Система онлайн-сервисов HT-Line®

Сервисы онлайн-тестирования
Мастер-Тесты и Maintest®-5i

ОПИСАНИЕ ИНТЕРФЕЙСА API

Версия интерфейса: 1.6
Дата выпуска: 14.09.2015

Лаборатория Гуманитарные Технологии
www.ht.ru

ЛИСТ ИЗМЕНЕНИЙ

Информация об изменениях в API представлена в таблице:

Версия	Дата	Описание основных изменений
1.2.2	23.11.2012	Добавлены методы взаимодействия с СДО HT-Line: isLMSServiceAvailable(), getLMSRespondentsCount(), getLMSRespondentCard(), getLMSRespondentsList(), setLMSSessionRespondent()
1.2.4	01.12.2012	Расширен перечень выходных параметров для метода getTestsAttribsList()
1.2.7	23.12.2012	Расширен перечень входных параметров для метода getResultsReportXml()
1.3.0	05.03.2013	Добавлен метод для взаимодействия с системой Maintest-6: getMT5TestContent()
1.3.1	16.05.2013	Добавлен метод для с результатами тестирования getResultsValuesList()
1.3.2	23.01.2014	Добавлены методы для работы с механизмом прерывания тестирования: getTestSetingsInterrupts(), isTestingSessionInterrupted()
1.3.2.1	10.06.2014	В описание метода isTestingSessionComplete() добавлена информация о возможности использования колбеков – методов обратного вызова
1.3.2.2	25.11.2014	Исправление ошибок, реструктуризация отдельных пунктов документа
1.4.0	01.12.2014	Добавлен метод поиска протоколов по данным анкеты getResponderProtocolsList() и соответствующие типы данных. Добавлен метод генерации отчета в XML-формате по ProtocolId getResultsReportByProtocolIdXml(). Замечания к методам генерации PDF-отчета.
1.5.0	26.11.2014	Рефакторинг генерации отчетов
1.6.0	16.12.2014	Изменена структура файлов. Добавлен метод получения контента в формате DOCX getResultsReportDocx() и getResultsReportDocxEx(). Добавлены методы ля взаимодействия с Maintest 6.

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ

1.1. Информация о документе

Документ описывает прикладной программный интерфейс системы HT-Line в части ее сервисов, предназначенных для проведения онлайн-тестирования.

Документ предназначен для разработчиков программного обеспечения, реализующих интеграцию тестов системы HT-Line в информационную систему заказчика. Для успешной работы с документом разработчики должны быть знакомы в той или иной степени с такими технологиями, как WebServices, WSDL, XML, SOAP.

1.2. Назначение интерфейса

Прикладной программный интерфейс (ApplicationProgrammingInterface, далее по тексту API или интерфейс) системы HT-Line предназначен для интеграции тестовых методик системы HT-Line (тестов) в информационную систему заказчика.

Тестовые методики HT-Line – это профессиональные психометрические тесты, разработанные в Лаборатории Гуманитарные Технологии. Более двадцати лет Лаборатория успешно занимается задачами профессиональнойразработки психометрического тестового контента, используемого для решения задач оценки персонала. Разработанные Лабораторией тестовые методики позволяют не только получить данные участников тестирования по первичным психологическим факторам, но также спроецировать эти факторы на модели компетенций, используемые в организации заказчика.

Информационная система заказчика – это система с пользовательским веб-интерфейсом, установленная на Интернет- или интранет-сервере заказчика и имеющая доступ к веб-сервисам системы HT-Line. С точки зрения функционала, в качестве такой системы обычно используются корпоративный сайт или портал заказчика, система дистанционного обучения или кадровая информационная система заказчика.

Таким образом, интерфейс позволяет интегрировать в информационную систему заказчика профессиональные психометрические инструменты

Лаборатории и использовать их в автоматизированном режиме для решения заказчиком задач оценки персонала.

