetInfo::Interface

The interface for the network status.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

This interface allows users to query the network interface, status, collected bandwidth information, and launch the login dialog for the PlayStation™Network.

Methods Summary

Methods	Description
getBandwidthInfo	Gets bandwidth information (asynchronous only).
getNetInfo	Gets basic network information.
getNetInfoDetailed	Gets detailed network information.
psnLoginDialogStart	Starts the network login dialog.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

getBandwidthInfo

Gets bandwidth information (asynchronous only).

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace NetInfo {
                class Interface {
                    static int getBandwidthInfo(
                        sce::Toolkit::NP::Utilities::Future
                        < SceNpBandwidthTestResult > *bandwidthInfo
                    );
                }
            }
        }
    }
}
```

Arguments

bandwidthInfo	Output. Receives the upstream and downstream results.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because the pointer passed to *bandwidthInfo* was invalid.
SCE_TOOLKIT_NP_OPERATION_IN_PROGRESS	The operation failed because the previous operation is in progress.

Description

Measures the network bandwidth between the client and the NP server. Both the upstream and downstream bandwidths are obtained. Only conduct bandwidth tests when necessary as they increase the load on the NP server.

Notes

Bandwidth cannot be measured on a network of NAT type 3.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getNetInfo

Gets basic network information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace NetInfo {
                class Interface {
                    static int getNetInfo(
                        sce::Toolkit::NP::Utilities::Future
                        < NetStateBasic > *info
                    );
                }
            }
        }
    }
}
```

Arguments

info	Output. Receives the basic network information

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because the pointer passed to *info* was invalid.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the user is not connected to the network.

Description

Gets useful network information such as connection status, IP address and NAT information.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getNetInfoDetailed

Gets detailed network information.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         namespace NetInfo {
            class Interface {
               static int getNetInfoDetailed(
                  sce::Toolkit::NP::Utilities::Future
                  < NetStateDetailed > *detailedInfo
               );
            }
         }
      }
   }
}
```

Arguments

detailedInfo	Output. Receives detailed network information.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because the pointer passed to *detailedInfo* was invalid.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the user is not connected to the network.

Description

Gets verbose network information. This information can only be used for debugging purposes.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

psnLoginDialogStart

Starts the network login dialog.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace NetInfo {
                class Interface {
                    static int psnLoginDialogStart();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Starts the network start dialog and prompts the user to sign into the PlayStation™Network.

Notes

`sceCommonDialogUpdate()` must be called to ensure the login dialog is displayed.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: NpSessionDetailedInformation

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::NpSessionDetailedInformation

Represents detailed information about an NP Session.

Definition

```
#include <np_toolkit.h>
struct NpSessionDetailedInformation {};
```

Description

Represents detailed information about an NP Session.

Fields

Public Instance Fields

`SceToolkitNpAvailablePlatform` *`availablePlatforms`*	The platforms the session is available on.
`uint64_t` *`creationTime`*	The time the session was created.
`bool` *`joinable`*	A flag that specifies whether the session is joinable.
`SceToolkitNpSessionLockFlag` *`locked`*	A flag that specifies whether the session is joinable.
`char` *`m_reserved[3]`*	Reserved.
`int32_t` *`maxSlots`*	The maximum number of slots in a session.
`SceNpSessionId` *`npSessionId`*	The session ID related to the Session server.
`NpSessionMember` *`sessionCreator`*	The creator of the session.
`NpSessionMemberList` *`sessionMembers`*	The list of session members.
`char` *`sessionName[SCE_TOOLKIT_NP_SESSION_NAME_MAX_SIZE]`*	The session name.
`char` *`sessionPrivacy[SCE_TOOLKIT_NP_SESSION_NAME_MAX_PRIVACY_DESC]`*	A flag that specifies whether the session is private or public.
`char` *`sessionStatus[SCE_TOOLKIT_NP_SESSION_STATUS_MAX_SIZE]`*	The status string, which will be register with Session server on registration.
`char` *`sessionType[SCE_TOOLKIT_NP_SESSION_NAME_SESSION_TYPE_DESC]`*	A flag that specifies whether the session is "owner-bind" or "owner-migration". By default if no flag is set "owner-bind". Specify `SCE_TOOLKIT_NP_CREATE_HOST_MIGRATION_SESSION` for "owner-migration".

Methods Summary

Methods	Description
`NpSessionDetailedInformation`	The default constructor.
`~NpSessionDetailedInformation`	The destructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

NpSessionDetailedInformation

The default constructor.

Definition

```
#include <np_toolkit.h>
inline NpSessionDetailedInformation();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~NpSessionDetailedInformation

The destructor.

Definition

```
#include <np_toolkit.h>
inline ~NpSessionDetailedInformation();
```

Arguments

None

Return Values

None

Description

The destructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::NpSessionRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::NpSessionRequest

Represents a request to get Session server information.

Definition

```
#include <np_toolkit.h>
struct NpSessionRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to get Session server information.

Fields

Public Instance Fields

`SceNpSessionId` *`npSessionId`*	The session ID related to the Session server.

Methods Summary

Methods	Description
`NpSessionRequest`	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

NpSessionRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline NpSessionRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Parameters

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Parameters

Contains the parameters for initializing the NP Toolkit library.

Definition

```
#include <np_toolkit.h>
class Parameters {};
```

Description

Contains the parameters for initializing the NP Toolkit library. These are all initialized to a default value when the constructor is called. The class members are all public so these starting values can easily be updated before the call to `Interface::init()`.

Fields

Public Instance Fields

`unsigned short` *`m_ageRating`*	The age rating of the title in years old.
`void` *`*m_appData`*	A pointer to the application data which is returned when *`m_callbackFunc2`* is called.
`NpToolkitCallback` *`m_callbackFunc`*	A pointer to a callback used for returning events to the application.
`NpToolkitCallback2` *`m_callbackFunc2`*	A pointer to a callback used for returning events to the application. This callback allows for application data to be returned via the callback.
`OStream` *`&m_err`*	An output stream for error information.
`CommunicationId` *`m_id`*	The NP Communication ID of this application.
`OStream` *`&m_out`*	An output stream for debug information.
`char` *`m_padding[1]`*	Padding.
`SceToolkitNpPushNotificationFlag` *`m_pushNotificationFlag`*	A flag that specifies whether to enable Push Notification. By default, `Presence`, Session and Friend Push Notification is enabled.
`ServiceId` *`m_title`*	The NP Service ID of this application.
`bool` *`m_trial`*	A flag that specifies whether the application is in trial mode.

Methods Summary

Methods	Description
`Parameters`	A constructor for the `Parameters` class.
`Parameters`	A constructor for the `Parameters` class.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

Parameters

A constructor for the Parameters class.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Parameters {
                inline Parameters(
                    NpToolkitCallback mCallback,
                    CommunicationId &id
                );
            }
        }
    }
}
```

Arguments

mCallback	The `NpToolkitCallback` to be registered for events.
id	The main NP Communication ID assumed to be used for all services.

Return Values

None

Description

A constructor for the `Parameters` class.

Default values:

`m_out` = `Cout`

`m_err` = `Cerr`

`m_ageRating` = 0 - indicates a universal game

`m_trial` = false - not a trial mode game

`m_id` = An empty `CommunicationId`. This will not work for games needing certain services.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Parameters

A constructor for the `Parameters` class.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class Parameters {
                inline Parameters(
                    NpToolkitCallback2 mCallback,
                    CommunicationId &id,
                    void *appData
                );
            }
        }
    }
}
```

Arguments

mCallback	The `NpToolkitCallback2` to be registered for events.
id	The main NP Communication ID assumed to be used for all services.
appData	A pointer to the application data which is returned when the `NpToolkitCallback2` callback is called.

Return Values

None

Description

A constructor for the `Parameters` class.

Default values:

`m_out` = `Cout`

`m_err` = `Cerr`

`m_ageRating` = 0 - indicates a universal game

`m_trial` = false - not a trial mode game

`m_id` = An empty `CommunicationId`. This will not work for games needing certain services.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: PostInvitationDataRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::PostInvitationDataRequest

Represents a request to post invitation data.

Definition

```
#include <np_toolkit.h>
struct PostInvitationDataRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to post invitation data.

Fields

Public Instance Fields

`char *data`	The invitation data. Optional.
`uint32_t dataSize`	The size of the invitation data. This only needs to be specified when setting invitation data.
`char m_reserved[3]`	Reserved.
`char message[512+1]`	The message content.
`SceNpSessionId npSessionId`	The invitation ID.
`uint32_t numOnlineIds`	The number of online IDs.
`SceNpOnlineId *onlineIds`	The online IDs.

Methods Summary

Methods	Description
`PostInvitationDataRequest`	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

PostInvitationDataRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline PostInvitationDataRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Presence

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Presence

The namespace containing presence functionality.

Definition

```
namespace Presence {}
```

Description

The namespace containing presence functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Presence::Interface	Allows a user's presence to be managed.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Presence::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Presence::Interface

Allows a user's presence to be managed.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

Allows a user's presence to be managed. This includes information on a user's current stage, location, or any other in-game information that may be of interest.

Methods Summary

Methods	Description
`getPresence`	Gets the presence information for a user's friend.
`setPresence`	Sets a user's presence.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

getPresence

Gets the presence information for a user's friend.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Presence {
                class Interface {
                    static int getPresence(
                        const PresenceRequest *request,
                        sce::Toolkit::NP::Utilities::Future
                        < PresenceInfo > *presenceResult,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

request	The request structure.
presenceResult	A Future object which will receive presence information for the specified user.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ERROR_INVALID_POINTER`	The operation failed because an invalid pointer was passed as an argument.

Description

Gets the presence information for a user's friend.

If the process is asynchronous, the application is notified by a `presenceGotInformation` Event.

Example

```
sce::Toolkit::NP::Utilities::Future<PresenceInfo> s_presenceInfo;
    SceUserServiceUserId userId = SCE_USER_SERVICE_USER_ID_INVALID;
    int ret = sceUserServiceGetInitialUser(&userId);
    if( ret < 0 ) {
        TTY::onScreenPrintf(MENU_TTY_TEXT_COLOUR_ERROR, "Error retrieving user
        id. ret = 0x%x\n", ret);
            return;
    }
```

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

```
        sce::Toolkit::NP::PresenceRequest presenceRequest;
        memset(&presenceRequest,0,sizeof(presenceRequest));

        strncpy(presenceRequest.onlineId.data,"SteveHd",strlen("SteveHd"));
        presenceRequest.presenceType =
        SCE_TOOLKIT_NP_PRESENCE_TYPE_PLATFORM_INFO;
        presenceRequest.userInfo.userId = userId;

    sce::Toolkit::NP::Presence::Interface::getPresence(
    &presenceRequest,&s_presenceInfo ,false);
```

Notes

Because the Future object of this function uses STL attributes, a call to the default `new` may be made when the attributes are set.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setPresence

Sets a user's presence.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Presence {
                class Interface {
                    static int setPresence(
                        const PresenceDetails *presDetails,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

presDetails	The status and data of a user.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_PRESENCE_STRING_TOO_LONG`	The operation failed because the string containing user's status was too long (over 63 characters).
`SCE_TOOLKIT_NP_PRESENCE_DATA_TOO_BIG`	The operation failed because the user data was too big (over 128 bytes).

Description

Sets a user's presence.

If the process is asynchronous the application is notified by a `presenceSet` Event.

Example

```
int ret = 0;
PresenceService::PresenceDetails testPres;
memset(&testPres, 0x00, sizeof(testPres));
testPres.status = "This is the NP Toolkit presence test\n";
int ret = Presence::Interface::setPresence(testPres, 1);
if(ret < 0){
    // Error handling
}
```

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::PresenceDetails

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::PresenceDetails

Holds the user's status string and data.

Definition

```
#include <np_toolkit.h>
struct PresenceDetails {};
```

Description

Holds the user's status string and data.

Fields

Public Instance Fields

`char` *`data [SCE_TOOLKIT_NP_IN_GAME_PRESENCE_DATA_SIZE_MAX]`*	Optional. The binary data for the user.
`uint32_t` *`presenceType`*	The type of presence.
`size_t` *`size`*	The size of the binary data.
`String` *`status`*	The status string of the user.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::PresenceRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::PresenceRequest

Represents a request to retrieve the presence information for a user's friend.

Definition

```
#include <np_toolkit.h>
struct PresenceRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to retrieve the presence information for a user's friend.

Fields

Public Instance Fields

`SceNpOnlineId` *`onlineId`*	The online ID of the user's friend.
`uint32_t` *`presencePlatform`*	The platform to get the presence for.
`uint32_t` *`presenceType`*	The type of presence to obtain information about.

Methods Summary

Methods	Description
`PresenceRequest`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

PresenceRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline PresenceRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
ProductBrowseParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ProductBrowseParams

Contains the parameters needed to browse a product.

Definition

```
#include <np_toolkit.h>
struct ProductBrowseParams : public sce::Toolkit::NP::Request {};
```

Description

Contains the parameters needed to browse a product.

Fields

Public Instance Fields

`bool inGame`	Used on the PlayStation®3 platform only. Added here for parity.
`void *memContainer`	Used on the PlayStation®3 platform only. Added here for parity. A value of `NULL` should be specified.
`SceChar8 padding[3]`	Padding of 3 bytes. Ensures alignment to a 4-byte boundary.
`char productId [SCE_TOOLKIT_NP_COMMERCE_PRODUCT_ID_LEN]`	The ID of the product to launch.
`uint32_t serviceLabel`	The PlayStation®4 service label.
`int userData`	Used on the PlayStation®3 platform only. Added here for parity.

Methods Summary

Methods	Description
ProductBrowseParams	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ProductBrowseParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ProductBrowseParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: ProductListInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ProductListInputParams

Contains information that is used to retrieve a list of products from a specific category.

Definition

```
#include <np_toolkit.h>
struct ProductListInputParams : public sce::Toolkit::NP::Request {};
```

Description

Contains information that is used to retrieve a list of products from a specific category.

Fields

Public Instance Fields

`char categoryId [SCE_TOOLKIT_NP_COMMERCE_CATEGORY_ID_LEN]`	The ID of the category to obtain the list of products for. Leave blank to get the list of products for the root category.
`uint32_t serviceLabel`	The PlayStation®4 service label.

Methods Summary

Methods	Description
`ProductListInputParams`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ProductListInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline ProductListInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: RangeOfRanksRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::RangeOfRanksRequest

Represents a request to retrieve a range of ranks.

Definition

```
#include <np_toolkit.h>
struct RangeOfRanksRequest : public sce::Toolkit::NP::RankingRequest {};
```

Description

Represents a request to retrieve a range of ranks.

Fields

Public Instance Fields

`SceNpScoreBoardId` *`boardId`*	The ID of the board from which the ranks need to be retrieved from.
`int32_t` *`range`*	The number of ranks, starting from *`startRank`*, to be retrieved.
`int32_t` *`startRank`*	The starting rank to be retrieved.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SCE CONFIDENTIAL

sce::Toolkit::NP::Ranking

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Ranking

The namespace containing ranking functionality.

Definition

```
namespace Ranking {}
```

Description

The namespace containing ranking functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Ranking::Interface	`Ranking` interface class.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Ranking::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Ranking::Interface

Ranking interface class.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

The ranking interface class has four functionalities. These are score registration, and retrieving a range of ranks, friends ranks and user ranks for display purposes.

Methods Summary

Methods	Description
displayFriendRank	Retrieves the rank of a friend of the user for display purposes.
displayFriendRank	Retrieves the rank of a friend of the user for display purposes.
displayRangeOfRanks	Retrieves a range of ranks for display purposes.
displayRangeOfRanks	Retrieves a range of ranks for display purposes.
displayUserRank	Retrieves user's own rank.
displayUserRank	Retrieves user's own rank.
rankingInit	Initializes the ranking service.
rankingTerm	Terminates the ranking service.
registerCache	Registers a ranking cache.
registerScore	Registers a user score.
registerScore	Registers a user score.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

displayFriendRank

Retrieves the rank of a friend of the user for display purposes.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int displayFriendRank(
                        sce::Toolkit::NP::Utilities::Future
                        < FriendsRankInformation > *friendScore,
                        SceNpScoreBoardId boardId,
                        bool async = false,
                        int userId = -1,
                        uint32_t serviceLabel = 0xffffffff
                    );
                }
            }
        }
    }
}
```

Arguments

friendScore	A `Future` object with a `sce::Toolkit::NP::TempRank` data type. Receives the friend's temporary rank back.
boardId	The ID of the board that the friend's rank is to be retrieved from.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to false so the function is blocking by default.
userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.
serviceLabel	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *friendScore*.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because the NP Toolkit library could not allocate the required internal memory.
`SCE_TOOLKIT_NP_RANKING_SLOT_FULL`	The operation failed because there were more than 32 simultaneous transactions going on at the time.
`rankingFriendsRetrieved`	In the case of an asynchronous operation, this event will be passed to an event callback to signify success.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Value	Description
`rankingFriendsRetrievedFail`	In the case of an asynchronous operation, this event will be passed to an event callback if the retrieval failed (general error).
`rankingFriendsRetrievedFailNoFriends`	In the case of an asynchronous operation, this event will be passed to an event callback if the retrieval failed because the user has no friends.
`rankingServerError`	In the case of an asynchronous operation, this event will be passed to an event callback if the retrieval failed because of a ranking server error. This could be a timeout for example.

Description

Retrieves the rank of a friend of the user for display purposes. When the operation has completed, the `Future` object passed to *`friendScore`* needs to be polled to see if the buffer is filled or not.

This function exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. A `FriendRankRequest` parameter should be used instead when calling `displayFriendRank()`.

Notes

The internal cache for friends has been disabled since the 3.10 SDK. This function will never use an internal cache system

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

displayFriendRank

Retrieves the rank of a friend of the user for display purposes.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int displayFriendRank(
                        const FriendRankRequest *rankRequest,
                        sce::Toolkit::NP::Utilities::Future
                        < FriendsRankInformation > *friendScore,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

rankRequest	Describes the details about the friends' ranking that are required.
friendScore	A `Future` object with a `sce::Toolkit::NP::TempRank` data type. Receives the friend's temporary rank back.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because an invalid pointer was passed to *friendScore*.
SCE_TOOLKIT_NP_OUT_OF_MEMORY	The operation failed because the NP Toolkit library could not allocate the required internal memory.
SCE_TOOLKIT_NP_RANKING_SLOT_FULL	The operation failed because there were more than 32 simultaneous transactions going on at the time.
rankingFriendsRetrieved	In the case of an asynchronous operation, this event will be passed to an event callback to signify success.
rankingFriendsRetrievedFail	In the case of an asynchronous operation, this event will be passed to an event callback if the retrieval failed (general error).
rankingFriendsRetrievedFailNoFriends	In the case of an asynchronous operation, this event will be passed to an event callback if the retrieval failed because the user has no friends.
rankingServerError	In the case of an asynchronous operation, this event will be passed to an event callback if the retrieval failed because of a ranking server error. This could be a timeout for example.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Description

Retrieves the rank of a friend of the user for display purposes. When the operation has completed, the `Future` object passed to *`friendScore`* needs to be polled to see if the buffer is filled or not.

