Проверка ЭП на сервере.

Необходимо реализовать сервис проверки ЭП на сервере. Сервис должен быть выполнен в виде EJB со следующим интерфейсом:

/\*\*

\* Сервис для выполнения криптографических операций на платформе CM5.

\* **@author** larin

\*

\*/

**public** **interface** CryptoTool{

/\*\*

\* Проверка усовершенствованной ЭП включеннной в документ

\* **@param** inputStream поток содержащий документ со встроенной электронной подписью

\* **@return**

\*/

VerifyResult verify(InputStream document);

/\*\*

\* Проверка Усовершенствованной ЭП не включенной в документ

\* **@param** document поток содержащий документ

\* **@param** signature электронная подпись

\* **@return**

\*/

VerifyResult verify(InputStream document, **byte**[] signature);

/\*\*

\* Проверка стандартной ЭП

\* **@param** document поток содержащий документ

\* **@param** signature электронная подпись

\* **@param** signerSertificate сертификат подписавшено документ в формате DER

\* **@return**

\*/

VerifyResult verify(InputStream document, **byte**[] signature, **byte**[] signerSertificate);

/\*\*

\* Проверка стандартной ЭП с помощью сертификата, указанного в доменном объекте персоны

\* **@param** document поток содержащий документ

\* **@param** signature электронная подпись

\* **@param** personId идентификатор персоны которой был подписан документ

\* **@return**

\*/

VerifyResult verify(InputStream document, **byte**[] signature, Id personId);

/\*\*

\* Проверка стандартной ЭП с помощью сертификата, указанного в доменном объекте персоны

\* @param documentId идентификатор документа

\* @return

\*/

VerifyResult verify(Id documrntId);

}

В качестве возвращаемого значения методов проверки ЭП используется структура:

public class VerifyResult implements Dto{

private static final long *serialVersionUID* = -7995499379143873587L;

private boolean valid;

private List<String> messages;

private List<String> warnings;

private List<String> errors;

private String signer;

public List<String> getMessages() {

return messages;

}

public void setMessages(List<String> messages) {

this.messages = messages;

}

public List<String> getWarnings() {

return warnings;

}

public void setWarnings(List<String> warnings) {

this.warnings = warnings;

}

public List<String> getErrors() {

return errors;

}

public void setErrors(List<String> errors) {

this.errors = errors;

}

public boolean isValid() {

return valid;

}

public void setValid(boolean valid) {

this.valid = valid;

}

public String getSigner() {

return signer;

}

public void setSigner(String signer) {

this.signer = signer;

}

}

EJB должен вызывать спринг бин с конкретной реализацией, использующей конкретный крипто провайдер. Интерфейс спринг бина аналогичен интерфейсу EJB. Для исключения дублирования интерфейсов необходимо создать 2 интерфейса наследников CryptoTool: CryptoService и CryptoBean. Уои должен имплементировать интерфейс CryptoService а спринг бин – CryptoBean. Имплементация спринг бина должна быть вынесена в отдельный мавен артифакт, например crypto-pro-crypto-service и опционально подключен в итоговую сборку платформы. beans.xml файл проекта имплементации должен объявлять спринг бин в профиле отличном от default и далее нужная имплементация должна подключатся с помощью выбора профиля spring в файле standalone.xml. Пример файла конфигурации спринг бина:

<beans profile=*"crypto-pro"*>

<bean id=*"cryptoService"* class=*"ru.intertrust.cm.core.dao.impl.CryptoProCryptoService"* />

</beans>

Подключение конкретной реализации в standalone.xml

<system-properties>

...

<property name="spring.profiles.active" value="default, *crypto-pro*"/>

</system-properties>

Структура VerifyResult содержит информацию о результате проверки. Поле valid указывает на корректность вычисления крипто функций. Поле signer содержит информацию о подписавшем. Структура так же может содержать дополнительную информацию по проверки ЭП разделенную на 3 категории. Информационные сообщения в поле messages, например использующиеся алгоритмы ЭП, шифрования или хеширования. Предупреждающая информация warnings, например отсутствие доверенного корневого сертификата в цепочке сертификатов подписавшего. Информация об ошибке errors, например ошибка при вызове криптографических функций.

Для реализации метода VerifyResult verify(InputStream document, **byte**[] signature, Id personId) необходимо изменить ДО персоны для возможности хранения нескольких сертификатов для одного пользователя. Для этого создается новый ДО person\_certificate со следующими полями:

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Примечания |
| person\_id | Ссылка на персону | Идентификатор персоны |
| certificate | Text | Сертификат в формате base64 |

При проверки ЭП методом VerifyResult verify(InputStream document, **byte**[] signature, Id personId) необходимо производить проверку со всеми сертификатами персоны до получения хотя бы одного валидного результата.

Для работы функции VerifyResult verify(Id documrntId) необходимо создать ДО содержащей ЭП доменных объектов со следующими полями.

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип | Примечания |
| document\_id | Ссылка на любой доменный объект | Идентификатор подписанного доменного объекта |
| certificate\_id | Ссылка на объект типа person\_certificate | Идентификатор сертификата, которым был подписан документ |
| algorithm | Строка | Имя алгоритма подписи. Алгоритм подписи устанавливается как конфигурация платформы и содержит описание полей до, которые участвуют в формирование строки, которая затем подписывается. Конфигурация привязывается к типу доменного объекта. Формат XML в данном документе не описывается и будет уточнен на этапе разработки функциональности по подписыванию документов с помощбю ЭП. |

Реализация метода VerifyResult verify(Id documrntId) надо производить после разработки функционала подписывания документов с помощью ЭП.