2. ПОДКЛЮЧЕНИЕ К ИНТЕРФЕЙСУ

2.1. Условия подключения к интерфейсу

Для работы с системой HT-Line через интерфейс API необходимо выполнение следующих условий:

- наличие учетной записи (личного кабинета) в системе HT-Line;
- для личного кабинета должна быть подключена услуга доступа к API;
- в личном кабинете должен быть подключен сервис онлайн-тестирования.

Для возможности использования API также необходимо наличие среды разработки или языка программирования, в которых реализована поддержка работы с веб-сервисами или протоколом SOAP.

Для быстрого подключения к интерфейсу рекомендуется использовать файл спецификации интерфейса в формате WSDL, который доступен на сервере HT-Line.

2.2. Спецификация интерфейса WSDL

В соответствии с правилами реализации веб-сервисов, структура интерфейса полностью описана при помощи спецификации в формате WSDL-документа. Спецификация текущей версии интерфейса в виде WSDL-документа доступна по следующимадресам (в зависимости от того, какой сервис является приоритетным в личном кабинете клиента – Мастер-Тесты или Maintest-5i):

https://**client.ht-line.ru**/m-tests/api/**1.6**/wsdl/
https://**client.ht-line.ru**/maintest-5i/api/**1.6**/wsdl/[1]

Как правило, наличие WSDL-спецификации позволяет разработчику автоматически сгенерировать описание класса SOAP-клиента для доступа к серверу с API, однако, такая возможность зависит от используемой среды разработки и языка программирования. Вообще, наличие WSDL-спецификации не является обязательным условием, но её использование обычно помогает упростить и ускорить работу с API.

[1] Номер версии и наименование сервера могут отличаться от указанных в руководстве и направляются каждому клиенту индивидуально вместе с логином и паролем для доступа к API

2.3. Авторизация пользователя в интерфейсе

При работе с интерфейсом для проверки прав доступа используется процедура авторизации пользователя. В качестве параметров авторизации используются логин и пароль пользователя для доступа в личный кабинет системы HT-Line.

Процедура авторизации основана на механизме HTTP-авторизации, при этом параметры авторизации (логин и пароль к личному кабинету) передаются при вызове каждого метода интерфейса.

2.4. Инструмент для интерактивной отладки

Для удобства первичного ознакомления и дальнейшей работы разработчика с интерфейсом API, ему предоставляется доступ в отладочный веб-интерфейс, который позволяет в интерактивном режиме выполнять методы интерфейса API.

Данный инструмент может быть полезен при изучении методов API, создания сценариев использования API и при отладке собственного приложения, работающего с API.

Отладочный веб-интерфейс доступен по следующимадресам (в зависимости от того, как сервис является приоритетным в личном кабинете клиента – Мастер-Тесты или Maintest-5i):

https://**client.ht-line.ru**/m-tests/api/**1.6**/guide/
https://**client.ht-line.ru**/maintest-5i/api/**1.6**/guide/

Как и для доступа к API, для доступа к отладочному веб-интерфейсу необходимо пройти процедуру авторизации, используя логин и пароль для доступа к личному кабинету HT-Line.

2.5. Пример запроса к интерфейсу

В качестве простого примера использования API рассмотрим вызов метода getApiVersionNumber(), который возвращает номер текущей версии интерфейса. В качестве языка программирования в примере используется PHP.

Пример 1. Запрос номера текущей версии API.

```
// Настройки подключения SOAP-клиента
$options = array(
      "encoding"  => "utf-8",
      "login"     => "user",
      "password"  => "pass"
);


// Для эффективности используем локальный WSDL-файл
$wsdlpath = "htline.wsdl";

// Создаем объект SOAP-клиента
$SoapClient = new SoapClient($wsdlpath,$options);

// Выполняемзапросномераверсии
$SoapResponse = $SoapClient->getApiVersionNumber();
$VersionNumber = $SoapResponse["VersionNumber"];

// Выводимнаэкранномерверсии
echo "HT-Line API version number: " . $VersionNumber;
```

В примере используется локально сохраненная копия файла WSDL-спецификации интерфейса. Это позволяет упростить работу с интерфейсом, так как в файле WSDL содержится вся необходимая информация для работы SOAP-клиента. В том случае, если WSDL не используется, нужно в массиве options заполнить такие поля, как location и uri, содержащие ссылки на адрес интерфейса и пространство имен.