Notes

The internal cache for friends has been disabled since the 3.10 SDK. This function will never use an internal cache system

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

displayRangeOfRanks

Retrieves a range of ranks for display purposes.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int displayRangeOfRanks(
                        sce::Toolkit::NP::Utilities::Future
                        < RankInformation > *score,
                        int boardId,
                        int startRank,
                        int range,
                        bool async = false,
                        int userId = -1,
                        uint32_t serviceLabel = 0xffffffff
                    );
                }
            }
        }
    }
}
```

Arguments

score	Output. A `Future` object with a `sce:Toolkit::NP::RankInformation` data type, which receives the list of ranks to be retrieved.
boardId	The board ID from which the ranks that needs to be retrieved.
startRank	The starting rank to be retrieved.
range	The number of ranks, starting from *startRank*, to be retrieved.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to false so the function is blocking by default.
userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.
serviceLabel	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *score*.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because the NP Toolkit library could not allocate the required internal memory.
`SCE_TOOLKIT_NP_RANKING_SLOT_FULL`	The operation failed because there were more than 32 simultaneous transactions going on at the time.
`rankingRangeRetrieved`	In the case of an asynchronous operation, this event will be passed to an event callback to signify success.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Value	Description
`rankingRangeRetrievedFail`	In the case of an asynchronous operation, this event will be passed to an event callback if the retrieval failed because the *`range`* or *`boardId`* arguments were invalid.
`rankingServerError`	In the case of an asynchronous operation, this event will be passed to an event callback if the retrieval failed because of a ranking server error. This could be a timeout for example.

Description

Retrieves a range of ranks to be displayed on the screen for the user. The maximum number of ranks that can be retrieved is 30. It would be difficult for the user to see any more than this on the screen. When the operation has completed, the `Future` object passed to *`score`* needs to be polled to see if the buffer is filled or not. If `registerCache()` has been called to register a ranking cache, the ranks will be held in the cache. This means that the next time the same scores are required they will be retrieved from a cache instead of pinging the server. This saves time and prevents server overload.

This function exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. A `RangeOfRanksRequest` parameter should be used instead when calling `displayRangeOfRanks()`.

Notes

If the function is called synchronously, it returns the number of ranks retrieved upon normal termination

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

displayRangeOfRanks

Retrieves a range of ranks for display purposes.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int displayRangeOfRanks(
                        const RangeOfRanksRequest *rangeRequest,
                        sce::Toolkit::NP::Utilities::Future
                        < RankInformation > *score,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

rangeRequest	Describes the range of ranks to retrieve.
score	Output. A `Future` object with a `sce:Toolkit::NP::RankInformation` data type, which receives the list of ranks to be retrieved.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *score*.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because the NP Toolkit library could not allocate the required internal memory.
`SCE_TOOLKIT_NP_RANKING_SLOT_FULL`	The operation failed because there were more than 32 simultaneous transactions going on at the time.
`rankingRangeRetrieved`	In the case of an asynchronous operation, this event will be passed to an event callback to signify success.
`rankingRangeRetrievedFail`	In the case of an asynchronous operation, this event will be passed to an event callback if the retrieval failed because the *range* or *boardId* arguments were invalid.
`rankingServerError`	In the case of an asynchronous operation, this event will be passed to an event callback if the retrieval failed because of a ranking server error. This could be a timeout for example.

Description

Retrieves a range of ranks to be displayed on the screen for the user. The maximum number of ranks that can be retrieved is 30. It would be difficult for the user to see any more than this on the screen.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

When the operation has completed, the `Future` object passed to *`score`* needs to be polled to see if the buffer is filled or not. If `registerCache()` has been called to register a ranking cache, the ranks will be held in the cache. This means that the next time the same scores are required they will be retrieved from a cache instead of pinging the server. This saves time and prevents server overload.

Notes

If the function is called synchronously, it returns the number of ranks retrieved upon normal termination

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

displayUserRank

Retrieves user's own rank.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int displayUserRank(
                        sce::Toolkit::NP::Utilities::Future
                            < sce::Toolkit::NP::UserRankInformation >
                            *currentUserScore,
                        SceNpId npId,
                        SceNpScoreBoardId boardId,
                        bool async = 0,
                        int userId = -1,
                        uint32_t serviceLabel = 0xffffffff
                    );
                }
            }
        }
    }
}
```

Arguments

currentUserScore	Output. A `Future` object with a `sce:Toolkit::NP::RankInformation` data type, which receives the ranks of the NP ID passed in.
npId	The `SceNPId` of the user to be retrieved.
boardId	The board ID from which the ranks that needs to be retrieved.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to false so the function is blocking by default.
userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.
serviceLabel	Not used. Added only to provide interface parity with the PlayStation®4 platform.

000004892117

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *currentUserScore*.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because the NP `Toolkit` library could not allocate the required internal memory.
`SCE_TOOLKIT_NP_RANKING_SLOT_FULL`	The operation failed because there were more than 32 simultaneous transactions going on at the time.
`rankingUserRankRetrieved`	In the case of an asynchronous operation, this event will be passed to an event callback to signal that user rank has been retrieved successfully

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Value	Description
`rankingUserRankRetrievedFailed`	In the case of an asynchronous operation, this event will be passed to an event callback to signal that user rank has been retrieved unsuccessfully

Description

This function retrieves the ranking information of the user whose `SceNpId` is passed in. You can retrieve current user's rank information by passing NULL to npId.

This function exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. A `UserRankRequest` parameter should be used instead when calling `displayUserRank()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

displayUserRank

Retrieves user's own rank.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int displayUserRank(
                        const UserRankRequest *rankRequest,
                        sce::Toolkit::NP::Utilities::Future
                            < sce::Toolkit::NP::UserRankInformation >
                            *currentUserScore,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

rankRequest	Describes the details about the user's ranking that are required.
currentUserScore	Output. A `Future` object with a `sce:Toolkit::NP::RankInformation` data type, which receives the ranks of the NP ID passed in.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to *`currentUserScore`*.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because the NP `Toolkit` library could not allocate the required internal memory.
`SCE_TOOLKIT_NP_RANKING_SLOT_FULL`	The operation failed because there were more than 32 simultaneous transactions going on at the time.
`rankingUserRankRetrieved`	In the case of an asynchronous operation, this event will be passed to an event callback to signal that user rank has been retrieved successfully
`rankingUserRankRetrievedFailed`	In the case of an asynchronous operation, this event will be passed to an event callback to signal that user rank has been retrieved unsuccessfully

Description

This function retrieves the ranking information of the user whose `SceNpId` is passed in. You can retrieve current user's rank information by specifying an NP ID of NULL.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

rankingInit

Initializes the ranking service.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int rankingInit();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The ranking service was successfully initialized.
SCE_TOOLKIT_NP_RANKING_ALREADY_INITED	The ranking service has already been initialised.

Description

Initializes the ranking service. There no need to call this function when the NP Toolkit library is first time initialized and the ranking service is automatically brought up. It only needs to be called when the ranking service has been manually terminated by calling `rankingTerm()`. This function calls `sceNpScoreInit()` and `sceNpScoreCreateTitleCtx()` internally and loads the `SCE_SYSMODULE_NP_SCORE_RANKING` module as well. For error messages, please refer to the descriptions for `sceNpScoreInit()` and `sceNpScoreCreateTitleCtx()` in the *NP_ScoreRanking-Reference* document. Note that this is a synchronous function.

Initializes the ranking service. There is no need to call this function. It only exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. This function will be called automatically when one of the NP Toolkit `Ranking` functions are called.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

rankingTerm

Terminates the ranking service.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int rankingTerm();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The ranking service was successfully terminated.

Description

Terminates the ranking service. This function free up ranking cache memory registered by the `registerCache()`, and calls `sceNpScoreDeleteTitleCtx()` and `sceNpScoreTerm()` internally. The `SCE_SYSMODULE_NP_SCORE_RANKING` module will be unloaded as well. For error messages, please refer to the descriptions for `sceNpScoreDestroyTitleCtx()` and `sceNpScoreTerm()` in the *NP_ScoreRanking-Reference* document. Note that this is a synchronous function.

Terminates the ranking service. There is no need to call this function. It only exists to maintain backwards compatibility with previous version of the PlayStation®Vita platform. Instead of this function, `terminateService()` should be called with `ranking` specified for the *`serviceType`* argument.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

registerCache

Registers a ranking cache.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int registerCache(
                        int boardLineCount,
                        int writeLineCount,
                        int rangeLineCount,
                        bool friendCache = false
                    );
                }
            }
        }
    }
}
```

Arguments

boardLineCount	The number of lines to allocate for the board cache.
writeLineCount	The number of lines to allocate for the write cache. This mainly for registering scores.
rangeLineCount	The number of lines to allocate for the read cache. This is used for range rank requests.
friendCache	A flag that specifies whether to create a friend cache. This should always be set to false as the friend cache has been disabled since the 3.10 SDK.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_FAILED_ALLOCATE`	The operation failed because memory could not be allocated for the cache.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because the NP `Toolkit` library could not allocate the required internal memory.

Description

Registers a ranking cache. There are 3 types of ranking cache: a board cache where the board configuration is held, a write cache where scores awaiting server registration are held and a read cache where a range of scores are held. The table below contains details on the three types of cache:

Board Cache	Each line is 24 bytes.
Write Cache	Each line is 289 bytes.
Read Cache	Each line is 416 bytes.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

registerScore

Registers a user score.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int registerScore(
                        sce::Toolkit::NP::Utilities::Future
                            < TempRank > *tempRank,
                        const RegisterScore *score,
                        bool async = true,
                        int userId = -1,
                        uint32_t serviceLabel = 0xffffffff
                    );
                }
            }
        }
    }
}
```

Arguments

tempRank	A Future object with a sce::Toolkit::NP::TempRank datatype. Receives a temporary rank back.
score	The user's score to be registered.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.
userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.
serviceLabel	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_RANKING_SLOT_FULL	The operation failed because there were more than 32 simultaneous transactions going on at the time.
SCE_TOOLKIT_NP_OUT_OF_MEMORY	The operation failed because the NP Toolkit library could not allocate the required internal memory.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because the score was lower than user's current high score.
rankingScoreRegistered	In the case of an asynchronous operation, this event will be passed to an event callback to signify success.
rankingScoreRegisteredFail	In the case of an asynchronous operation, this event will be passed to an event callback to signify failure. For example, this could be due to server errors.
rankingScoreRegisteredFailNotBest	In the case of an asynchronous operation, this event will be passed to an event callback if score registration failed because the score was not the best score.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Value	Description
`rankingServerError`	In the case of an asynchronous operation, this event will be passed to an event callback if score registration failed because of a ranking server error. This could be a timeout for example.

Description

Registers a user score. The `sce::Toolkit::NP::RegisterScore` data type should be used to pass in the score, comments and game information to be registered. There is server delay in processing the user's actual rank after a score has been registered, and a `sce::Toolkit::NP::TempRank` `Future` object can be used to receive a temporary rank back upon successful completion of the operation. If a ranking cache is being used, calling the `registerCache()` function will cause the ranking service to cache scores that there has been an attempt to register. Before pinging the server to register the score, the ranking service will check new scores against the cache to see if an attempt to register the new score has been made, or if the new score is lower than the one in the cache. This will save time and resources by avoiding trying to ping the server with lower scores or scores that there has already been an attempt to register.

This function exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. A `RegisterScoreRequest` parameter should be used instead when calling `registerScore()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

registerScore

Registers a user score.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Ranking {
                class Interface {
                    static int registerScore(
                        const RegisterScoreRequest *scoreRequest,
                        sce::Toolkit::NP::Utilities::Future
                            < TempRank > *tempRank,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

scoreRequest	The user's score to be registered.
tempRank	A `Future` object with a `sce::Toolkit::NP::TempRank` datatype. Receives a temporary rank back.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_RANKING_SLOT_FULL`	The operation failed because there were more than 32 simultaneous transactions going on at the time.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	The operation failed because the NP Toolkit library could not allocate the required internal memory.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the score was lower than user's current high score.
`rankingScoreRegistered`	In the case of an asynchronous operation, this event will be passed to an event callback to signify success.
`rankingScoreRegisteredFail`	In the case of an asynchronous operation, this event will be passed to an event callback to signify failure. For example, this could be due to server errors.
`rankingScoreRegisteredFailNotBest`	In the case of an asynchronous operation, this event will be passed to an event callback if score registration failed because the score was not the best score.
`rankingServerError`	In the case of an asynchronous operation, this event will be passed to an event callback if score registration failed because of a ranking server error. This could be a timeout for example.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Description

Registers a user score. The `sce::Toolkit::NP::RegisterScore` data type should be used to pass in the score, comments and game information to be registered. There is server delay in processing the user's actual rank after a score has been registered, and a `sce::Toolkit::NP::TempRank` `Future` object can be used to receive a temporary rank back upon successful completion of the operation. If a ranking cache is being used, calling the `registerCache()` function will cause the ranking service to cache scores that there has been an attempt to register. Before pinging the server to register the score, the ranking service will check new scores against the cache to see if an attempt to register the new score has been made, or if the new score is lower than the one in the cache. This will save time and resources by avoiding trying to ping the server with lower scores or scores that there has already been an attempt to register.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::RankingRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::RankingRequest

Represents the service label of the ranking service.

Definition

```
#include <np_toolkit.h>
struct RankingRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents the service label of the ranking service. This structure is used as a base for all other ranking input structures.

Fields

Public Instance Fields

`uint32_t serviceLabel`	This attribute is for future expansions. To register a service label use `registerServiceLabel()` in NP Toolkit.

Methods Summary

Methods	Description
RankingRequest	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

RankingRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline RankingRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::RegisterScoreRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::RegisterScoreRequest

Represents a request to register a user's score.

Definition

```
#include <np_toolkit.h>
struct RegisterScoreRequest : public sce::Toolkit::NP::RankingRequest {};
```

Description

Represents a request to register a user's score.

Fields

Public Instance Fields

`SceNpScoreBoardId` *`boardId`*	The ID of the board on which the score is to be registered.
`SceNpScoreComment` *`comment`*	A comment to be passed along with the score.
`SceNpScoreGameInfo` *`gameInfo`*	Application specific information to be passed along with the score.
`SceNpScoreValue` *`score`*	The user's score.
`SceNpScoreRankNumber` *`tmpRank`*	The user's temporary rank.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
RegisterTrophyRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::RegisterTrophyRequest

Represents a request to register a trophy package for a specific user.

Definition

```
#include <np_toolkit.h>
struct RegisterTrophyRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to register a trophy package for a specific user. It is required by the `Trophy::Interface::trophyRegisterSet()` function.

Fields

Public Instance Fields

`bool cacheGameIcon`	Not used. Added only to provide interface parity with the PlayStation®4 platform.
`bool cacheGroupsIcons`	Not used. Added only to provide interface parity with the PlayStation®4 platform.
`bool cacheIcons`	A flag that specifies whether the trophy icons are to be cached or not.
`bool cacheTrophiesIcons`	Not used. Added only to provide interface parity with the PlayStation®4 platform.
`bool cacheTrophyList`	A flag that specifies whether the trophy list is to be cached or not.
`int hddSizeInBytes`	Not used. Added only to provide interface parity with the PlayStation®3 platform.
`char reserved[3]`	Reserved.
`int saveDataInBytes`	Not used. Added only to provide interface parity with the PlayStation®3 platform.
`uint32_t serviceLabel`	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Methods Summary

Methods	Description
`RegisterTrophyRequest`	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

RegisterTrophyRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline RegisterTrophyRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Request

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Request

The base structure for a request, which contains the information required for all types of request.

Definition

```
#include <np_toolkit.h>
struct Request {};
```

Description

The base structure for a request, which contains the information required for all types of request.
This structure exists only to provide interface parity with the PlayStation®4 platform.

Fields

Public Instance Fields

`char` *`reserved[4]`*	Reserved.
`UserInfo` *`userInfo`*	Not used. Added only to provide interface parity with the PlayStation®4 platform.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
RetrieveTrophyGameRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::RetrieveTrophyGameRequest

Represents a request to retrieve information about a games's trophy set.

Definition

```
#include <np_toolkit.h>
struct RetrieveTrophyGameRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to retrieve information about a games's trophy set.

It is required by the `Trophy::Interface::trophyRetrieveGame()` function.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: RetrieveTrophyGroupRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::RetrieveTrophyGroupRequest

Represents a request to retrieve trophy group information for a user.

Definition

```
#include <np_toolkit.h>
struct RetrieveTrophyGroupRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to retrieve trophy group information for a user.

It is required by the `Trophy::Interface::trophyRetrieveGroups()` function.

Fields

Public Instance Fields

`SceNpTrophyGroupId` *`groupId`*	The ID of the group to retrieve.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: RetrieveTrophyListRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::RetrieveTrophyListRequest

Represents a request to retrieve a list of detailed trophy information for all the trophies in a games's trophy set.

Definition

```
#include <np_toolkit.h>
struct RetrieveTrophyListRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to retrieve a list of detailed trophy information for all the trophies in a games's trophy set.

It is required by the `Trophy::Interface::trophyRetrieveList()` function.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
RetrieveUserTrophyProgressRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP:: RetrieveUserTrophyProgressRequest

Represents a request to retrieve a user's trophy progress.

Definition

```
#include <np_toolkit.h>
struct RetrieveUserTrophyProgressRequest : public sce::Toolkit::NP::Request
{};
```

Description

Represents a request to retrieve a user's trophy progress.

RetrieveUserTrophyProgressRequest objects are passed as arguments to the `Trophy::Interface::trophyRetrieveUserProgress()` function.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: SearchNpSessionRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::SearchNpSessionRequest

Represents a request to search sessions on the Session server.

Definition

```
#include <np_toolkit.h>
struct SearchNpSessionRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to search sessions on the Session server.

Fields

Public Instance Fields

`SceNpOnlineId` *`onlineId`*	The online ID of the user whose session is to be searched.

Methods Summary

Methods	Description
SearchNpSessionRequest	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

SearchNpSessionRequest

The default constructor.

