

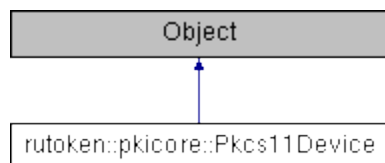
Класс Pkcs11Device

Устройство, подключенное к компьютеру. Может быть получено с помощью вызова `Pkcs11Device::enumerate`.

Пространство имен: `rutoken::pki-core`

Заголовочный файл: `pki-core-cpp.h`

Граф наследования



Классы

class CommonKeyGenParams	Общие параметры генерации ключевой пары для всех типов ключей независимо от алгоритма
class Gost34102001KeyGenParams	Параметры генерации ключевой пары, специфичные для алгоритма ГОСТ 34.10-2001

Структуры

struct Feature	Дополнительная функциональность, поддерживаемая устройством
struct Format	Форматы данных, которые устройство может подписывать

Открытые члены

<code>Pkcs11Device (Pkcs11Device &&v)</code>	
<code>Pkcs11Device & operator= (Pkcs11Device &&v)</code>	
<code>void changePin (const std::string &oldPin, const std::string &newPin)</code>	Смена пользовательского PIN устройства
<code>void deleteCert (const Pkcs11Cert &cert)</code>	Удаление сертификата с устройства
<code>void deleteKeyPair (const Pkcs11PrivateKey &key)</code>	Удаление ключевой пары с устройства
<code>std::vector< Pkcs11Cert > enumerateCerts ()</code>	Получение списка сертификатов, которые хранятся на устройстве
<code>std::vector< Pkcs11PrivateKey > enumeratePrivateKeys ()</code>	Получение списка закрытых ключей, которые хранятся на устройстве
<code>std::vector< Pkcs11UserCert > enumerateUserCerts ()</code>	Получение списка сертификатов, которые связаны с закрытыми ключами
<code>Pkcs11PrivateKey generateKeyPair (const Gost34102001KeyGenParams &keyGenParams)</code>	Генерация ключевой пары ГОСТ Р 34.10-2001 на устройстве
<code>std::string getLabel ()</code>	Получение метки устройства
<code>std::string getSerialNumber ()</code>	Получение серийного номера устройства

<code>Pkcs11Cert importCert (const ExternalCert &externalCert, bool trusted=false)</code>	Сохранение сертификата на устройстве
<code>Pkcs11UserCert importUserCert (const ExternalCert &externalCert)</code>	Сохранение сертификата на устройстве
<code>bool isFeatureSupported (Feature::Type feature)</code>	Поддерживается ли возможность устройством
<code>bool isFormatSupported (Format::Type format)</code>	Поддерживается ли устройством формат данных
<code>bool isLoggedIn ()</code>	Произведена ли авторизация на устройстве
<code>void login (const std::string &pin)</code>	Авторизация на устройстве
<code>void logout ()</code>	Сброс прав доступа на устройстве

Открытые статические члены

<code>static Pkcs11Device cast (Object &&v)</code>	
<code>static std::vector< Pkcs11Device > enumerate ()</code>	Получение списка подключенных устройств

Подробное описание

<code>static std::vector< Pkcs11Device > enumerate</code>	<code>()</code>
---	-----------------

Получение списка подключенных устройств.

Вызов этой функции делает объекты, полученные при предыдущем вызове, недействительными.

<code>void changePin</code>	<code>(const std::string &oldPin, const std::string &newPin)</code>
-----------------------------	---

Смена PIN пользователя.

Аргументы

`oldPin` - старый PIN пользователя.

`newPin` - новый PIN пользователя.

<code>void deleteCert</code>	<code>(const Pkcs11Cert &cert)</code>
------------------------------	---

Добавление произвольного расширения в запрос.

Аргументы

`cert` - сертификат, который необходимо удалить

<code>void deleteKeyPair</code>	<code>(const Pkcs11PrivateKey &key)</code>
---------------------------------	--

Удаление ключевой пары с устройства.

Аргументы

key - закрытый ключ, который необходимо удалить.

<code>std::vector< Pkcs11Cert ></code>	<code>enumerateCerts ()</code>
--	--------------------------------

Перечисление всех сертификатов, который хранятся на устройстве.

<code>std::vector< Pkcs11PrivateKey ></code>	<code>enumeratePrivateKeys ()</code>	<code>enumeratePrivateKeys ()</code>
--	--------------------------------------	--------------------------------------

Получение списка закрытых ключей, хранящихся на устройстве.

<code>std::vector< Pkcs11UserCert ></code>	<code>enumerateUserCerts ()</code>
--	------------------------------------

Получение списка сертификатов, для которых есть закрытый ключ на устройстве.

<code>Pkcs11PrivateKey generateKeyPair</code>	<code>(const Gost34102001KeyGenParams &keyGenParams)</code>
---	---

Генерация ключевой пары ГОСТ Р 34.10-2001 на устройстве.

Аргументы

keyGenParams - параметры генерации ключевой пары.

Возвращает

Созданный ключ.

<code>std::string</code>	<code>getLabel ()</code>
--------------------------	--------------------------

Получение названия устройства.

<code>std::string</code>	<code>getSerialNumber ()</code>
--------------------------	---------------------------------

Получение серийного номера устройства.

<code>Pkcs11Cert importCert</code>	<code>(const ExternalCert &externalCert, bool trusted=false)</code>
------------------------------------	---

Запись сертификата на устройство.

Аргументы

externalCert - внешний сертификат, который требуется записать на устройство.

trusted - сертификат будет записан как "CA". При этом он будет доверенным в рамках устройства.

Возвращает

Сертификат, связанный с устройством.

<code>Pkcs11UserCert importUserCert</code>	<code>(const ExternalCert &externalCert)</code>
--	---

Запись сертификата на устройство. Для данного сертификата есть закрытый ключ на устройстве.

Аргументы

externalCert - внешний сертификат, который требуется записать на устройство..

Возвращает

Сертификат, связанный с устройством.

<code>Pkcs11Cert importCert</code>	<code>(const ExternalCert &externalCert, bool trusted=false)</code>
------------------------------------	---

Запись сертификата на устройство.

Аргументы

externalCert - внешний сертификат, который требуется записать на устройство.

trusted - сертификат будет записан как "CA". При этом он будет доверенным в рамках устройства.

Возвращает

Сертификат, связанный с устройством.

<code>bool isFeatureSupported</code>	<code>(Feature::Type feature)</code>
--------------------------------------	--------------------------------------

Получение информации о поддерживаемых устройством дополнительных возможностях.

Аргументы

feature - дополнительная возможность, поддержку которой необходимо проверить.

<code>bool isFormatSupported</code>	<code>(Format::Type format)</code>
-------------------------------------	------------------------------------

Получение информации о поддерживаемых устройством форматах данных для подписи.

Аргументы

format - формат, поддержку которого необходимо проверить.

<code>bool isLoggedIn</code>	<code>()</code>
------------------------------	-----------------

Была ли произведена успешная авторизация на устройство.

<code>void login</code>	<code>(const std::string &pin)</code>
-------------------------	---

Авторизация на устройстве.

Аргументы

pin - PIN пользователя.

<code>void logout</code>	<code>()</code>
--------------------------	-----------------

Сброс прав доступа на устройство.