Более подробную информацию об использовании SOAP-клиента и WSDL-спецификации Вы можете найти в документации к своей среде разработке, языку программирования или библиотеке, используемой для работы с SOAP.

3. ТИПОВЫЕ СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

Интерфейс обладает достаточной гибкостью и позволяет реализовывать различные сценарии работы, в зависимости от задач и возможностей заказчика.

3.1. Краткое описание сценариев использования

Типовыми сценариями использования интерфейса для интеграции тестов в информационную систему заказчика являются следующие сценарии:

- Сценарий 1: Стандартная интеграция
  (тестирование и генерация отчетов в системе HT-Line).
- Сценарий 2: Расширенная интеграция
  (тестирование в системе заказчика, генерация отчетов - в HT-Line).
- Сценарий 3: Специальная интеграция
  (тестирование и генерация отчетов в системе заказчика).

С учетом универсальности интерфейса кроме типовых сценариев интеграции возможны также альтернативные сценарии, рассчитанные под потребности конкретного заказчика.

Основные различия в распределении функций между системой HT-Line и ИС заказчика при различных сценариях интеграции представлены в таблице ниже.

Распределение функций при различных сценариях интеграции	**Стандартная интеграция**	**Расширенная интеграция**	**Специальная интеграция**	**Произвольный сценарий**
Предъявление тестовых заданий (тестовый плеер)	HT-Line (через IFRAME)	ИС заказчика (свой плеер)	ИС заказчика (свой плеер)	HT-Line или ИС заказчика
Обработка протокола тестирования (ответов участника на вопросы теста)	HT-Line	HT-Line	HT-Line	HT-Line
Генерация HTML-страниц отчетов с результатами тестирования	HT-Line (через IFRAME)	HT-Line (через IFRAME)	ИС заказчика	HT-Line или ИС заказчика
Генерация структуры отчетов с результатами тестирования в формате XML	HT-Line	HT-Line	HT-Line	HT-Line

Генерация документов отчетов с результатами тестирования в форматах HTML/DOC/PDF	HT-Line	HT-Line или ИС заказчика	ИС заказчика	HT-Line или ИС заказчика

3.1.1. Сценарий 1: Стандартная интеграция.

В этом сценарии система HT-Line выполняет максимальное число функций:

- реализует режим тестирования при помощи своего тестового плеера;
- сохраняет и обрабатывает полученные протоколы тестирования;
- генерирует индивидуальные отчеты по результатам тестирования.

Информационная система заказчика может работать с системой HT-Line, используя фрейм (IFRAME) для загрузки тестового плеера и отчетов с результатами тестирования на страницы, загружаемые пользователями системы.

Архитектура тестового плеера позволяет кастомизировать внешний вид интерфейса. Если заказчик желает кастомизировать предъявляемый тестовый плеер, он должен предоставить макет интерфейса разработчикам HT-Line. Соответствующие работы закладываются в зависимости от сложности макета.

Сценарий удобен для быстрого встраивания тестов в корпоративный сайт, систему дистанционного обучения или кадровую информационную систему заказчика.

Этот сценарий будет подробно рассмотрен в данном руководстве далее.

3.1.2. Сценарий 2: Расширенная интеграция

В отличие от первого типового сценария в этом сценарии предъявление теста респонденту происходит через тестовый плеер информационной системы заказчика. Система HT-Line выполняет роль обработчика протоколов тестирования и генератора отчетов.

Такой подход позволяет унифицировать пользовательский веб-интерфейс, с которым работает респондент (ему показывается привычный для него тестовый плеер той системы, с которой он обычно работает), что упрощает работу с системой для конечного пользователя.