Definition

```
        #include <np_toolkit.h>
        inline SearchNpSessionRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: SearchSessionsRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::SearchSessionsRequest

A search descriptor to search for a session.

Definition

```
#include <np_toolkit.h>
struct SearchSessionsRequest : public sce::Toolkit::NP::Request {};
```

Description

A search descriptor to search for a session.

Fields

Public Instance Fields

`SceNpId *friendsList`	The list of friends whose session is to be searched for.
`int32_t numFriends`	The number of friends in the list of friends to search for.
`int32_t numSearchFilters`	The number of search filters in the search filter list.
`uint8_t padding[2]`	Padding.
`SessionRequestAttribute *searchFilters`	The list of search filters.
`SceToolkitNpSessionSearchFlag searchFlags`	The flags to specify type of session to search for (`SCE_TOOLKIT_NP_SEARCH_*`).
`SceNpMatching2ServerId serverId`	The server ID. Used to specify if there is a specific server on which to search for the session.
`uint32_t startIndex`	The position to start searching for the session.
`SceNpMatching2WorldId worldId`	The world ID. Used to specify if there is a specific world in which to search for the session.

Methods Summary

Methods	Description
SearchSessionsRequest	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

SearchSessionsRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline SearchSessionsRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::ServiceId

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::ServiceId

Wraps the string being used as a NP Service ID for commerce, ticketing, etc.

Definition

```
#include <np_toolkit.h>
class ServiceId {};
```

Description

Wraps the string being used as a NP Service ID for commerce, ticketing, etc.

Methods Summary

Methods	Description
getId	Gets the NP Service ID as a string.
operator<	Less than comparison operator needed for some storage types.
ServiceId	The default constructor for the ServiceId class.
ServiceId	A constructor for the ServiceId class, which sets the NP Service ID.
setId	Sets the NP Service ID of this object.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

ServiceId

The default constructor for the ServiceId class.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class ServiceId {
                inline ServiceId();
            }
        }
    }
}
```

Arguments

None

Return Values

None

Description

The default constructor for the ServiceId class.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

ServiceId

A constructor for the ServiceId class, which sets the NP Service ID.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class ServiceId {
                inline ServiceId(
                    const String &id
                );
            }
        }
    }
}
```

Arguments

id	A string representing the NP Service ID.

Return Values

None

Description

A constructor for the ServiceId class, which sets the NP Service ID.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Operator Methods

operator<

Less than comparison operator needed for some storage types.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class ServiceId {
                inline bool operator<(
                    const ServiceId &rhs
                ) const;
            }
        }
    }
}
```

Arguments

rhs	The NP Service ID to compare the stored ID against.

Return Values

If the supplied NP Service ID is greater than the stored ID, a value of true is returned. A value of false is returned if this is not the case.

Description

Less than comparison operator needed for some storage types.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

getId

Gets the NP Service ID as a string.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            class ServiceId {
                inline const String &getId() const;
            }
        }
    }
}
```

Arguments

None

Return Values

The NP Service ID as a string reference.

Description

Gets the NP Service ID as a string.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

setId

Sets the NP Service ID of this object.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         class ServiceId {
            inline void setId(
               const String &id
            );
         }
      }
   }
}
```

Arguments

id	A string representing the new NP Service ID.

Return Values

None

Description

Sets the NP Service ID of this object.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::SessionInformation

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::SessionInformation

Contains information about a session.

Definition

```
#include <np_toolkit.h>
struct SessionInformation {};
```

Description

Contains information about a session.

Fields

Public Instance Fields

SceNpMatching2ContextId *matchingContext*	The Matching2 context ID. This should be used when using the NP Matching2 library chat function to send messages to members in the current session.
int32_t *maxMembers*	The maximum number of members in the session.
SessionMemberList *memberData*	The information about the members of the session. Note that this information is only populated when a user has created/joined the session.
NpSessionInformation *npSession*	The information about the NP session.
int32_t *numMembers*	The current number of members in the session.
int32_t *numSessionAttributes*	The number of attributes in the session.
SceNpMatching2RoomId *roomId*	The ID of the current session.
SceNpMatching2ServerId *serverId*	The server ID for the session.
SessionAttributeList *sessionAttributes*	The list of attributes for the session.
char *sessionName [SCE_TOOLKIT_NP_SESSION_NAME_MAX_SIZE]*	The session name.
SessionSlotsInfo *slotsInformation*	Information about number of slots in the session open for public or private use.
SceNpMatching2WorldId *worldId*	The world ID for the session.

Methods Summary

Methods	Description
reset	Resets the session information.
SessionInformation	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

SessionInformation

The default constructor.

Definition

```
#include <np_toolkit.h>
inline SessionInformation();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

reset

Resets the session information.

Definition

```
#include <np_toolkit.h>
inline void reset();
```

Arguments

None

Return Values

None

Description

Resets the session information.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::SessionMember

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::SessionMember

Provides information about the member in a session.

Definition

```
#include <np_toolkit.h>
struct SessionMember {};
```

Description

Provides information about the member in a session.

Fields

Public Instance Fields

`SceRtcTick` *`joinDate`*	The date and time the user joined the session.
`SessionAttributeList` *`memberAttributes`*	The attributes related to the member.
`MemberAddress` *`memberConnInfo`*	The member's address.
`SceToolkitNpSessionMemberFlag` *`memberFlag`*	Specifies whether this member is the owner, another member or the current user.
`SceNpMatching2RoomMemberId` *`memberId`*	The room member ID of the member.
`SceNpMatching2NatType` *`natType`*	The member's NAT Type.
`SceNpId` *`userInfo`*	The NP ID information of the user.

Methods Summary

Methods	Description
`SessionMember`	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

SessionMember

The default constructor.

Definition

```
#include <np_toolkit.h>
inline SessionMember();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Sessions

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Sessions

The namespace containing session functionality on the PlayStation®4.

Definition

```
namespace Sessions {}
```

Description

The namespace containing session functionality on the PlayStation®4.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Sessions::Interface	The session interface allows users to find other players and game sessions for online play.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Sessions::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Sessions::Interface

The session interface allows users to find other players and game sessions for online play.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

The session interface allows users to find other players and game sessions for online play.

Methods Summary

Methods	Description
create	Creates a session on the NP Session server.
getChangeableSessionData	Gets session changeable Data on the NP Session server.
getInfo	Gets information about a session on the NP Session server.
getInvitationData	Gets session data from a session invite received by the user.
getInvitationInfo	Gets information about a session invite received by the user.
getInvitationList	Gets a list of the invites received by the user.
getSessionData	Gets session Data on the NP Session server.
invite	Sends a session invite to a friend of the user.
join	Joins a session on the NP matching server.
leave	Leaves a currently joined/created session.
postInvitationData	Posts an invitation to the session.
search	Searches for a session on the NP Session server.
setInvitationDataUsedFlag	Sets the flag that specifies that an invitation's data has been used.
update	Updates the current session information.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

create

Creates a session on the NP Session server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int create(
                        const CreateNpSessionRequest *sessionRequest,
                        sce::Toolkit::NP::Utilities::Future
                            < NpSessionInformation > *sessionInformation,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

sessionRequest	A structure that describes the session details. This will be assigned to a room on successful creation.
sessionInformation	Output. Receives the session information upon successful creation of the session.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.
SCE_TOOLKIT_NP_ERROR_MATCHING_SERVICE_BUSY	The operation failed because the matching service is busy processing a previous request.
SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_ALREADY_ACTIVE	The operation failed because the user is already in a session. They must leave a session in order to join or create a new session.
other	An NP Library Error Code.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Description

Creates a session on the NP Session server. The session creation process is kicked off on a different thread.

Non-Blocking (*async* = true) If the process was not able to kick off, the function returns an appropriate error code. On completion of the process the application is notified using a `npSessionCreateResult` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking (*async* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve session information using the `get()` method of the `Future` object.

Notes

If this function is called from the main thread, it should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

See Also

`searchSessions()`, `joinSession()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getChangeableSessionData

Gets session changeable Data on the NP Session server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int getChangeableSessionData(
                        const GetInfoNpSessionRequest *sessionInfoRequest,
                        sce::Toolkit::NP::Utilities::Future
                            < MessageAttachment > *sessionData,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

sessionInfoRequest	A structure which contains information about session whose changeable Data is to be retrieved.
sessionData	Output. Receives the session data when a request is successfully completed.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.
SCE_TOOLKIT_NP_ERROR_MATCHING_SERVICE_BUSY	The operation failed because the matching service is busy processing a previous request.
SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_ALREADY_ACTIVE	The operation failed because the user is already in a session. They must leave a session in order to join or create a new session.
other	An NP Library Error Code.

Description

Gets session changeable Data on the NP Session server. This function kicks off a process that requests to get information about a session on a different thread.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On completion of the process, the application is notified by a `npSessionGetChangeableSessionDataResult` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking Process (*async* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

See Also

`search()`, `create()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getInfo

Gets information about a session on the NP Session server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int getInfo(
                        const GetInfoNpSessionRequest *sessionInfoRequest,
                        sce::Toolkit::NP::Utilities::Future
                            < NpSessionDetailedInformation > *sessionInformation,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

sessionInfoRequest	A structure which contains information about session whose information is to be retrieved.
sessionInformation	Output. Receives the session information when a request is successfully completed.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.
`SCE_TOOLKIT_NP_ERROR_MATCHING_SERVICE_BUSY`	The operation failed because the matching service is busy processing a previous request.
`SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_ALREADY_ACTIVE`	The operation failed because the user is already in a session. They must leave a session in order to join or create a new session.
`other`	An NP Library Error Code.

Description

Gets information about a session on the NP Session server. This function kicks off a process that requests to get information about a session on a different thread.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On completion of the process the application is notified by a `npSessionGetInfoResult` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking Process (*async* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

Because the Future object of this function uses STL attributes, a call to the default `new` may be made when the attributes are set.

See Also

`search()`, `create()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getInvitationData

Gets session data from a session invite received by the user.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int getInvitationData(
                        const InvitationDataRequest *infoRequest,
                        sce::Toolkit::NP::Utilities::Future
                            < MessageAttachment > *sessionData,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

infoRequest	The details about the request for session data from a received session invite.
sessionData	Output. Receives the session data when the request is successfully completed.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was kicked off successfully.
SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_DOES_NOT_EXIST	The operation failed because the session that an invite was sent for is invalid.

Description

Gets session data from a session invite received by the user.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On successful completion of the process the application is notified by a `npSessionInviteGetDataResult` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getInvitationInfo

Gets information about a session invite received by the user.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int getInvitationInfo(
                        const InvitationInfoRequest *requestInfo,
                        sce::Toolkit::NP::Utilities::Future
                            < NpSessionInvitationInfo > *inviteInfo,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

requestInfo	The details about the request for more information about a received session invite.
inviteInfo	Output. Receives the invitation information when the request is successfully completed.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_DOES_NOT_EXIST`	The operation failed because the session that an invite was sent for is invalid.

Description

Gets information about a session invite received by the user.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On successful completion of the process the application is notified by a `npSessionInviteGetInfoResult` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Notes

Because the Future object of this function uses STL attributes, a call to the default `new` may be made when the attributes are set.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getInvitationList

Gets a list of the invites received by the user.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int getInvitationList(
                        const InvitationListRequest *requestInfo,
                        sce::Toolkit::NP::Utilities::Future
                            < NpSessionInvitationInfoList >
                            *sessionInvitationList,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

requestInfo	The details about the request for a list of received session invites.
sessionInvitationList	Output. Receives the list of invitations when the request is successfully completed.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because a pointer was invalid.

Description

Gets a list of the invites received by the user.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On successful completion of the process the application is notified by a `npSessionInviteGetInfoListResult` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Notes

Because the Future object of this function uses STL attributes, a call to the default `new` may be made when the attributes are set.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getSessionData

Get session Data on the NP Session server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int getSessionData(
                        const GetInfoNpSessionRequest *sessionInfoRequest,
                        sce::Toolkit::NP::Utilities::Future
                            < MessageAttachment > *sessionData,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

sessionInfoRequest	A structure which contains information about session whose data is to be retrieved.
sessionData	Output. Receives the session data when a request is successfully completed.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.
`SCE_TOOLKIT_NP_ERROR_MATCHING_SERVICE_BUSY`	The operation failed because the matching service is busy processing a previous request.
`SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_ALREADY_ACTIVE`	The operation failed because the user is already in a session. They must leave a session in order to join or create a new session.
`other`	An NP Library Error Code.

Description

Gets session Data on the NP Session server. This function kicks off a process that requests to get information about a session on a different thread.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On completion of the process the application is notified by a `npSessionGetSessionDataResult` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking Process (*async* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

See Also

`search()`, `create()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

invite

Sends a session invite to a friend of the user.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int invite(
                        const InviteNpSessionRequest *currentSession,
                        const InviteMessage *msg
                    );
                }
            }
        }
    }
}
```

Arguments

currentSession	A pointer to the session the user currently is in.
msg	The user-specified message.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_DOES_NOT_EXIST`	The operation failed because the session that an invite was sent for is invalid.

Description

Sends a session invite to a friend of the user.

Notes

This triggers a invitation dialog. Because of this, the Common Dialog library needs to have first been initialized. If application intends to send a non-GUI invite or an invite with specific invitation data, please refer to `postInvitationData()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

join

Joins a session on the NP matching server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int join(
                        const JoinNpSessionRequest *sessionJoinRequest,
                        sce::Toolkit::NP::Utilities::Future
                            < NpSessionInformation > *sessionInformation,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

sessionJoinRequest	A structure which contains information about session to be joined.
sessionInformation	Output. Receives the session information when a session is successfully joined.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.
SCE_TOOLKIT_NP_ERROR_MATCHING_SERVICE_BUSY	The operation failed because the matching service is busy processing a previous request.
SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_ALREADY_ACTIVE	The operation failed because the user is already in a session. They must leave a session in order to join or create a new session.
other	An NP Library Error Code.

Description

Joins a session on the NP matching server. This function kicks off a process that requests to join a session on a different thread.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Non-Blocking Process (*`async`* = true) If the process was not able to kick off, then the function returns an appropriate error code. On completion of the process the application is notified by a `npSessionJoinResult` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking Process (*`async`* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

See Also

`search()`, `create()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

leave

Leaves a currently joined/created session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int leave(
                        const LeaveNpSessionRequest *leavingSession,
                        sce::Toolkit::NP::Utilities::Future
                            < int > *processResult,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

leavingSession	A pointer to the session the user wants to leave. Set to `NULL` if the session is not known.
processResult	Output. Receives the result of this process when the session has been successfully left.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.
`SCE_TOOLKIT_NP_ERROR_MATCHING_SERVICE_BUSY`	The operation failed because the matching service is busy processing a previous request.
`SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_DOES_NOT_EXIST`	The operation failed because the session which the user is trying to leave does not exist.
`other`	An NP Library Error Code.

Description

Leaves a currently joined/created session. This request is processed on a different thread.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On successful completion of the process the application is notified by a `npSessionLeaveResult` Event. The application can then verify whether there was an error during

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking Process (*async* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed. Even when an application receives a `matchingSessionError` `Event`, the session will still be destroyed and therefore no longer be valid.

See Also

`searchSessions()`, `createSession()`, `joinSession()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

postInvitationData

Posts an invitation to the session.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int postInvitationData(
                        const PostInvitationDataRequest *infoRequest,
                        sce::Toolkit::NP::Utilities::Future< int > *result,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

infoRequest	The details about the request post an invitation to the session.
result	Output. Receives the session data when the request is successfully completed.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was kicked off successfully.

Description

Posts an invitation to the session. An application can use this API to send an invitation with specific invite data.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On successful completion of the process the application is notified by a `npSessionInvitePostInvitationResult` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

search

Searches for a session on the NP Session server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int search(
                        const SearchNpSessionRequest *searchRequest,
                        sce::Toolkit::NP::Utilities::Future
                            < NpSessionsList > *sessionList,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

searchRequest	Specifies the type of a session to look for.
sessionList	Output. Receives the results of the search.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.
SCE_TOOLKIT_NP_ERROR_MATCHING_SERVICE_BUSY	The operation failed because the matching service is busy processing a previous request.
other	An NP Library Error Code.

Description

Searches for a session on the NP Session server. On calling, this function kicks off a process that searches for a session on a NP Toolkit library thread.

Non-Blocking Process (*async* = true) If the process was not able to kick off, the function returns an appropriate error code. On completion of the process, the application is notified by a `npSessionSearchResult` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Blocking Process (*async* = false) The function blocks until a result is returned. If the function is successful, it returns `SCE_TOOLKIT_NP_SUCCESS`; otherwise an appropriate error code is returned.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

On successful completion of the operation, the application can retrieve the result using the `get()` method of the `Future` object.

Notes

If calling from main thread, the function should always be non-blocking. If calling asynchronously, the `Future` object should be valid until the callback of the event is processed.

Because the Future object of this function uses STL attributes, a call to the default `new` may be made when the attributes are set.

See Also

`createSession()`, `joinSession()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setInvitationDataUsedFlag

Sets the flag that specifies that an invitation's data has been used.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int setInvitationDataUsedFlag(
                        const InvitationDataRequest *infoRequest,
                        sce::Toolkit::NP::Utilities::Future< int > *result,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

infoRequest	The details about the request to set the used flag for some invitation data.
result	Output. Receives the session data when the request is successfully completed.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was kicked off successfully.

Description

Sets the flag that specifies that an invitation's data has been used.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On successful completion of the process the application is notified by a `npSessionInviteSetDataUsedResult` Event. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

update

Updates the current session information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sessions {
                class Interface {
                    static int update(
                        const UpdateNpSessionRequest *updateSessionRequest,
                        sce::Toolkit::NP::Utilities::Future
                            < int > *processResult,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

updateSessionRequest	Input/Output. The current session information structure to be updated.
processResult	Output. Receives the result of an event.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_KICKEDOUT	The operation failed because the user has been kicked out of the matching session.
SCE_TOOLKIT_NP_ERROR_MATCHING_SERVICE_BUSY	The operation failed because the matching service is busy processing a previous request.
SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_DOES_NOT_EXIST	The operation failed because the session the user was trying to update does not exist.
SCE_TOOLKIT_NP_ERROR_MATCHING_SESSION_ROOM_DESTROYED	The operation failed because the session the user was in has been destroyed.

Description

Updates the current session information.

Non-Blocking Process (*async* = true) If the process was not able to kick off, then the function returns an appropriate error code. On successful completion of the process the application is notified by a

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

`npSessionUpdateResult` `Event`. The application can then verify whether there was an error during the process or whether the process was successful by using the `hasError()` or `hasResult()` method of the `Future` object. If an error has occurred, the application can get the error code using the `getError()` method.

Notes

On receiving an error, the application should clear the current session.

This operation should be called from the same thread on which the callback was received (the NP Toolkit thread).

See Also

`searchSessions()`, `createSession()`, `joinSession()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Sns

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Sns

The namespace containing the PlayStation™Network SNS service.

Definition

```
namespace Sns {}
```

Description

The namespace containing the PlayStation™Network SNS service.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Sns::Interface	The SNS interface allows users to post a message to Facebook.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Sns::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Sns::Interface

The SNS interface allows users to post a message to Facebook.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

The SNS interface allows users to post a message to Facebook.

Methods Summary

Methods	Description
postMessageFb	Posts a message to the user's Facebook "Wall".
setAppIdFb	Sets the Facebook Application ID.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

postMessageFb

Posts a message to the user’s Facebook “Wall”.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sns {
                class Interface {
                    static int postMessageFb(
                        const SnsPostFacebook &msgDetails
                    );
                }
            }
        }
    }
}
```

Arguments

msgDetails	A reference to a structure containing information about the message to post.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_SNS_ACCESS_TOKEN_ERROR`	The operation failed because an error occurred when obtaining an access token from Facebook.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library was not initialized.
`SCE_TOOLKIT_NP_SNS_INVALID_MESSAGE`	The operation failed because the contents of the *msgDetails* parameter was invalid.

Description

Posts a message to the user’s Facebook “Wall”.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setAppIdFb

Sets the Facebook Application ID.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Sns {
                class Interface {
                    static int setAppIdFb(
                        const uint64_t &id
                    );
                }
            }
        }
    }
}
```

Arguments

id	The value to set the ID to.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library was not initialized.

Description

Sets the Facebook Application ID, which is stored in the SNS Service. This ID is supplied by Facebook when an application is created.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::SnsPostFacebook

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::SnsPostFacebook

Holds the necessary information needed to post a message to a Facebook user's wall.

Definition

```
#include <np_toolkit.h>
struct SnsPostFacebook {};
```

Description

Holds the necessary information needed to post a message to a Facebook user's wall. It is comprised of the PhotoFb and ActionLinkFb structures, and in addition it contains a variable to hold the user's text which will accompany the post.

Fields

Public Instance Fields

`ActionLinkFb` *`actionLink`*	The action link information.
`PhotoFb` *`photo`*	The photo information which includes the title, caption and description.
`String` *`userText`*	A string to hold the text that is input by the user to personalize the stream story.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Ticket

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Ticket

Represents a ticket.

Definition

```
#include <np_toolkit.h>
struct Ticket {};
```

Description

Represents a ticket.

Fields

Public Instance Fields

`void *buffer`	The buffer in which the ticket is stored.
`uint32_t size`	The size of the ticket. The maximum file size of the ticket is defined by `SCE_NP_TICKET_MAX_SIZE`.

Methods Summary

Methods	Description
Ticket	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

Ticket

The default constructor.

Definition

```
#include <np_toolkit.h>
inline Ticket();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Trophy

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Trophy

The namespace containing trophy functionality.

Definition

```
namespace Trophy {}
```

Description

The namespace containing trophy functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Trophy::Interface	`Trophy` interface class.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Trophy::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Trophy::Interface

`Trophy` interface class.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

This class contains the set of static methods for managing trophies.

Methods Summary

Methods	Description
`trophyInit`	Initializes the trophy service.
`trophyRegisterSet`	Registers a trophy set.
`trophyRegisterSet`	Registers a trophy set.
`trophyRetrieveGame`	Retrieves overview information about a game's trophy set.
`trophyRetrieveGame`	Retrieves overview information about a game's trophy set.
`trophyRetrieveGroups`	Retrieves trophy group information.
`trophyRetrieveGroups`	Retrieves trophy group information.
`trophyRetrieveList`	Retrieves a trophy list.
`trophyRetrieveList`	Retrieves a trophy list.
`trophyRetrieveProgress`	Retrieves the user's trophy progress.
`trophyRetrieveProgress`	Retrieves the user's trophy progress.
`trophyTerm`	Terminates the trophy service.
`trophyUnlock`	Unlocks a particular trophy.
`trophyUnlock`	Unlocks a particular trophy.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

trophyInit

Initializes the trophy service.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyInit();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Initializes the trophy service. This function does not need to be called when the NP Toolkit library first sets up the service. It should only be called after the trophy service has been explicitly terminated by calling `trophyTerm()`. Internally, the function loads the trophy modules and calls the following functions: `sceNpTrophyCreateContext()` and `sceNpTrophyCreateHandle()`. For the related error messages, please refer to these function's descriptions in the *NP Trophy Library Reference* document.

There is no need to call this function. It only exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. This function will be called automatically when one of the NP Toolkit `Trophy` functions are called.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

trophyRegisterSet

Registers a trophy set.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyRegisterSet(
                        bool cacheTrophyList,
                        bool cacheIcons,
                        int saveDataInBytes = 0,
                        int hddSizeInBytes = 0,
                        int userId = -1,
                        uint32_t serviceLabel = 0xFFFFFFFF
                    );
                }
            }
        }
    }
}
```

Arguments

cacheTrophyList	A flag that specifies whether the trophy list is to be cached or not.
cacheIcons	A flag that specifies whether the trophy icons are to be cached or not.
saveDataInBytes	Not used. Added only to provide interface parity with the PlayStation®3 platform.
hddSizeInBytes	Not used. Added only to provide interface parity with the PlayStation®3 platform.
userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.
serviceLabel	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_TROPHY_NOT_INITIALISED`	The operation failed because the trophy service has not been initialized yet.
`trophySetSetupSuccess`	This event is passed to an event callback to signify success.
`trophySetSetupCancelled`	This event is passed to an event callback when the trophy setup has been cancelled by the user.
`trophySetSetupAborted`	This event is passed to an event callback when the trophy setup has been aborted.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Description

Registers a trophy set. Because the registration process can take a while, the NP Toolkit library spawns a sub-thread to register the trophy set. The function provides the option to cache the trophy list or the trophy icons during the registration process.

This function exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. A `RegisterTrophyRequest` parameter should be used instead when calling `trophyRegisterSet()`. This is always a synchronous (blocking) function.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

trophyRegisterSet

Registers a trophy set.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         namespace Trophy {
            class Interface {
               static int trophyRegisterSet(
                  const RegisterTrophyRequest *trophyRequest,
                  bool async = true
               );
            }
         }
      }
   }
}
```

Arguments

trophyRequest	Describes the details about the trophy package that needs to be registered for the specified user.
async	Not used. Added only to provide interface parity with the PlayStation®4 platform. This function is always a synchronous (blocking) function.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was kicked off successfully.
`SCE_TOOLKIT_NP_TROPHY_NOT_INITIALISED`	The operation failed because the trophy service has not been initialized yet.
`trophySetSetupSuccess`	This event is passed to an event callback to signify success.
`trophySetSetupCancelled`	This event is passed to an event callback when the trophy setup has been canceled by the user.
`trophySetSetupAborted`	This event is passed to an event callback when the trophy setup has been aborted.

Description

Registers a trophy set. Because the registration process can take a while, the NP Toolkit library spawns a sub-thread to register the trophy set. The function provides the option to cache the trophy list or the trophy icons during the registration process.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

trophyRetrieveGame

Retrieves overview information about a game's trophy set.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyRetrieveGame(
                        sce::Toolkit::NP::Utilities::Future
                            < TrophyGameInfo > *gameInfo,
                        bool async = true,
                        int userId = -1
                    );
                }
            }
        }
    }
}
```

Arguments

gameInfo	A `Future` object, which receives the game information.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.
userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *gameInfo* pointer was invalid.
`SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED`	The operation failed because the trophy set has not been registered yet.
`SCE_TOOLKIT_NP_TROPHY_BUSY`	The operation failed because the trophy service is currently busy processing other requests.
`SCE_TOOLKIT_NP_INIT_START_THREAD`	The operation failed because the trophy service failed to spawn a thread for trophy information retrieval.
`trophyGameInfoRetrievalSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the game's trophy set information has been retrieved successfully.
`trophyGameInfoRetrievalFail`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy service has failed to retrieve the game's trophy set information.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Description

Retrieves overview information about a game's trophy set. This includes the title's name, a description, the trophy set icon, the number of trophies available etc.

This function exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. A `RetrieveTrophyGameRequest` parameter should be used instead when calling `trophyRetrieveGame()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

trophyRetrieveGame

Retrieves overview information about a game's trophy set.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyRetrieveGame(
                        const RetrieveTrophyGameRequest *request,
                        sce::Toolkit::NP::Utilities::Future
                            < TrophyGameInfo > *gameInfo,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

request	The user making the request needs to be a valid user.
gameInfo	A `Future` object, which receives the game information.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because the *gameInfo* pointer was invalid.
SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED	The operation failed because the trophy set has not been registered yet.
SCE_TOOLKIT_NP_TROPHY_BUSY	The operation failed because the trophy service is currently busy processing other requests.
SCE_TOOLKIT_NP_INIT_START_THREAD	The operation failed because the trophy service failed to spawn a thread for trophy information retrieval.
trophyGameInfoRetrievalSuccess	In the case of an asynchronous operation, this event will be passed to an event callback when the game's trophy set information has been retrieved successfully.
trophyGameInfoRetrievalFail	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy service has failed to retrieve the game's trophy set information.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Description

Retrieves overview information about a game's trophy set. This includes the title's name, a description, the trophy set icon, the number of trophies available etc.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

trophyRetrieveGroups

Retrieves trophy group information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyRetrieveGroups(
                        sce::Toolkit::NP::Utilities::Future
                            < TrophyGroupInfo > *groupInfo,
                        SceNpTrophyGroupId groupId,
                        bool async = true,
                        int userId = -1
                    );
                }
            }
        }
    }
}
```

Arguments

groupInfo	A `Future` object, which receives the trophy group information.
groupId	The ID of the group the information is required for.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.
userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *groupInfo* pointer was invalid.
`SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED`	The operation failed because the trophy set has not been registered yet.
`SCE_TOOLKIT_NP_TROPHY_BUSY`	The operation failed because the trophy service is currently busy processing other requests.
`trophyGameInfoRetrievalSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy group information has been retrieved successfully.
`trophyGameInfoRetrievalFail`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy service has failed to retrieve the trophy group information.

Description

Retrieves trophy group information. This includes information such as the group name, description, icon, number of trophies within it etc.

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

This function exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. A `RetrieveTrophyGroupRequest` parameter should be used instead when calling `trophyRetrieveGroups()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

trophyRetrieveGroups

Retrieves trophy group information.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyRetrieveGroups(
                        const RetrieveTrophyGroupRequest *request,
                        sce::Toolkit::NP::Utilities::Future
                            < TrophyGroupInfo > *groupInfo,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

request	Describes the details about the group that wants to be retrieved and the user making the request.
groupInfo	A `Future` object, which receives the trophy group information.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *groupInfo* pointer was invalid.
`SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED`	The operation failed because the trophy set has not been registered yet.
`SCE_TOOLKIT_NP_TROPHY_BUSY`	The operation failed because the trophy service is currently busy processing other requests.
`trophyGameInfoRetrievalSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy group information has been retrieved successfully.
`trophyGameInfoRetrievalFail`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy service has failed to retrieve the trophy group information.

Description

Retrieves trophy group information. This includes information such as the group name, description, icon, number of trophies within it etc.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

trophyRetrieveList

Retrieves a trophy list.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyRetrieveList(
                        sce::Toolkit::NP::Utilities::Future
                            < TrophyInfo > *trophyList,
                        bool async = true,
                        int userId = -1
                    );
                }
            }
        }
    }
}
```

Arguments

trophyList	A `Future` object, which receives the trophy information.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.
userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *trophyList* pointer was invalid.
`SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED`	The operation failed because the trophy set has not been registered yet.
`SCE_TOOLKIT_NP_TROPHY_BUSY`	The operation failed because the trophy service is currently busy processing other requests.
`SCE_TOOLKIT_NP_INIT_START_THREAD`	The operation failed because the trophy service failed to spawn a thread for trophy information retrieval.
`trophyListRetrievalSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy list has been retrieved successfully.
`trophyListRetrievalFail`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy service has failed to retrieve the trophy list.

Description

Retrieves a trophy list. This is filled with trophy information on all the trophies in the trophy set.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

This function exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. A `RetrieveTrophyListRequest` parameter should be used instead when calling `trophyRetrieveList()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

trophyRetrieveList

Retrieves a trophy list.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyRetrieveList(
                        const RetrieveTrophyListRequest *request,
                        sce::Toolkit::NP::Utilities::Future
                            < TrophyInfo > *trophyList,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

request	Describes the details about the trophy list that needs to be retrieved to the specified user.
trophyList	A `Future` object, which receives the trophy information.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *trophyList* pointer was invalid.
`SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED`	The operation failed because the trophy set has not been registered yet.
`SCE_TOOLKIT_NP_TROPHY_BUSY`	The operation failed because the trophy service is currently busy processing other requests.
`SCE_TOOLKIT_NP_INIT_START_THREAD`	The operation failed because the trophy service failed to spawn a thread for trophy information retrieval.
`trophyListRetrievalSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy list has been retrieved successfully.
`trophyListRetrievalFail`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy service has failed to retrieve the trophy list.

Description

Retrieves a trophy list. This is filled with trophy information on all the trophies in the trophy set.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

trophyRetrieveProgress

Retrieves the user's trophy progress.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyRetrieveProgress(
                        sce::Toolkit::NP::Utilities::Future
                            < SceNpTrophyGameData > *trophyUserProgress,
                        bool async = true,
                        int userId = -1
                    );
                }
            }
        }
    }
}
```

Arguments

trophyUserProgress	A `Future` object, which receives the user's trophy progress.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.
userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *trophyUserProgress* pointer was invalid.
`SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED`	The operation failed because the trophy set has not been registered yet.
`SCE_TOOLKIT_NP_TROPHY_BUSY`	The operation failed because the trophy service is currently busy processing other requests.
`trophyProgressSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the user's trophy progress has been retrieved successfully.
`trophyProgressFail`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy service has failed to retrieve the user's trophy progress.

Description

Retrieves the user's trophy progress.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

This function exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. A `RetrieveUserTrophyProgressRequest` parameter should be used instead when calling `trophyRetrieveProgress()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

trophyRetrieveProgress

Retrieves the user's trophy progress.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyRetrieveProgress(
                        const RetrieveUserTrophyProgressRequest *request,
                        sce::Toolkit::NP::Utilities::Future
                            < SceNpTrophyGameData > *trophyUserProgress,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

request	Describes the details about the user whose trophy information wants to be retrieved.
trophyUserProgress	A `Future` object, which receives the user's trophy progress.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *trophyUserProgress* pointer was invalid.
`SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED`	The operation failed because the trophy set has not been registered yet.
`SCE_TOOLKIT_NP_TROPHY_BUSY`	The operation failed because the trophy service is currently busy processing other requests.
`trophyProgressSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the user's trophy progress has been retrieved successfully.
`trophyProgressFail`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy service has failed to retrieve the user's trophy progress.

Description

Retrieves the user's trophy progress.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

trophyTerm

Terminates the trophy service.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyTerm();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.

Description

Terminates the trophy service. Internally, this function deallocates the memory used to cache trophy icons, trophy group information game information. It will also unload the trophy modules and calls the following termination functions: `sceNpTrophyDestroyHandle()`, `sceNpTrophyDestroyContext()` and `sceNpTrophyTerm()`. For the related error messages, please refer to these function's descriptions in the *NP Trophy Library Reference* document.

There is no need to call this function. It only exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. Instead of this function, `terminateService()` should be called with `trophy` specified for the *`serviceType`* argument.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

trophyUnlock

Unlocks a particular trophy.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyUnlock(
                        int trophyId,
                        bool async = true,
                        int userId = -1
                    );
                }
            }
        }
    }
}
```

Arguments

trophyId	The ID of the trophy that needs to be unlocked.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.
userId	Not used. Added only to provide interface parity with the PlayStation®4 platform.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED`	The operation failed because the trophy set has not been registered yet.
`SCE_TOOLKIT_NP_TROPHY_BUSY`	The operation failed because the trophy service is currently busy processing other requests.
`trophyUnlockSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy has been unlocked successfully.
`trophyPlatinumUnlocked`	In the case of an asynchronous operation, this event will be passed to an event callback when a platinum trophy has been unlocked.
`trophyUnlockFail`	In the case of an asynchronous operation, this event will be passed to an event callback when a trophy has failed to unlock.

Description

Unlocks a particular trophy. This function should be executed straight after a user has carried out an action that met the unlock criteria. A platinum trophy will also be automatically unlocked if the last trophy in the trophy set was unlocked by this operation.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

This function exists to maintain backwards compatibility with previous versions of the PlayStation®Vita platform. A `UnlockTrophyRequest` parameter should be used instead when calling `trophyUnlock()`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

trophyUnlock

Unlocks a particular trophy.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Trophy {
                class Interface {
                    static int trophyUnlock(
                        const UnlockTrophyRequest *trophyRequest,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

trophyRequest	Describes the details about the trophy that needs to be unlocked.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED`	The operation failed because the trophy set has not been registered yet.
`SCE_TOOLKIT_NP_TROPHY_BUSY`	The operation failed because the trophy service is currently busy processing other requests.
`trophyUnlockSuccess`	In the case of an asynchronous operation, this event will be passed to an event callback when the trophy has been unlocked successfully.
`trophyPlatinumUnlocked`	In the case of an asynchronous operation, this event will be passed to an event callback when a platinum trophy has been unlocked.
`trophyUnlockFail`	In the case of an asynchronous operation, this event will be passed to an event callback when a trophy has failed to unlock.

Description

Unlocks a particular trophy. This function should be executed straight after a user has carried out an action that met the unlock criteria. A platinum trophy will also be automatically unlocked if the last trophy in the trophy set was unlocked by this operation.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::TSS

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TSS

The namespace containing PlayStation™Network `TSS` (title small storage) functionality.

Definition