Данный сценарий используется для интеграции с такими информационными системами заказчика, как системы дистанционного обучения или системы тестирования, которые имеют развитый функционал предъявления тестовых заданий (развитый тестовый плеер). При этом тестовый плеер системы заказчика должен быть совместим с типами тестовых заданий и правилами их предъявления, используемыми в тестах системы HT-Line.

3.1.3. Сценарий 3: Специальная интеграция

В отличие от первого и второго типовых сценариев в этом сценарии система HT-Line выполняет наиболее узкую роль, которая заключается в обработке протоколов тестирования и генерации структуры отчета.

Информационная система заказчика при таком подходе берет на себя реализацию функций предъявления теста и функций генерации отчетов, а также, возможно, функции использования числовых данных, полученных по результатам тестирования (в качестве которых могут выступать, например, оценки по компетенциям).

Этот сценарий является наиболее сложным в реализации, однако позволяет наиболее гибко встроить тестовые возможности системы HT-Line в собственную информационную систему заказчика, так как внешний вид и процедуры тестирования, и отчетов по результатам тестирования, полностью определяются правилами информационной системы заказчика.

3.2. Описание сценария стандартной интеграции

Рассмотрим более подробно сценарий стандартной интеграции. Этот сценарий является единственным доступным для большинства заказчиков, в связи с тем, что второй и третий сценарии требуют трудоемкой реализации в информационной системе заказчика механизма тестового плеера, удовлетворяющего структуре и логике тестовых методик системы HT-Line. Второй и третий сценарии, как правило, используются только в случае интеграции с системами дистанционного обучения и системами тестирования, имеющими развитые механизмы предъявления тестовых заданий.

3.2.1. Обобщенный алгоритм работы

Основная логика работы сценария выглядит следующим образом:

1) Создание нового сеанса тестирования:
   Метод createTestingSession().

2) Получение ссылки для сеанса тестирования:
   Метод getTestingSessionUrl() или getTestingSessionUrlEx().

3) Загрузка полученной ссылки во фрейм на странице в информационной системе заказчика.

4) Периодическая проверка статуса сеанса тестирования:
   Метод isTestingSessionComplete().

5) По окончании тестирования – получение ссылки на результат:
   Метод getResultsReportUrl().

6) Загрузка полученной ссылки во фрейм на странице в информационной системе заказчика.

7) При необходимости – получение результатов в других форматах:
   Методы: getResultsReportHtml(), getResultsReportHtmlEx(), getResultsReportPdf(), getResultsReportPdfEx(), getResultsReportXml() и getResultsValuesList().

8) Сохранение полученных результатов в других форматах в системе.

3.2.2. Параметры вызова методов

Метод createTestingSession()

Параметры вызова:

- TestName – системное имя теста.
  Это идентификатор теста в каталоге тестов. Список доступных в каталоге тестов со всеми их атрибутами можно заранее получить при помощи метода getTestsAttribsList().

- ExternalSessionGuid – идентификатор сеанса тестирования.
  Это произвольный уникальный идентификатор сеанса тестирования, определяемый на стороне удаленной системы. Этот идентификатор в

дальнейшем используется почти во всех последующих запросах к интерфейсу.

- IsAnonymousSession – признак анонимного сеанса.
  В случае установки значения параметра в 1, тестовый плеер не выводит анкету с личными данными, которая обычно предъявляется в начале тестирования, и тестирование проходит без передачи персональных данных респондента в систему HT-Line.

Для большинства остальных методов основным параметром является параметр ExternalSessionGuid – использованный при создании сеанса ранее уникальный идентификатор сеанса тестирования.

Это относится к следующим методам:
createTestingSession(), getTestingSessionUrl(), isTestingSessionComplete(). getResultsReportUrl(). getResultsReportHtml(), getResultsReportPdf(), getResultsReportXml() и getResultsValuesList().

3.2.3. Типовой алгоритм

Рассмотрим типовой алгоритм интеграции Портала X (далее «портал») с системой HT-Line.

Для любой системы Заказчика система HT-Line представляет собой совокупность контента, тестового плеера и базы данных с результатами тестирования (рисунок 3.1).