```
namespace TSS {}
```

Description

The namespace containing PlayStation™Network `TSS` (title small storage) functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::TSS::Interface	The external interface to the PlayStation™Network `TSS` (title small storage) functionality.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::TSS::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TSS::Interface

The external interface to the PlayStation™Network TSS (title small storage) functionality.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

The external interface to the PlayStation™Network TSS (title small storage) functionality. This class is used to obtain data from the a TSS server.

Methods Summary

Methods	Description
getData	Retrieves TSS data from a TSS server.
getDataFromSlot	Retrieves TSS data from a specified slot on a TSS server.
getDataStatus	Retrieves the status of data at specified slot on a TSS server.
getDataStatus	Retrieves the status of data at specified slot on a TSS server.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

getData

Retrieves TSS data from a TSS server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace TSS {
                class Interface {
                    static int getData(
                        sce::Toolkit::NP::Utilities::Future< TssData > *data,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

data	A pointer to the Future object that will hold the TSS data.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because an invalid pointer was passed to the *data* parameter.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Retrieves TSS data from a TSS server.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getDataFromSlot

Retrieves TSS data from a specified slot on a TSS server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace TSS {
                class Interface {
                    static int getDataFromSlot(
                        sce::Toolkit::NP::Utilities::Future< TssData > *data,
                        TssInputParams inputParams,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

data	A pointer to the `Future` object that will hold the TSS data.
inputParams	The input parameters specifying the receive buffer and TSS slot to get the data from.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because an invalid pointer was passed to the *data* parameter.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Retrieves TSS data from a specified slot on a TSS server.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getDataStatus

Retrieves the status of data at specified slot on a TSS server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace TSS {
                class Interface {
                    static int getDataStatus(
                        sce::Toolkit::NP::Utilities::Future
                            < SceNpTssDataStatus > *status,
                        const TssGetStatusInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

status	A pointer to the `Future` object that will hold the TSS data status.
params	The input parameters specifying the slot ID of the data we want to query and the Service Label.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_ERROR_INVALID_POINTER`	The operation failed because an invalid pointer was passed to the *status* parameter.
`SCE_TOOLKIT_NP_ERROR_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Retrieves the status of data at specified slot on a TSS server.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getDataStatus

Retrieves the status of data at specified slot on a TSS server.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace TSS {
                class Interface {
                    static int getDataStatus(
                        sce::Toolkit::NP::Utilities::Future
                            < SceNpTssDataStatus > *status,
                        uint32_t slotId,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

status	A pointer to the `Future` object that will hold the TSS data status.
slotId	The slot ID of the data to query.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed to the *status* parameter.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Retrieves the status of data at specified slot on a TSS server.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::TssData

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TssData

Represents TSS (title small storage) data.

Definition

```
#include <np_toolkit.h>
struct TssData {};
```

Description

Represents TSS (title small storage) data. This structure contains a pointer to the start of a buffer and the buffer size. The buffer will be filled with a file, held on a TSS server, which contains the data.

Fields

Public Instance Fields

`void *buffer`	The buffer to store the TSS data in.
`size_t size`	`SCE_NET_NP_TSS_MAX_SIZE` for slot 0 or `SCE_NP_TSS_EXTRA_SLOT_MAX_SIZE` for slots 1-15. The size of the data that was obtained from TSS. The maximum size of the buffer is defined by
`SceNpTssDataStatus status`	The status of data on the TSS server.

Methods Summary

Methods	Description
TssData	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

TssData

The default constructor.

Definition

```
#include <np_toolkit.h>
inline TssData();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
TssGetStatusInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TssGetStatusInputParams

Represents the input parameters used when getting the status of a `TSS` (title small storage) file from a specified slot.

Definition

```
#include <np_toolkit.h>
struct TssGetStatusInputParams {};
```

Description

Represents the input parameters used when getting the status of a `TSS` (title small storage) file from a specified slot.

Fields

Public Instance Fields

`uint32_t` *`serviceLabel`*	The service label (only used on the PlayStation®4).
`uint32_t` *`slotId`*	The Slot ID on the `TSS` server that the status of the `TSS` file needs to be retrieved from.

Methods Summary

Methods	Description
`TssGetStatusInputParams`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

TssGetStatusInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline TssGetStatusInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::TssInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TssInputParams

Represents the input parameters used when getting a TSS (title small storage) file from a specified slot.

Definition

```
#include <np_toolkit.h>
struct TssInputParams {};
```

Description

Represents the input parameters used when getting a TSS (title small storage) file from a specified slot.

Fields

Public Instance Fields

void *buffer	The user provided buffer to store the TSS data in.
size_t *size*	The size of the buffer that is being passed in.
uint32_t *slotId*	The Slot ID on the TSS server that the data needs to be retrieved from.

Methods Summary

Methods	Description
TssInputParams	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

TssInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline TssInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::TUS

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TUS

The namespace containing PlayStation™Network TUS (title user storage) functionality.

Definition

```
namespace TUS {}
```

Description

The namespace containing PlayStation™Network TUS (title user storage) functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::TUS::Interface	The external interface to the PlayStation™Network TUS (title user storage) functionality.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::TUS::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TUS::Interface

The external interface to the PlayStation™Network TUS (title user storage) functionality.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

The external interface to the PlayStation™Network TUS (title user storage) functionality. This class is used to set and obtain variables and data from a TUS server.

Methods Summary

Methods	Description
getData	Gets a specified user's TUS binary data.
getVariables	Gets a specified user's TUS variables.
setData	Sets a specified user's TUS binary data.
setVariables	Sets a specified user's TUS variables.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

getData

Gets a specified user's TUS binary data.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace TUS {
                class Interface {
                    static int getData(
                        sce::Toolkit::NP::Utilities::Future
                            < TusDataOutput > *data,
                        TusGetDataInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

data	A `Future` object, which receives the specified user's TUS data.
params	The input parameters required to get a specified user's TUS data.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the user is not connected to the network.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because the *params* or *data* argument was invalid.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Gets a specified user's TUS binary data.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getVariables

Gets a specified user's TUS variables.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace TUS {
                class Interface {
                    static int getVariables(
                        sce::Toolkit::NP::Utilities::Future
                            < SceNpTusVariableList > *vars,
                        TusGetVarsInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

vars	A `Future` object, which receives the TUS variables.
params	The input parameters required to get a specified user's TUS variables.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the user is not connected to the network.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because the *vars* or *params* argument was invalid.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Gets a specified user's TUS variables.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

setData

Sets a specified user's TUS binary data.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace TUS {
                class Interface {
                    static int setData(
                        TusSetDataInputParams &params,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

params	The input parameters required to set a specified user's TUS data.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the user is not connected to the network.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because the *params* argument was invalid.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library is not initialized.

Description

Sets a specified user's TUS binary data.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

setVariables

Sets a specified user's TUS variables.

Definition

```
#include <np_toolkit.h>
namespace sce {
   namespace Toolkit {
      namespace NP {
         namespace TUS {
            class Interface {
               static int setVariables(
                  TusSetVarsInputParams &params,
                  bool async = true
               );
            }
         }
      }
   }
}
```

Arguments

params	The input parameters required to set the TUS variables of the specified user.
async	A flag that indicates whether the function is non-blocking or blocking. Defaults to true so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the user is not connected to the network.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because the *params* argument was invalid.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library is not initialized.

Description

Sets a specified user's TUS variables.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::TusData

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TusData

Represents TUS (title user storage) data.

Definition

```
#include <np_toolkit.h>
struct TusData {};
```

Description

Represents TUS (title user storage) data.

Fields

Public Instance Fields

`void *buffer`	The TUS data buffer.
`size_t bufferSize`	The TUS data buffer size.

Methods Summary

Methods	Description
TusData	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

TusData

The default constructor.

Definition

```
        #include <np_toolkit.h>
        inline TusData();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: TusGetDataInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TusGetDataInputParams

Represents input parameters used when getting TUS (title user storage) data.

Definition

```
#include <np_toolkit.h>
struct TusGetDataInputParams {};
```

Description

Represents input parameters used when getting TUS (title user storage) data.

Fields

Public Instance Fields

`bool` *`isVirtualUser`*	A flag that specifies whether this update is for a virtual user.
`SceNpId` *`npid`*	The NP ID of the user to retrieve the data for.
`char` *`padding[3]`*	Padding.
`uint32_t` *`serviceLabel`*	The PlayStation®4 service label.
`int32_t` *`slotId`*	The ID of the slot that the data belongs to.

Methods Summary

Methods	Description
`TusGetDataInputParams`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

TusGetDataInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline TusGetDataInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: TusGetVarsInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TusGetVarsInputParams

Represents the input parameters used when getting TUS (title user storage) variables.

Definition

```
#include <np_toolkit.h>
struct TusGetVarsInputParams {};
```

Description

Represents the input parameters used when getting TUS (title user storage) variables.

Fields

Public Instance Fields

`bool` *`isVirtualUser`*	A flag that specifies whether this update is for a virtual user.
`SceNpId` *`npid`*	The NP ID of the user that is being updated.
`size_t` *`numSlots`*	The number of slots used to retrieve data from.
`char` *`padding[3]`*	Padding.
`uint32_t` *`serviceLabel`*	The service label.
`int32_t` *`slotIds [SCE_TOOLKIT_NP_TUS_MAX_SLOTS]`*	The ID of the slot that the data belongs to.

Methods Summary

Methods	Description
TusGetVarsInputParams	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

TusGetVarsInputParams

The default constructor.

Definition

```
        #include <np_toolkit.h>
        inline TusGetVarsInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: TusSetDataInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TusSetDataInputParams

Represents input parameters used when setting TUS (title user storage) data.

Definition

```
#include <np_toolkit.h>
struct TusSetDataInputParams {};
```

Description

Represents input parameters used when setting TUS (title user storage) data.

Fields

Public Instance Fields

TusData *data*	The TUS data to update.
bool *isVirtualUser*	A flag that specifies whether this update is for a virtual user.
uint64_t *lastChangedDate*	The date and time for conflict prevention. This is the equivalent to `SceRtcTick` on the PlayStation®4 and PlayStation®Vita, and `CellRtcTick` on the PlayStation®3. Processing is only executed when the time of the TUS data's last update, which is registered on the server, is identical with or older than the specified time. When no TUS data is registered on the server, no processing is performed. Specify 0 if no comparison is necessary.
SceNpId *npid*	The NP ID of the user that is being updated.
char *padding*[3]	Padding.
SceNpId *requiredLastChangeUser*	The NP ID of the update's author for conflict prevention. Processing is only executed when the author of the TUS data's last update, which is registered on the server, is identical with the specified NP ID. When no TUS data is registered on the server, processing is not performed. Use `memset()` to set this value to 0 if no comparison is necessary.
uint32_t *serviceLabel*	The service label (only used on the PlayStation®4).
int32_t *slotId*	The ID of the slot that the data belongs to.

Methods Summary

Methods	Description
TusSetDataInputParams	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

TusSetDataInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline TusSetDataInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: TusSetVarsInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TusSetVarsInputParams

Represents the input parameters used when setting TUS (title user storage) variables.

Definition

```
#include <np_toolkit.h>
struct TusSetVarsInputParams {};
```

Description

Represents the input parameters used when setting TUS (title user storage) variables.

Fields

Public Instance Fields

`bool` *`isVirtualUser`*	A flag that specifies whether this update is for a virtual user.
`SceNpId` *`npid`*	The NP ID of the user to set the variables for.
`char` *`padding`*`[3]`	Padding.
`uint32_t` *`serviceLabel`*	The service label (only used on the PlayStation®4).
`TusVariableList` *`vars`*	The list of TUS variables to update.

Methods Summary

Methods	Description
`TusSetVarsInputParams`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

TusSetVarsInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline TusSetVarsInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::TusVariable

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::TusVariable

Represents a TUS (title user storage) variable.

Definition

```
#include <np_toolkit.h>
struct TusVariable {};
```

Description

Represents a TUS (title user storage) variable.

Fields

Public Instance Fields

`int32_t` *`reserved`*	A reserved value.
`int32_t` *`slotId`*	The slot that the variable belongs to.
`int64_t` *`value`*	The TUS variable value.

Methods Summary

Methods	Description
`TusVariable`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

TusVariable

The default constructor.

Definition

```
#include <np_toolkit.h>
inline TusVariable(
   int32_t s = 0,
   int64_t v = 0
);
```

Arguments

s	The slot ID.
v	The value.

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::UnlockTrophyRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::UnlockTrophyRequest

Represents a request to unlock a trophy.

Definition

```
#include <np_toolkit.h>
struct UnlockTrophyRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to unlock a trophy.

It is required by the `Trophy::Interface::trophyUnlock()` function.

Fields

Public Instance Fields

`int32_t` *`trophyId`*	The ID of the trophy to be unlocked.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::
UpdateAttributeRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::UpdateAttributeRequest

A request structure used to specify the session attributes to update.

Definition

```
#include <np_toolkit.h>
struct UpdateAttributeRequest : public sce::Toolkit::NP::Request {};
```

Description

A request structure used to specify the session attributes to update.

Fields

Public Instance Fields

`SceToolkitNpSessionAttributeType` *`attributeType`*	The type of attribute.
`SceNpMatching2RoomMemberId` *`memberId`*	The member ID. Specify this if the update request is for a `SCE_TOOLKIT_NP_SESSION_MEMBER_ATTRIBUTE` attribute type.
`SceUInt8` *`padding`*	Padding.

Methods Summary

Methods	Description
`UpdateAttributeRequest`	The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

UpdateAttributeRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline UpdateAttributeRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP:: UpdateNpSessionRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::UpdateNpSessionRequest

Represents a request to update information on the Session server.

Definition

```
#include <np_toolkit.h>
struct UpdateNpSessionRequest : public sce::Toolkit::NP::Request {};
```

Description

Represents a request to update information on the Session server.

Fields

Public Instance Fields

LocalizedNpSessionName *localizedSessionNames	The localized session names.
LocalizedNpSessionStatus *localizedSessionStatuses	The pieces of localized session status information.
SceToolkitNpSessionLockFlag *lockFlag*	A flag that specifies whether the session is locked.
uint32_t *maxSlots*	The maximum number of slots available for the session.
SceNpSessionId *npSessionId*	The session ID related to the Session server.
uint32_t *numlocalizedSessionName*	The number of the localized session names.
uint32_t *numlocalizedSessionStatus*	The number of pieces of localized session status information.
char *sessionChangeableData	The binary data, which can be up to 1MiB in size.
uint32_t *sessionChangeableDataSize*	The size of the session data.
SceToolkitNpSessionTypeFlag *sessionFlag*	A flag that specifies whether the session is private or public.
char *sessionImage	The session image, which should be in JPEG format and can be up to 160KiB in size.
uint32_t *sessionImageSize*	The size of the image data.
char *sessionName[SCE_TOOLKIT_NP_SESSION_NAME_MAX_SIZE]*	The session name.
char *sessionStatus[SCE_TOOLKIT_NP_SESSION_STATUS_MAX_SIZE]*	The status string which will be registered with the Session server.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Methods Summary

Methods	Description
UpdateNpSessionRequest	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

UpdateNpSessionRequest

The default constructor.

Definition

```
#include <np_toolkit.h>
inline UpdateNpSessionRequest();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::UserProfile

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::UserProfile

The namespace containing PlayStation™Network user profile functionality.

Definition

```
namespace UserProfile {}
```

Description

The namespace containing PlayStation™Network user profile functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::UserProfile::Interface	The external interface to PlayStation™Network user profile functionality.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::UserProfile::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::UserProfile::Interface

The external interface to PlayStation™Network user profile functionality.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

This class contains the set of static functions for accessing information pertaining to the user's PlayStation™Network account.

Methods Summary

Methods	Description
`getAvatarUrl`	Gets the PlayStation™Network user's Avatar URL.
`getCachedUserInfo`	Gets a PlayStation™Network user's cached info.
`getCountryInfo`	Gets a PlayStation™Network user's country details.
`getNpId`	Gets the PlayStation™Network user's NP ID.
`getOnlineId`	Gets the PlayStation™Network user's Online ID.
`getParentalControlInfo`	Gets a PlayStation™Network user's parental control details.
`getPlatform`	Gets the current platform the application is running on.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

getAvatarUrl

Gets the PlayStation™Network user's Avatar URL.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace UserProfile {
                class Interface {
                    static int getAvatarUrl(
                        sce::Toolkit::NP::Utilities::Future
                            < SceNpAvatarUrl > *avatarUrl,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

avatarUrl	A string containing the PlayStation™Network user's Avatar URL. It must be of size SCE_NET_NP_AVATAR_URL_MAX_LENGTH.
async	A flag that indicates whether the function will be called asynchronously. It defaults to true (asynchronous), so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the network was unavailable.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because an invalid pointer was passed as an argument.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library was not initialized.

Description

Gets the PlayStation™Network user's Avatar URL, which is the URL for the user's image. The avatar is used as an icon to represent the user's account on the system software.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getCachedUserInfo

Gets a PlayStation™Network user's cached info.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace UserProfile {
                class Interface {
                    static int getCachedUserInfo(
                        sce::Toolkit::NP::Utilities::Future
                            < SceNpManagerCacheParam > *userInfo,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

userInfo	Output. Receives the PlayStation™Network user's cached info (`onlineId`, `npId`, `onlineName` and `avatarUrl`).
async	A flag that indicates whether the function will be called asynchronously. It defaults to true (asynchronous), so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because an invalid pointer was passed as an argument.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library was not initialized.

Description

Gets a PlayStation™Network user's cached info. This is information about the NP user that is saved to the internal hard disk drive every time the information is updated. The information is also saved when the user signs up or signs into the PlayStation™Network.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getCountryInfo

Gets a PlayStation™Network user's country details.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace UserProfile {
                class Interface {
                    static int getCountryInfo(
                        sce::Toolkit::NP::Utilities::Future< CountryInfo > *info,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

info	Output. Receives the PlayStation™Network user's country info, which includes the country code and language. Language is a `CELL_SYSUTIL_LANG_XXX` value defined in the system utilities.
async	A flag that indicates whether the function will be called asynchronously. It defaults to true (asynchronous), so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the network was unavailable.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed as an argument.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library was not initialized.

Description

Gets a PlayStation™Network user's country details.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getNpId

Gets the PlayStation™Network user's NP ID.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace UserProfile {
                class Interface {
                    static int getNpId(
                        sce::Toolkit::NP::Utilities::Future< SceNpId > *npid,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

npid	A pointer to the `SceNpId`.
async	A flag that indicates whether the function will be called asynchronously. It defaults to true (asynchronous), so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the network was unavailable.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed as an argument.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library was not initialized.

Description

Gets the PlayStation™Network user's NP ID. The NP ID appends option fields and version information necessary for server access to the Online ID. It is used by the system utilities of PlayStation™Network for identifying the user.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getOnlineId

Gets the PlayStation™Network user's Online ID.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace UserProfile {
                class Interface {
                    static int getOnlineId(
                        sce::Toolkit::NP::Utilities::Future
                            < SceNpOnlineId > *onlineId,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

onlineId	An array that must be of size `SCE_NET_NP_ONLINEID_MAX_LENGTH`.
async	A flag that indicates whether the function will be called asynchronously. It defaults to true (asynchronous), so the function is non-blocking by default.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful.
`SCE_TOOLKIT_NP_OFFLINE`	The operation failed because the network was unavailable.
`SCE_TOOLKIT_NP_INVALID_POINTER`	The operation failed because an invalid pointer was passed as an argument.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	The operation failed because the NP Toolkit library was not initialized.

Description

Gets the PlayStation™Network user's Online ID. The Online ID is selected by the user when signing up to PlayStation™Network. It is composed of 3 to 16 characters and can be made up of alphanumeric characters, hyphens and underscores. An Online ID is guaranteed to be unique.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getParentalControlInfo

Gets a PlayStation™Network user's parental control details.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace UserProfile {
                class Interface {
                    static int getParentalControlInfo(
                        sce::Toolkit::NP::Utilities::Future
                            < ParentalControlInfo > *info,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

info	Output. Receives the PlayStation™Network user's parental control info.
async	A flag that indicates whether the function will be called asynchronously. It defaults to true (asynchronous), so the function is non-blocking by default.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the network was unavailable.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because an invalid pointer was passed as an argument.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library was not initialized.

Description

Gets a PlayStation™Network user's parental control details. This includes content and chat restrictions as well as the user's age.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getPlatform

Gets the current platform the application is running on.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace UserProfile {
                class Interface {
                    static int getPlatform(
                        sce::Toolkit::NP::Utilities::Future
                            < SceNpPlatformType > *platform,
                        bool async = true
                    );
                }
            }
        }
    }
}
```

Arguments

platform	Output. Receives the current platform.
async	A flag that indicates whether the function will be called synchronously or asynchronously. Defaults to true.

Return Values

Value	Description
SCE_TOOLKIT_NP_SUCCESS	The operation was successful.
SCE_TOOLKIT_NP_OFFLINE	The operation failed because the network was unavailable.
SCE_TOOLKIT_NP_INVALID_POINTER	The operation failed because an invalid pointer was passed as an argument.
SCE_TOOLKIT_NP_NOT_INITIALISED	The operation failed because the NP Toolkit library was not initialized.

Description

Gets the current platform the application is running on.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::UserRankRequest

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::UserRankRequest

Represents a request to retrieve ranking information for a user.

Definition

```
#include <np_toolkit.h>
struct UserRankRequest : public sce::Toolkit::NP::RankingRequest {};
```

Description

Represents a request to retrieve ranking information for a user.

Fields

Public Instance Fields

`SceNpScoreBoardId` *`boardId`*	The ID of the board to retrieve the ranking information from.
`SceNpId` *`npId`*	The NP ID of the user.
`int32_t` *`userIndex`*	The user’s index in the board. This is used on the PlayStation®3 platform only.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Utilities

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Utilities

The namespace for utilities used by the NP Toolkit library.

Definition

```
namespace Utilities {}
```

Description

The namespace for utilities used by the NP Toolkit library.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::Utilities::Future	A template implementation of the future class.
sce::Toolkit::NP::Utilities::FutureImpl	Represents a piece of data, for which an asynchronous reference is provided, that will be finalized at some point in the future.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Utilities::Future

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Utilities::Future

A template implementation of the future class.

Definition

```
#include <np_toolkit.h>
template <class T>
class Future : public sce::Toolkit::NP::Utilities::FutureImpl {};
```

Description

This class provides the implementation of future objects for the NP Toolkit library. A future object is effectively "locked" until the result is ready or an error occurs.

The template conforms to the same functionality as `FutureImpl`. However, it contains the pointer to data of template type T.

Methods Summary

Methods	Description
`Future`	The default constructor.
`~Future`	The default virtual destructor.
`get`	Gets a pointer to the internal data that has been set.
`getAdditionalInfo`	Gets a pointer to the additional information associated with the current web request.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

Future

The default constructor.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class Future {
                    inline Future();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

None

Description

This will call the default constructor of class T.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~Future

The default virtual destructor.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class Future {
                    virtual inline ~Future();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

None

Description

The destructor of the `Future` will also destroy the enclosed T object.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

get

Gets a pointer to the internal data that has been set.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class Future {
                    inline T *get();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

A pointer to the encapsulated data. A `NULL` pointer is returned if the data has not been set.

Description

Gets a pointer to the internal data that has been set.

Notes

The internal data is effectively inaccessible until it has been set.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getAdditionalInfo

Gets a pointer to the additional information associated with the current web request.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class Future {
                    inline const AdditionalInfo *getAdditionalInfo() const;
                }
            }
        }
    }
}
```

Arguments

None

Return Values

A pointer to the encapsulated data. A `NULL` pointer is returned if the data has not been set.

Description

Gets a pointer to the additional information associated with the current web request.

Notes

Only access this once the result for the `Future` object has been set.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::Utilities::FutureImpl

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::Utilities::FutureImpl

Represents a piece of data, for which an asynchronous reference is provided, that will be finalized at some point in the future.

Definition

```
#include <np_toolkit.h>
class FutureImpl {};
```

Description

A FutureImpl object contains the synchronization required for one particular thread to wait on the result of another thread, which is in the process of completing an operation. When the operation is complete, the referenced piece of data will be written to asynchronously by that thread. The thread requesting the data can either wait on it being provided, or at periodic points in code query the FutureImpl object regarding completion of the other thread. This provides flexibility as to when the data synchronization is actually performed.

Methods Summary

Methods	Description
getError	Gets the error code for an error after checking if one occurred.
hasError	Checks whether an error has occurred.
hasResult	Checks whether the data has been provided.
isBusy	Checks if a FutureImpl object is busy.
reset	Resets the FutureImpl object to the state it had at initialization.
setBusy	Sets the FutureImpl object as busy.
waitFor	Waits for the data to be set by an asynchronous thread.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

getError

Gets the error code for an error after checking if one occurred.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class FutureImpl {
                    int getError() const;
                }
            }
        }
    }
}
```

Arguments

None

Return Values

The error code. If no error has occurred, a value of 0 / `SCE_TOOLKIT_NP_SUCCESS` will be returned.

Description

Gets the error code for an error after checking if one occurred.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

hasError

Checks whether an error has occurred.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class FutureImpl {
                    bool hasError() const;
                }
            }
        }
    }
}
```

Arguments

None

Return Values

A value of true indicates that an error has occurred; otherwise false is returned.

Description

Checks whether an error has occurred.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

hasResult

Checks whether the data has been provided.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class FutureImpl {
                    bool hasResult() const;
                }
            }
        }
    }
}
```

Arguments

None

Return Values

A value of true if the data has been provided; otherwise false is returned.

Description

Checks whether the data has been provided.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

isBusy

Checks if a FutureImpl object is busy.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class FutureImpl {
                    bool isBusy();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

A value true if the object is busy; otherwise a value of false is returned.

Description

Checks if a FutureImpl object is busy.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

reset

Resets the FutureImpl object to the state it had at initialization.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class FutureImpl {
                    bool reset();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

A value true if the object was correctly reset; otherwise a value of false is returned. A value of false indicates the object was still in use by a thread.

Description

Resets the FutureImpl object to the state it had at initialization.

Notes

This function should not be used while a thread is waiting on the FutureImpl object.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setBusy

Sets the FutureImpl object as busy.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class FutureImpl {
                    void setBusy();
                }
            }
        }
    }
}
```

Arguments

None

Return Values

None

Description

Sets the FutureImpl object as busy.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

waitFor

Waits for the data to be set by an asynchronous thread.

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace Utilities {
                class FutureImpl {
                    void waitFor(
                        unsigned int timeout
                    ) const;
                }
            }
        }
    }
}
```

Arguments

timeout	The amount of time in microseconds to wait for the data to be set. A value of 0 will result in the operation returning immediately.

Return Values

None

Description

Waits for the data to be set by an asynchronous thread. This function will wait until the timeout is reached unless the data is set.

Notes

Because future objects are not tightly bound to the thread objects that are responsible for them, indefinite waits are not permitted as this could cause deadlocks. If the future objects were more tightly coupled and the threads could guarantee that all future objects were properly terminated, then this would be less of an issue.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::VoucherInputParams

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::VoucherInputParams

Contains details of how a voucher will be redeemed.

Definition

```
#include <np_toolkit.h>
struct VoucherInputParams : public sce::Toolkit::NP::Request {};
```

Description

Contains details of how a voucher will be redeemed.

Fields

Public Instance Fields

`bool` *`inGame`*	Used on the PlayStation®3 platform only. Added here for parity.
`void *`*`memContainer`*	Used on the PlayStation®3 platform only. Added here for parity.
`SceChar8` *`padding`*`[3]`	Padding of 3 bytes. Ensures alignment to a 4-byte boundary.
`uint32_t` *`serviceLabel`*	The PlayStation®4 service label.
`int` *`userData`*	Used on the PlayStation®3 platform only. Added here for parity.

Methods Summary

Methods	Description
`VoucherInputParams`	The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

VoucherInputParams

The default constructor.

Definition

```
#include <np_toolkit.h>
inline VoucherInputParams();
```

Arguments

None

Return Values

None

Description

The default constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::WordFilter

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::WordFilter

The namespace containing word filter functionality.

Definition

```
namespace WordFilter {}
```

Description

The namespace containing word filter functionality.

Inner Classes, Structures, and Namespaces

Item	Description
sce::Toolkit::NP::WordFilter::Interface	The external interface to the word filter functionality.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Toolkit::NP::WordFilter::Interface

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Toolkit::NP::WordFilter::Interface

The external interface to the word filter functionality.

Definition

```
#include <np_toolkit.h>
class Interface {};
```

Description

The external interface to the word filter functionality.

Methods Summary

Methods	Description
filterWord	Filters or sanitizes a comment (or a singular word).

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Static Methods

filterWord

Filters or sanitizes a comment (or a singular word).

Definition

```
#include <np_toolkit.h>
namespace sce {
    namespace Toolkit {
        namespace NP {
            namespace WordFilter {
                class Interface {
                    static int filterWord(
                        sce::Toolkit::NP::Utilities::Future
                            < WordFilterSanitized > *sanitizedComment,
                        WordFilterParam *paramIn
                    );
                }
            }
        }
    }
}
```

Arguments

sanitizedComment	A `Future` object of the WordFilterSanitized type which will return the sanitized comment. Specifying `NULL` for this parameter indicates that the comment should be censored rather than sanitized.
paramIn	The comment to filter or sanitize.

Return Values

Value	Description
`SCE_TOOLKIT_NP_SUCCESS`	The operation was successful. In the case of "censoring", this result indicates that the comment passed the censor.
`SCE_NP_COMMUNITY_SERVER_ERROR_CENSORED`	The operation was successful but the comment has been censored.
`SCE_TOOLKIT_NP_WORD_FILTER_SLOT_FULL`	The operation failed because the maximum number of simultaneous word filter operations allowed are currently being processed.
`SCE_TOOLKIT_NP_WORD_FILTER_NOT_INITIALISED`	The operation failed because the word filter has not been initialized.
`other`	An NP Library Error Code.

Description

Filters or sanitizes a comment (or a singular word).

There are two methods of filtering comments. The first method is to "censor" a comment, which results in an error code of `SCE_NP_COMMUNITY_SERVER_ERROR_CENSORED` being returned if inappropriate

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

comment is discovered. To use this method, a value of `NULL` should be passed to the *`sanitizedComment`* parameter.

The second method is to "sanitize" a comment, which results in a sanitized version of the original comment being returned via a `Future` object. In the sanitized version of the comment inappropriate words are replaced with a '`*`'. To use this method, a `Future` object of the `WordFilterSanitized` type should be passed to the *`sanitizedComment`* parameter.

Notes

It is possible to specify that this function should run asynchronously. To do this, use the *`isAsync`* member of the `WordFilterParam` object given as the *`paramIn`* argument.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Defines

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Define Summary

Define	Value	Description
SCE_TOOLKIT_NP_ALREADY_INITIALISED	0x81000003	An error occurred because the NP Toolkit library has already been initialized.
SCE_TOOLKIT_NP_ATTACHMENT_DATA_URL_LENGTH	1128	SCE_TOOLKIT_NP_ATTACHMENT_DATA_URL_LENGTH
SCE_TOOLKIT_NP_ATTRIBUTE_MAX_BIN_VALUE	252	The maximum value of a binary attribute.
SCE_TOOLKIT_NP_AUTH_NO_VALID_CACHED_TICKET	0x8D000001	An error occurred because there was no valid cached ticket.
SCE_TOOLKIT_NP_AVAILABLE_PLATFORM_PS3	1<<0	This flag indicates that the feature is available on PlayStation®3.
SCE_TOOLKIT_NP_AVAILABLE_PLATFORM_PS4	1<<2	This flag indicates that the feature is available on PlayStation®4.
SCE_TOOLKIT_NP_AVAILABLE_PLATFORM_PSVITA	1<<1	This flag indicates that the feature is available on PlayStation®Vita.
SCE_TOOLKIT_NP_CHALLENGES_NOT_A_CHALLENGE	0x80552cf3	An error occurred because the requested item is not a challenge.
SCE_TOOLKIT_NP_CHALLENGES_RETRIEVE_TYPE_ALL	(0)	A flag that specifies to retrieve all challenge and response types.
SCE_TOOLKIT_NP_CHALLENGES_RETRIEVE_TYPE_CHALLENGE	(1)	A flag that specifies to retrieve only challenges.
SCE_TOOLKIT_NP_CHALLENGES_RETRIEVE_TYPE_RESPONSE	(2)	A flag that specifies to retrieve only responses to challenges.
SCE_TOOLKIT_NP_CHALLENGES_SEND_ATTACHMENT_MAX_SIZE	(1024*1023)	Maximum size of the challenge binary data.
SCE_TOOLKIT_NP_CHAR_LENGTH_128	128	Specifies a length of 128 characters.
SCE_TOOLKIT_NP_CHAR_LENGTH_256	256	Specifies a length of 256 characters.
SCE_TOOLKIT_NP_CHAR_LENGTH_512	512	Specifies a length of 512 characters.
SCE_TOOLKIT_NP_CHAR_LENGTH_64	64	Specifies a length of 64 characters.
SCE_TOOLKIT_NP_COMMERCE_CATEGORY_DESCRIPTION_LEN	1024	The size of the category description.
SCE_TOOLKIT_NP_COMMERCE_CATEGORY_ID_LEN	56	The size of the category ID.
SCE_TOOLKIT_NP_COMMERCE_CATEGORY_NAME_LEN	256	The size of the category name.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_COMMERCE_CURRENCY_CODE_LEN	3	The size of currency code.
SCE_TOOLKIT_NP_COMMERCE_CURRENCY_SYMBOL_LEN	3	The size of currency symbol.
SCE_TOOLKIT_NP_COMMERCE_DECIMAL_LETTER_LEN	4	The size of the character indicating the decimal point in the price.
SCE_TOOLKIT_NP_COMMERCE_ENTITLEMENT_ID_LEN	32	The size of entitlement ID.
SCE_TOOLKIT_NP_COMMERCE_ENTITLEMENT_TYPE_DRM	(1<<2)	Specifies a DRM entitlement.
SCE_TOOLKIT_NP_COMMERCE_ENTITLEMENT_TYPE_NONE	(0)	No entitlement type specified.
SCE_TOOLKIT_NP_COMMERCE_ENTITLEMENT_TYPE_SERVICE	(1<<0)	Specifies a service entitlement.
SCE_TOOLKIT_NP_COMMERCE_ENTITLEMENT_TYPE_SERVICE_ CONSUMABLE	(1<<1)	Specifies a consumable service entitlement.
SCE_TOOLKIT_NP_COMMERCE_ENTITLEMENT_TYPE_UNIFIED	(1<<3)	Specifies a unified DRM entitlement.
SCE_TOOLKIT_NP_COMMERCE_NOT_PURCHASED	0	Specifies that a user has not purchased a product.
SCE_TOOLKIT_NP_COMMERCE_PRODUCT_ID_LEN	48	The size of the product ID.
SCE_TOOLKIT_NP_COMMERCE_PRODUCT_LEGAL_DESCRIPTION_ LEN	1	The size of the product legal description.
SCE_TOOLKIT_NP_COMMERCE_PRODUCT_LONG_DESCRIPTION_LEN	4000	The size of the product long description.
SCE_TOOLKIT_NP_COMMERCE_PRODUCT_NAME_LEN	256	The size of the product name.
SCE_TOOLKIT_NP_COMMERCE_PRODUCT_SHORT_DESCRIPTION_ LEN	1	The size of the product short description.
SCE_TOOLKIT_NP_COMMERCE_PURCHASED_CAN_PURCHASE_AGAIN	0x40000000	Specifies that a product has already been purchased and can be purchased again. These will be consumable or time limited service entitlements.
SCE_TOOLKIT_NP_COMMERCE_PURCHASED_CANNOT_PURCHASE_ AGAIN	0x80000000	Specifies that a product has already been purchased and cannot be purchased again.
SCE_TOOLKIT_NP_COMMERCE_RATING_SYSTEM_ID_LEN	16	The size of the rating system ID.
SCE_TOOLKIT_NP_COMMERCE_SKU_ID_LEN	56	The size of the SKU ID.
SCE_TOOLKIT_NP_COMMERCE_SP_NAME_LEN	256	The size of the licensee (publisher) name.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
`SCE_TOOLKIT_NP_COMMERCE_THOUSAND_SEPARATOR_LEN`	`4`	The size of the character separating every 3 digits of the price.
`SCE_TOOLKIT_NP_COMMERCE_URL_LEN`	`256`	The size of the URL.
`SCE_TOOLKIT_NP_COUNTRY_CODE_LEN`	`2`	The maximum size of a country code string.
`SCE_TOOLKIT_NP_COUNTRY_CODE_LEN`	`2`	The maximum size of a country code string.
`SCE_TOOLKIT_NP_CREATE_ALLOW_BLOCK_LIST_SESSION`	`(1<<5)`	This flag specifies that the current session can be joined by a blocked user. This is not allowed by default.
`SCE_TOOLKIT_NP_CREATE_HOST_MIGRATION_SESSION`	`(1<<6)`	This flag specifies that the session should be allowed to migrate when the owner quits. The ability to grant room ownership is not supported by default.
`SCE_TOOLKIT_NP_CREATE_NAT_RESTRICTED_SESSION`	`(1<<7)`	This flag specifies that users who cannot establish P2P connections are not allowed to join the session.
`SCE_TOOLKIT_NP_CREATE_PASSWORD_SESSION`	`(1<<4)`	This flag specifies that the session is password protected.
`SCE_TOOLKIT_NP_CREATE_SESSION_TYPE_PRIVATE`	`(1<<3)`	This flag specifies that all the slots in the session are reserved for private players or friends. If this flag is set, then the session will not be visible to other users during searches.
`SCE_TOOLKIT_NP_CREATE_SESSION_TYPE_PUBLIC`	`(1<<2)`	This flag specifies that all the slots in the session are available to the public.
`SCE_TOOLKIT_NP_CREATE_SIGNALING_MESH_SESSION`	`(1<<2)`	This flag specifies that the session supports signaling.
`SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_DATA_DESCRIPTION_LEN`	`512`	The maximum length of a custom data description.
`SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_DATA_NAME_LEN`	`128`	The maximum length of a data name.
`SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_NP_LANG_CODE_LEN`	`5`	The maximum length of a custom data NP language code.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_SIZE	(1024*1024)	Defines the maximum size of some custom data.
SCE_TOOLKIT_NP_CUSTOM_DATA_MAX_USER_MESSAGE_LEN	512	The maximum length of a user message.
SCE_TOOLKIT_NP_DATE_LENGTH	20	Length of the date string.
SCE_TOOLKIT_NP_DIALOG_TYPE_USER_EDITABLE	(2)	Defines that the users addressed in the "To" box of the dialog can be edited by the user.
SCE_TOOLKIT_NP_DIALOG_TYPE_USER_NON_EDITABLE	(1)	Defines that the users addressed in the "To" box of the dialog cannot be edited by the user. They have been pre-defined by the application.
SCE_TOOLKIT_NP_FAILED_ALLOCATE	0x81000002	An error occurred because the NP Toolkit library failed to allocate memory for an object.
SCE_TOOLKIT_NP_FRIENDS_LIST_ADDITIONAL_INFO	0x10	A flag to specify additional information is required. This can be ORed with the SCE_TOOLKIT_NP_FRIENDS_LIST_IN_CONTEXT and SCE_TOOLKIT_NP_FRIENDS_LIST_ONLINE flags.
SCE_TOOLKIT_NP_FRIENDS_LIST_ALL	0x02	A flag that specifies to retrieve a complete list of friends.
SCE_TOOLKIT_NP_FRIENDS_LIST_ALL	0x02	A flag that specifies to retrieve a complete list of friends.
SCE_TOOLKIT_NP_FRIENDS_LIST_CACHED	0x08	A flag that specifies to retrieve a cached list of friends.
SCE_TOOLKIT_NP_FRIENDS_LIST_IN_CONTEXT	0x04	A flag that specifies to retrieve a list of friends who are currently playing on the same game.
SCE_TOOLKIT_NP_FRIENDS_LIST_IN_CONTEXT	0x04	A flag that specifies to retrieve a list of friends who are currently playing on the same game.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
`SCE_TOOLKIT_NP_FRIENDS_LIST_ONLINE`	`0x01`	A flag that specifies to retrieve a list of friends which are currently online.
`SCE_TOOLKIT_NP_FRIENDS_LIST_ONLINE`	`0x01`	A flag that specifies to retrieve a list of friends which are currently online.
`SCE_TOOLKIT_NP_FUTURE_IS_IN_USE`	`0x81000006`	An error occurred because the `Future` object passed as an argument is already in use.
`SCE_TOOLKIT_NP_GAME_CUSTOM_DATA_TYPE_ATTACHMENT_DATA`	`(1)`	Defines that the type of game custom data is attachment.
`SCE_TOOLKIT_NP_GAME_CUSTOM_DATA_TYPE_ATTACHMENT_URL`	`(2)`	Defines that the type of game custom data is attachment URL.
`SCE_TOOLKIT_NP_GENERIC_ERRORS`	`0x81000010`	A generic NP Toolkit library error.
`SCE_TOOLKIT_NP_GIFT_DATA_TOO_BIG`	`0x88000007`	An error occurred because an attempt was made to register a gift whose data is over `SCE_TOOLKIT_NP_MAX_GIFT_BODY_SIZE` in size.
`SCE_TOOLKIT_NP_GIFT_IMAGE_TOO_BIG`	`0x88000008`	An error occurred because an attempt was made to register a gift whose image is over `SCE_NEAR_GIFT_IMAGE_MAX_SIZE` in size.
`SCE_TOOLKIT_NP_GIFT_INVALID_INFO`	`0x88000012`	An error occurred because the parameter passed in for gift creation is missing or invalid.
`SCE_TOOLKIT_NP_GIFT_NOT_COMPATIBLE`	`0x88000011`	An error occurred because the specified gift is incompatible with NP Toolkit. A gift needs to be created with NP Toolkit to be compatible with this service.
`SCE_TOOLKIT_NP_GIFT_NOT_MATCH`	`0x88000009`	An error occurred because the gift IDs specified in a call to `compareGiftId()` do not match.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
`SCE_TOOLKIT_NP_GIFT_NOT_RECEIVED`	`0x88000010`	An error occurred because the gift has not been “received” yet.
`SCE_TOOLKIT_NP_IN_GAME_PRESENCE_DATA_SIZE_MAX`	`128`	The maximum size of the presence binary data.
`SCE_TOOLKIT_NP_IN_GAME_PRESENCE_DATA_SIZE_MAX`	`170`	The maximum size of the presence binary data.
`SCE_TOOLKIT_NP_IN_GAME_PRESENCE_STATUS_SIZE_MAX`	`64`	The maximum size of the presence status string.
`SCE_TOOLKIT_NP_IN_GAME_PRESENCE_STATUS_SIZE_MAX`	`64`	The maximum size of the presence status string.
`SCE_TOOLKIT_NP_INIT_ERRORS`	`0x81000000`	An error was caused during the execution of initialization code.
`SCE_TOOLKIT_NP_INIT_INVALID_MEM_MANAGER`	`0x81000001`	An error occurred because the memory manager passed as an argument that was invalid.
`SCE_TOOLKIT_NP_INIT_START_THREAD`	`0x81000004`	An error occurred because the thread starting function failed. This was most likely due to locking errors.
`SCE_TOOLKIT_NP_INVALID_ARGUMENT`	`0x81000013`	An error occurred because an argument to a method was incorrect.
`SCE_TOOLKIT_NP_INVALID_MODULE`	`0x81000014`	An error occurred because the module specified was an invalid ID.
`SCE_TOOLKIT_NP_INVALID_NUM_SESSION_ATTRIBUTES`	`0x86000012`	An error occurred because there was an invalid number of session attributes. The maximum is 64.
`SCE_TOOLKIT_NP_INVALID_POINTER`	`0x81000011`	An error occurred because an invalid pointer was passed as an argument.
`SCE_TOOLKIT_NP_INVALID_WEBAPI_RESPONSE`	`0x80552c04`	An error occurred because an invalid response was received from the SEN server.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_KICK_MEMBER_FLAG_ALLOW_REJOIN	0	This flag specifies that a room member is allowed to join after being kicked out.
SCE_TOOLKIT_NP_KICK_MEMBER_FLAG_NO_REJOIN	1	This flag specifies that a room member is not allowed to join after being kicked out.
SCE_TOOLKIT_NP_LANGUAGE_CODE_MAX_LEN	(5)	The maximum length of a language code.
SCE_TOOLKIT_NP_MATCHING_ATTRIBUTES_ALREADY_ REGISTERED	0x86000019	An error occurred because the session attributes where already registered.
SCE_TOOLKIT_NP_MATCHING_CALLBACK_FAILURE_ERROR	0x86000004	An error occurred because there an internal failure during processing of a request.
SCE_TOOLKIT_NP_MATCHING_FAIL_TO_REGISTER_ATTRIBUTES	0x8600000C	An error occurred because the operation failed to register session attributes.
SCE_TOOLKIT_NP_MATCHING_INVALID_JOIN_DESCRIPTOR	0x8600000A	An error occurred because the join descriptor parameters are invalid.
SCE_TOOLKIT_NP_MATCHING_INVALID_MODIFY_ATTRIBUTES	0x86000017	An error occurred because the operation failed to modify the current session attributes.
SCE_TOOLKIT_NP_MATCHING_INVALID_PARAMETERS	0x86000007	An error occurred because invalid parameters were passed in when registering session attributes.
SCE_TOOLKIT_NP_MATCHING_INVALID_ROOM_ID	0x8600000D	An error occurred because an invalid room ID was passed in.
SCE_TOOLKIT_NP_MATCHING_INVALID_ROOM_MESSAGE	0x8600001A	An error occurred because the wrong message flag was specified.
SCE_TOOLKIT_NP_MATCHING_INVALID_SEARCH_CRITERIA	0x86000008	An error occurred because invalid search criteria were passed in.
SCE_TOOLKIT_NP_MATCHING_INVALID_SESSION_DESC	0x86000001	An error occurred because a create session descriptor was invalid.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_MATCHING_INVALID_SLOTS_INFORMATION	0x8600001B	An error occurred because the wrong combination of slots information was provided for the session.
SCE_TOOLKIT_NP_MATCHING_JOIN_FAILED	0x8600000B	An error occurred because the operation failed to join a session.
SCE_TOOLKIT_NP_MATCHING_LEAVING_FAILED	0x8600000E	An error occurred because the operation failed to leave a session.
SCE_TOOLKIT_NP_MATCHING_NO_SESSION_ACTIVE	0x86000018	An error occurred because the operation failed to modify the current session as the session you trying to update is not active.
SCE_TOOLKIT_NP_MATCHING_NO_SESSION_TO_JOIN	0x8600000F	An error occurred because there was no session to join.
SCE_TOOLKIT_NP_MATCHING_SEARCH_FAILED	0x86000009	An error occurred while searching for a session.
SCE_TOOLKIT_NP_MATCHING_SERVICE_BUSY	0x86000005	An error occurred because the previous operation requested by the application was still being processed.
SCE_TOOLKIT_NP_MATCHING_SESSION_ALREADY_ACTIVE	0x86000006	An error occurred because a session can be created or joined when the user is already in a session.
SCE_TOOLKIT_NP_MATCHING_SESSION_CREATION_FAILED	0x86000003	An error occurred during creation of a session.
SCE_TOOLKIT_NP_MATCHING_SESSION_DOES_NOT_EXIST	0x86000013	An error occurred because the session does not exist.
SCE_TOOLKIT_NP_MATCHING_SESSION_KICKEDOUT	0x86000015	An error occurred because the user has been kicked out of the current session.
SCE_TOOLKIT_NP_MATCHING_SESSION_ROOM_DESTROYED	0x86000014	An error occurred because the session has been destroyed.
SCE_TOOLKIT_NP_MATCHING_SESSION_UPDATE_FAILED	0x86000016	An error occurred because the operation failed to update the current session.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_MAX_ATTRIBUTE_LENGTH	32	The maximum length of the session attribute.
SCE_TOOLKIT_NP_MAX_ATTRIBUTES_IN_A_SESSION	64	The maximum number of attributes a session can have.
SCE_TOOLKIT_NP_MAX_BOARD_LINE	24	The maximum lines in a ranking board.
SCE_TOOLKIT_NP_MAX_FRIEND_LINE	101	The maximum number of friends.
SCE_TOOLKIT_NP_MAX_GIFT_BODY_SIZE	102144	The maximum size of the gift data body without the header part.
SCE_TOOLKIT_NP_MAX_IMAGE_SIZE	(160*1024)	Defines the maximum size of an image.
SCE_TOOLKIT_NP_MAX_NUM_BOARD	4	The maximum number of boards.
SCE_TOOLKIT_NP_MAX_NUM_NEIGHBORS	100	The maximum number of nearby users that can be retrieved at one time.
SCE_TOOLKIT_NP_MAX_ONLINEID	(SCE_NP_ONLINEID_MAX_LENGTH +1)	The maximum length of an online ID. This is the same as SCE_NP_ONLINEID_MAX_LENGTH plus NULL.
SCE_TOOLKIT_NP_MAX_PERSONAL_DETAIL_NAME_SIZE	16	The maximum length, in the personal details, the first, middle or last name of the user can be.
SCE_TOOLKIT_NP_MAX_PLATFORM_STRING_LENGTH	8	The maximum size of the platform string.
SCE_TOOLKIT_NP_MAX_RANGE	30	The maximum range of ranking scores requested.
SCE_TOOLKIT_NP_MAX_READ_LINE	128	The maximum board lines read.
SCE_TOOLKIT_NP_MAX_URL_SIZE	256	The maximum length a URL of the user's profile picture can be.
SCE_TOOLKIT_NP_MAX_WRITE_LINE	24	The maximum board lines written to.
SCE_TOOLKIT_NP_MESSAGE_ARGUMENTS_INVALID	0x86000033	An error occurred because invalid arguments were passed.
SCE_TOOLKIT_NP_MESSAGE_ATTACHMENT_INVALID	0x86000030	An error occurred because the attachment is invalid.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_MESSAGE_DATA_TOO_LARGE	0x81000031	An error occurred because a data attachment was too large.
SCE_TOOLKIT_NP_MESSAGE_INVALID_ID	0x81000034	An error occurred because a message had an invalid ID.
SCE_TOOLKIT_NP_MESSAGE_NO_DATA	0x81000033	An error occurred because there was no waiting data attachment.
SCE_TOOLKIT_NP_MESSAGE_SERVICE_BUSY	0x86000032	An error occurred because messaging service is processing a previous request.
SCE_TOOLKIT_NP_MESSAGE_TYPE_CUSTOM_DATA	(1)	A custom data message.
SCE_TOOLKIT_NP_MESSAGE_TYPE_INVALID	0x86000031	An error occurred because the message type is invalid.
SCE_TOOLKIT_NP_MESSAGE_TYPE_INVITE	(2)	An invite data message.
SCE_TOOLKIT_NP_MESSAGE_TYPE_MISMATCH	0x81000032	An error occurred because the message specification was invalid.
SCE_TOOLKIT_NP_MESSAGE_USER_CANCEL	0x81000030	An error occurred because the user canceled the sending of a message.
SCE_TOOLKIT_NP_MODULE_NOT_OWNED	0x81000015	An error occurred because a module was not owned by NP Toolkit library, and therefore could not be loaded or unloaded.
SCE_TOOLKIT_NP_NEAR_ALREADY_INITED	0x88000001	An error occurred because the “near” service is already initialized.
SCE_TOOLKIT_NP_NEAR_ALREADY_TERMINATED	0x88000002	An error occurred because the “near” service is already terminated.
SCE_TOOLKIT_NP_NEAR_NO_NEIGHBORS	0x88000003	An error occurred because no nearby users were discovered.
SCE_TOOLKIT_NP_NEAR_NO_NEW_NEIGHBORS	0x88000004	An error occurred because no recent nearby users were discovered.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
`SCE_TOOLKIT_NP_NEAR_NO_RECENT_NEIGHBORS`	`0x88000005`	An error occurred because no new nearby users were discovered.
`SCE_TOOLKIT_NP_NETWORK_ERROR`	`0x81000018`	A network error occurred during execution of an operation.
`SCE_TOOLKIT_NP_NO_WORK_MEMORY`	`0x88000006`	An error occurred while trying to allocate memory for the NP Toolkit work area.
`SCE_TOOLKIT_NP_NOT_INITIALISED`	`0x81000016`	An error occurred because the NP Toolkit library was not initialized.
`SCE_TOOLKIT_NP_OFFLINE`	`0x81000019`	A network error occurred as the network was unavailable.
`SCE_TOOLKIT_NP_OPERATION_IN_PROGRESS`	`0x8100001A`	An error occurred because two operations that cannot be performed concurrently were requested. For example, clashing system utilities.
`SCE_TOOLKIT_NP_OUT_OF_DISKSPACE`	`0x81000017`	An error occurred because there was no more disk space.
`SCE_TOOLKIT_NP_OUT_OF_MEMORY`	`0x81000012`	An error occurred because memory allocation failed.
`SCE_TOOLKIT_NP_POLLING_COMPLETE`	`1`	The operation was successfully completed and polling is no longer needed.
`SCE_TOOLKIT_NP_PRESENCE_DATA`	`0x01`	A flag that specifies whether to set presence data.
`SCE_TOOLKIT_NP_PRESENCE_DATA_TOO_BIG`	`0x84000002`	An error occurred because the data passed into `PresenceInterface::setPresence()` was too big.
`SCE_TOOLKIT_NP_PRESENCE_NO_TOKENS`	`0x84000003`	An error occurred because all the presence tokens had been used up.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_PRESENCE_STATUS	0x02	A flag that specifies whether to set presence status.
SCE_TOOLKIT_NP_PRESENCE_STRING_TOO_LONG	0x84000001	An error occurred because the string passed into the NP Toolkit library was too long for setting presence details.
SCE_TOOLKIT_NP_PRESENCE_TYPE_INCONTEXT_INFO	0x04	A flag to specify the type of presence is "in context" and is about the game specific presence of the user.
SCE_TOOLKIT_NP_PRESENCE_TYPE_PLATFORM_INFO	0x02	A flag to specify the type of presence is "platform" and is about the platform specific presence of the user.
SCE_TOOLKIT_NP_PRESENCE_TYPE_PRIMARY_INFO	0x01	A flag to specify the type of presence is "primary" and is about the primary presence of the user.
SCE_TOOLKIT_NP_PUSH_NOTIFICATION_SERVICE_DISABLE_ALL	0	A flag that specifies all Push Notification services should be disabled. Note that by default, all Push Notification services are enabled.
SCE_TOOLKIT_NP_PUSH_NOTIFICATION_SERVICE_ENABLE_ALL	0xFFFFFFFF	A flag that specifies all Push Notification services should enabled. This is the case by default. In order to use all these services, make sure appropriate services are requested through DevSupport.
SCE_TOOLKIT_NP_PUSH_NOTIFICATION_SERVICE_ENABLE_FRIENDS	0x00000004	A flag that specifies that the Friends Push Notification service should be enabled.
SCE_TOOLKIT_NP_PUSH_NOTIFICATION_SERVICE_ENABLE_GAME_CUSTOM_DATA	0x00000008	Flag to enable Friends Push Notification.
SCE_TOOLKIT_NP_PUSH_NOTIFICATION_SERVICE_ENABLE_PRESENCE	0x00000002	A flag that specifies that the Presence Push Notification service should be enabled.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_PUSH_NOTIFICATION_SERVICE_ENABLE_SESSIONS	0x00000001	A flag that specifies that the Session Push Notification service should be enabled.
SCE_TOOLKIT_NP_RANKING_ALREADY_INITED	0x85000014	An error occurred because the ranking service is already initialized.
SCE_TOOLKIT_NP_RANKING_FRIEND_LIST_EMPTY	0x85000007	An error occurred because there was no friends rank in the leaderboard.
SCE_TOOLKIT_NP_RANKING_INVALID_BOARD	0x85000012	An error occurred because an invalid board ID was passed into ranking functions.
SCE_TOOLKIT_NP_RANKING_NO_FRIEND_CACHE	0x85000005	An error occurred because there was no cache for a friend.
SCE_TOOLKIT_NP_RANKING_NO_FRIEND_RANK	0x85000008	An error occurred because there was no previous score in the friend cache.
SCE_TOOLKIT_NP_RANKING_NO_MATCHING_BOARD	0x85000003	An error occurred because there is no ranking board in the cache.
SCE_TOOLKIT_NP_RANKING_NO_MATCHING_SCORE	0x85000006	An error occurred because the same score was not found in the write cache.
SCE_TOOLKIT_NP_RANKING_NO_OUTSTANDING	0x85000013	An error occurred because all the outstanding scores had been registered from the log.
SCE_TOOLKIT_NP_RANKING_NO_RANK_IN_READ_CACHE	0x85000009	An error occurred because there was no previous score in the read cache.
SCE_TOOLKIT_NP_RANKING_NO_SCORES	0x85000010	An error occurred because there was no previous score in the write cache.
SCE_TOOLKIT_NP_RANKING_NOT_HIGH_SCORE	0x85000004	An error occurred because the user's score was not the high score.
SCE_TOOLKIT_NP_RANKING_NOT_INITIALISED	0x85000002	An error occurred because an attempt was made to execute ranking services without initializing them first.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
`SCE_TOOLKIT_NP_RANKING_NOT_TERMINATED`	`0x85000015`	An error occurred because the ranking service has not been terminated.
`SCE_TOOLKIT_NP_RANKING_RANGE_INVALID`	`0x85000011`	An error occurred because there the range specified was too big.
`SCE_TOOLKIT_NP_RANKING_REQUEST_ABORTED`	`0x85000016`	An error occurred because the ranking request has been aborted.
`SCE_TOOLKIT_NP_RANKING_SLOT_FULL`	`0x85000001`	An error occurred because the transaction ID slots are full. There are only 6 slots available.
`SCE_TOOLKIT_NP_ROOM_MESSAGE_CHAT`	`(1<<4)`	This flag specifies that a message is a chat message.
`SCE_TOOLKIT_NP_ROOM_MESSAGE_GENERAL`	`(1<<3)`	This flag specifies that a message is a room message.
`SCE_TOOLKIT_NP_ROOM_MESSAGE_RECEIVED`	`(1<<1)`	This flag specifies that a room message was received.
`SCE_TOOLKIT_NP_ROOM_MESSAGE_SENT`	`(1<<2)`	This flag specifies that a room message was sent.
`SCE_TOOLKIT_NP_SEARCH_FRIENDS_SESSIONS`	`(1<<10)`	This flag specifies that the search is for a friends session.
`SCE_TOOLKIT_NP_SEARCH_NAT_RESTRICTED_SESSIONS`	`(1<<20)`	This flag specifies that the search is for a session with whom a P2P session can be established.
`SCE_TOOLKIT_NP_SEARCH_RANDOM_SESSIONS`	`(1<<18)`	This flag specifies that the search will return a randomly selected session.
`SCE_TOOLKIT_NP_SEARCH_RECENTLY_MET_SESSIONS`	`(1<<14)`	This flag specifies that the search is for a session hosted by users in the Recently Met List.
`SCE_TOOLKIT_NP_SEARCH_REGIONAL_SESSIONS`	`(1<<12)`	This flag specifies that the search is for a session that is hosted in your region.
`SCE_TOOLKIT_NP_SERVICE_ID_NOT_OVERRIDEN`	`0x81000021`	An error occurred because the service could not override the ID required.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
`SCE_TOOLKIT_NP_SERVICE_NOT_INITIALISED`	`0x81000024`	An error occurred because the NP Toolkit library tried to perform an action on a service which has not been initialized.
`SCE_TOOLKIT_NP_SERVICE_OFFLINE`	`0x81000022`	An error occurred because the NP Toolkit library tried to bring the service up while offline.
`SCE_TOOLKIT_NP_SESSION_ATTRIBUTE_MAX_SIZE_12`	`(1<<1)`	The value (`SessionAttributeValue`) of this attribute is a binary value with a maximum size of 12 characters.
`SCE_TOOLKIT_NP_SESSION_ATTRIBUTE_MAX_SIZE_124`	`(1<<4)`	The value (`SessionAttributeValue`) of this attribute is a binary value with a maximum size of 124 characters.
`SCE_TOOLKIT_NP_SESSION_ATTRIBUTE_MAX_SIZE_252`	`(1<<5)`	The value (`SessionAttributeValue`) of this attribute is a binary value with a maximum size of 252 characters.
`SCE_TOOLKIT_NP_SESSION_ATTRIBUTE_MAX_SIZE_28`	`(1<<2)`	The value (`SessionAttributeValue`) of this attribute is a binary value with a maximum size of 28 characters.
`SCE_TOOLKIT_NP_SESSION_ATTRIBUTE_MAX_SIZE_60`	`(1<<3)`	The value (`SessionAttributeValue`) of this attribute is a binary value with a maximum size of 60 characters.
`SCE_TOOLKIT_NP_SESSION_ATTRIBUTE_NOT_REGISTERED`	`0x86000011`	An error occurred because the session attribute are not registered.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_SESSION_ATTRIBUTE_VALUE_BINARY	(1<<2)	The value (`SessionAttributeValue`) of this attribute is a binary value.
SCE_TOOLKIT_NP_SESSION_ATTRIBUTE_VALUE_INT	(1<<1)	The value (`SessionAttributeValue`) of this attribute is an integer.
SCE_TOOLKIT_NP_SESSION_CHANGEABLE_DATA_MAX_SIZE	512	The maximum size of the session changeable data path.
SCE_TOOLKIT_NP_SESSION_DEFAULT	(4)	This flag specifies that session is unlocked and joinable or while updating there is no change. Do not explicitly set this.
SCE_TOOLKIT_NP_SESSION_EXTERNAL_ATTRIBUTE	(1<<2)	This flag specifies that the session attribute is used as external session room data.
SCE_TOOLKIT_NP_SESSION_ID_MAX_SIZE	45	The maximum size of the session ID.
SCE_TOOLKIT_NP_SESSION_IMAGE_PATH_MAX_SIZE	256	The maximum size of the session image path.
SCE_TOOLKIT_NP_SESSION_INTERNAL_ATTRIBUTE	(1<<3)	This flag specifies that the session attribute is used as internal session room data.
SCE_TOOLKIT_NP_SESSION_LOCKED	(0)	This flag specifies that session is locked.
SCE_TOOLKIT_NP_SESSION_MAX_LANGUAGES	10	The maximum number of supported languages for sessions.
SCE_TOOLKIT_NP_SESSION_MAX_NUMBER_ONLINE_IDS	16	The maximum size of the online IDs supported.
SCE_TOOLKIT_NP_SESSION_MEMBER_ATTRIBUTE	(1<<4)	This flag specifies that the session attribute is used as session member data.
SCE_TOOLKIT_NP_SESSION_MEMBER_MYSELF	(1<<2)	This flag specifies that the member is the user themselves (local).
SCE_TOOLKIT_NP_SESSION_MEMBER_OWNER	(1<<1)	This flag specifies that the current member is the session owner.
SCE_TOOLKIT_NP_SESSION_NAME_MAX_PRIVACY_DESC	8	The maximum size of the session privacy description.
SCE_TOOLKIT_NP_SESSION_NAME_MAX_SIZE	64	The maximum size of the session name.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
`SCE_TOOLKIT_NP_SESSION_NAME_SESSION_TYPE_DESC`	`16`	The maximum size of the session type description.
`SCE_TOOLKIT_NP_SESSION_SEARCH_ATTRIBUTE`	`(1<<1)`	This flag specifies that the session attribute is used as a search filter.
`SCE_TOOLKIT_NP_SESSION_STATUS_MAX_SIZE`	`64`	The maximum size of the session status.
`SCE_TOOLKIT_NP_SESSION_UNLOCKED`	`(1)`	This flag specifies that session is unlocked and joinable.
`SCE_TOOLKIT_NP_SIGNALING_DEAD`	`2`	This flag specifies the signaling status of a member compared to current user; in this case, signaling is dead.
`SCE_TOOLKIT_NP_SIGNALING_ESTABLISHED`	`1`	This flag specifies the signaling status of a member compared to current user; in this case, signaling has been established.
`SCE_TOOLKIT_NP_SIGNALING_ESTABLISHED_FAILED_TO_GET_INFO`	`4`	This flag specifies the signaling status of a member compared to current user; in this case, there was a failure to obtain user signaling information.
`SCE_TOOLKIT_NP_SIGNALING_NA`	`0`	This flag specifies the signaling status of a member compared to current user; in this case, signaling is not applicable.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_SKU_PRICE_LEN	(SCE_ TOOLKIT_ NP_ COMMERCE_ CURRENCY_ CODE_ LEN \ + SCE_ TOOLKIT_ NP_ COMMERCE_ CURRENCY_ SYMBOL_ LEN \ + SCE_ TOOLKIT_ NP_ COMMERCE_ THOUSAND_ SEPARATOR_ LEN \ + SCE_ TOOLKIT_ NP_ COMMERCE_ DECIMAL_ LETTER_ LEN)	The total size of the formatted SKU price.
SCE_TOOLKIT_NP_SNS_ACCESS_TOKEN_ERROR	0x89000001	An error occurred while retrieving an access token from Facebook.
SCE_TOOLKIT_NP_SNS_INVALID_MESSAGE	0x89000002	An error occurred because the contents of a Facebook message were invalid.
SCE_TOOLKIT_NP_SNS_MESSAGE_POST_FAILED	0x89000003	An error occurred because an attempt to post a message to Facebook failed.
SCE_TOOLKIT_NP_SUCCESS	0	The operation was successfully completed.
SCE_TOOLKIT_NP_TERMINATED	0x81000005	An error occurred because the NP Toolkit library has been terminated.
SCE_TOOLKIT_NP_TERMINATION_NOT_SUPPORTED	0x81000023	An error occurred because termination of the requested service is not supported.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
`SCE_TOOLKIT_NP_TROPHY_ABORT_FAIL`	`0x87000005`	An error occurred because an to abort a trophy registration failed.
`SCE_TOOLKIT_NP_TROPHY_ALREADY_INITIALISED`	`0x87000009`	An error occurred because the trophy service is already initialized.
`SCE_TOOLKIT_NP_TROPHY_ALREADY_REGISTERED`	`0x87000008`	An error occurred because the trophy sets is already registered.
`SCE_TOOLKIT_NP_TROPHY_BUSY`	`0x87000007`	An error occurred because the trophy service is processing a long request.
`SCE_TOOLKIT_NP_TROPHY_NOT_ENOUGH_SPACE`	`0x8700000B`	An error occurred because there is not enough space to install the trophy set.
`SCE_TOOLKIT_NP_TROPHY_NOT_INITIALISED`	`0x87000002`	An error occurred because the trophy service has not been initialized.
`SCE_TOOLKIT_NP_TROPHY_NOT_REGISTERED`	`0x87000003`	An error occurred because the trophy set file has not been registered.
`SCE_TOOLKIT_NP_TROPHY_NOT_TERMINATED`	`0x8700000A`	An error occurred because the trophy service is not terminated.
`SCE_TOOLKIT_NP_TROPHY_REGISTERING`	`0x87000004`	An error occurred because an attempt to register another trophy is still processing.
`SCE_TOOLKIT_NP_TROPHY_SETUP_DIALOG_ALREADY_RUNNING`	`0x87000006`	An error occurred because an attempt was made invoke more than one setup dialog at the same time.
`SCE_TOOLKIT_NP_TROPHY_UPDATE_ERROR`	`0x87000001`	An error occurred because the trophy list cache cannot be updated.
`SCE_TOOLKIT_NP_TSS_BUFFER_TOO_SMALL`	`0x8B000002`	An error occurred because the buffer passed in was too small.
`SCE_TOOLKIT_NP_TSS_NO_DATA`	`0x8B000001`	An error occurred because there was no data on the TSS server.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_TUS_MAX_SLOTS	16	The maximum number of slots that can be updated at a time.
SCE_TOOLKIT_NP_TUS_NO_DATA	0x8C000002	An error occurred because there was no data on the TUS server.
SCE_TOOLKIT_NP_TUS_NO_VARS	0x8C000001	An error occurred because an invalid parameter which contained no variables was passed into a function.
SCE_TOOLKIT_NP_USER_PROFILE_CACHED_INFORMATION	0x01	A flag that specifies to retrieve information from the cached user profile.
SCE_TOOLKIT_NP_USER_PROFILE_UPDATED_INFORMATION	0x02	A flag that specifies to retrieve updated user profile information.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_ACCESS_DENIED_ PRIVACY	0x80552c80	An error occurred because the access was denied.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_ACCESS_DENIED_ RESOURCE	0x80552c76	An error occurred because access to a resource was attempted by a non-owner.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_BAD_REQUEST	0x80552c78	An error occurred because an invalid value was included in the request.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_BASEURL_INVALID_ API_GROUP	0x80552ca5	An error occurred because the base URL API group was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_BASEURL_UNKNOWN_ CLIENT_ID	0x80552ca6	An error occurred because the base URL Client ID was unknown.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_EXCEEDED_RATE_LIMIT	0x80552c81	An error occurred because the API rate limit was exceeded.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_EXPIRED_ACCESS_ TOKEN	0x80552c88	An error occurred because the access token has expired.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_FRIEND_INVALID_ STATUS	0x80552c9c	An error occurred because the friend status was invalid.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_GAME_CUSTOM_DATA_ALREADY_USED	0x80552cab	An error occurred because the game custom data has already been used (but has not expired).
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_GAME_CUSTOM_DATA_EXPIRED	0x80552caa	An error occurred because the game custom data has expired and is no longer available.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INTERNAL_SERVER_ERROR	0x80552c83	An error occurred because an internal server error occurred.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_ACCESS_TOKEN	0x80552c7d	An error occurred because the access token was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_ENVIRONMENT	0x80552c72	An error occurred because the NP environment name is invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_FIELDS_PARAM	0x80552c77	An error occurred because a query string value is invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_FILTER_PARAM	0x80552ca8	An error occurred because there was an invalid filter parameter.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_LIMIT_PARAM	0x80552c89	An error occurred because the limit parameter was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_OFFSET_PARAM	0x80552c8a	An error occurred because the offset parameter was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_ONLINE_ID	0x80552c75	An error occurred because the Online ID was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_ORDERBY_PARAM	0x80552c8c	An error occurred because the order by parameter was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_PARAM_COMBINATION	0x80552c79	An error occurred because the query value in the query string was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_PLATFORM	0x80552c7a	An error occurred because the platform was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_QUERY_STRING	0x80552c71	An error occurred because the query string was invalid.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_REQUEST_BODY	0x80552c70	An error occurred because the body of the request was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_SESSION_DATA	0x80552c8f	An error occurred because the session data was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_SESSION_IMAGE	0x80552c8e	An error occurred because the session image was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_SESSION_MAX_USER	0x80552c96	An error occurred because the session max user value was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_SESSION_NAME	0x80552c97	An error occurred because the session name was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_SESSION_PLATFORM	0x80552c8d	An error occurred because the session target platform was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_SESSION_PRIVACY	0x80552c95	An error occurred because the session privacy value was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_SESSION_STATUS	0x80552c98	An error occurred because the session status was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_INVALID_SESSION_TYPE	0x80552c94	An error occurred because the session type was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_LARGE_BODY	0x80552c73	An error occurred because the request body was too long.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_NO_SUCH_USER	0x80552c7b	An error occurred because the target user does not exist.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_NOT_ALLOWED_OFFLINE	0x80552c8b	An error occurred because the operation is not allowed offline.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_NOT_AUTHORISED	0x80552c7e	An error occurred because the request was not authorized.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_NOT_AUTHORISED_TITLE	0x80552c82	An error occurred because the title cannot be used for this service.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_NOT_PERMITTED_SESSION_ACCESS	0x80552c90	An error occurred because the session access was not permitted.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_NULL_VALUE	0x80552c67	An error occurred because the WebAPI request had a NULL value.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_PRESENCE_INVALID_COMMS_ID	0x80552ca4	An error occurred because the presence Communication ID was too long.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_PRESENCE_INVALID_PLATFORM	0x80552c9d	An error occurred because the presence platform was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_PRESENCE_INVALID_TITLE_ID	0x80552ca3	An error occurred because the presence Title ID was too long.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_PRESENCE_INVALID_TYPE	0x80552c9e	An error occurred because the presence type was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_PRESENCE_NON_FRIEND_ACCESS	0x80552ca0	An error occurred because access was attempted by a non-owner or non-friend.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_PRESENCE_TOO_LONG_DATA	0x80552ca2	An error occurred because the presence data was too long.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_PRESENCE_TOO_LONG_STATUS	0x80552ca1	An error occurred because the presence status was too long.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_PROFILE_INVALID_AVATAR_SIZE	0x80552c9b	An error occurred because the avatar size was invalid.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_RESOURCE_NOT_FOUND	0x80552c7c	An error occurred because the resource could not be found.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_RESPONSE	0x80552ca7	An error occurred because an HTTP error response was received.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_SERVER_BUSY	0x80552c84	An error occurred because the server is overloaded.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_SERVICE_CLOSED	0x80552c86	An error occurred because the service has been temporarily suspended.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_SESSION_DATA_REQUIRED	0x80552c92	An error occurred because the session data was not supplied.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_SESSION_FULL	0x80552c91	An error occurred because the session was full.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Define	Value	Description
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_SESSION_IMAGE_REQUIRED	0x80552c93	An error occurred because the session image was not supplied.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_SESSION_ONLY_CREATOR_PERMITTED	0x80552c9a	An error occurred because only the session creator is permitted to perform this operation.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_SESSION_ONLY_MEMBER_PERMITTED	0x80552c99	An error occurred because only session members are permitted to perform this operation.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_TITLE_MISMATCH	0x80552c7f	An error occurred because the titles do not match.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_UNDER_MAINTENANCE	0x80552c85	An error occurred because the server is undergoing maintenance.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_UNEXPECTED	0x80552c87	An error occurred because an unexpected error occurred.
SCE_TOOLKIT_NP_WEBAPI_HTTP_ERROR_UNSUPPORTED_API	0x80552c74	An error occurred because the API is unsupported.
SCE_TOOLKIT_NP_WORD_FILTER_NOT_INITIALISED	0x8A000002	An error occurred because the word filter service has not been initialized yet.
SCE_TOOLKIT_NP_WORD_FILTER_SLOT_FULL	0x8A000001	An error occurred because more than 32 requests have been made to the word filter service.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117