Контент представляет собой саму тестовую методику, алгоритмы расчета шкал и показателей этой методики, словесные интерпретации результатов тестирования.

Тестовый плеер – это интерфейс предъявления заданий теста. Респондент в тестовом плеере отвечает на вопросы, и его результаты сохраняются в БД HT-Line.

БД результатов хранит данные о сеансе тестирования, самом тестировании, а также рассчитанные результаты в различных видах (сырой балл, бал в процентах и т.д.).

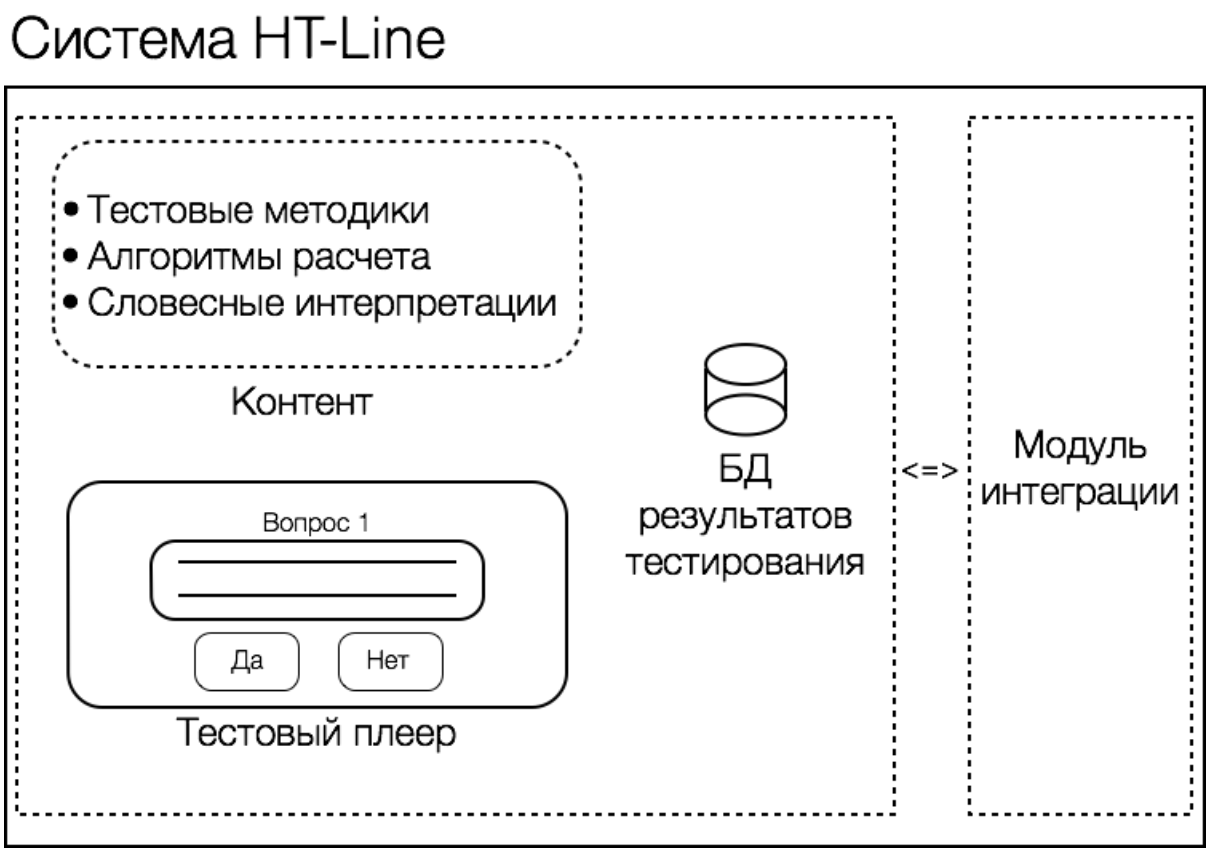

Рисунок 3.1.

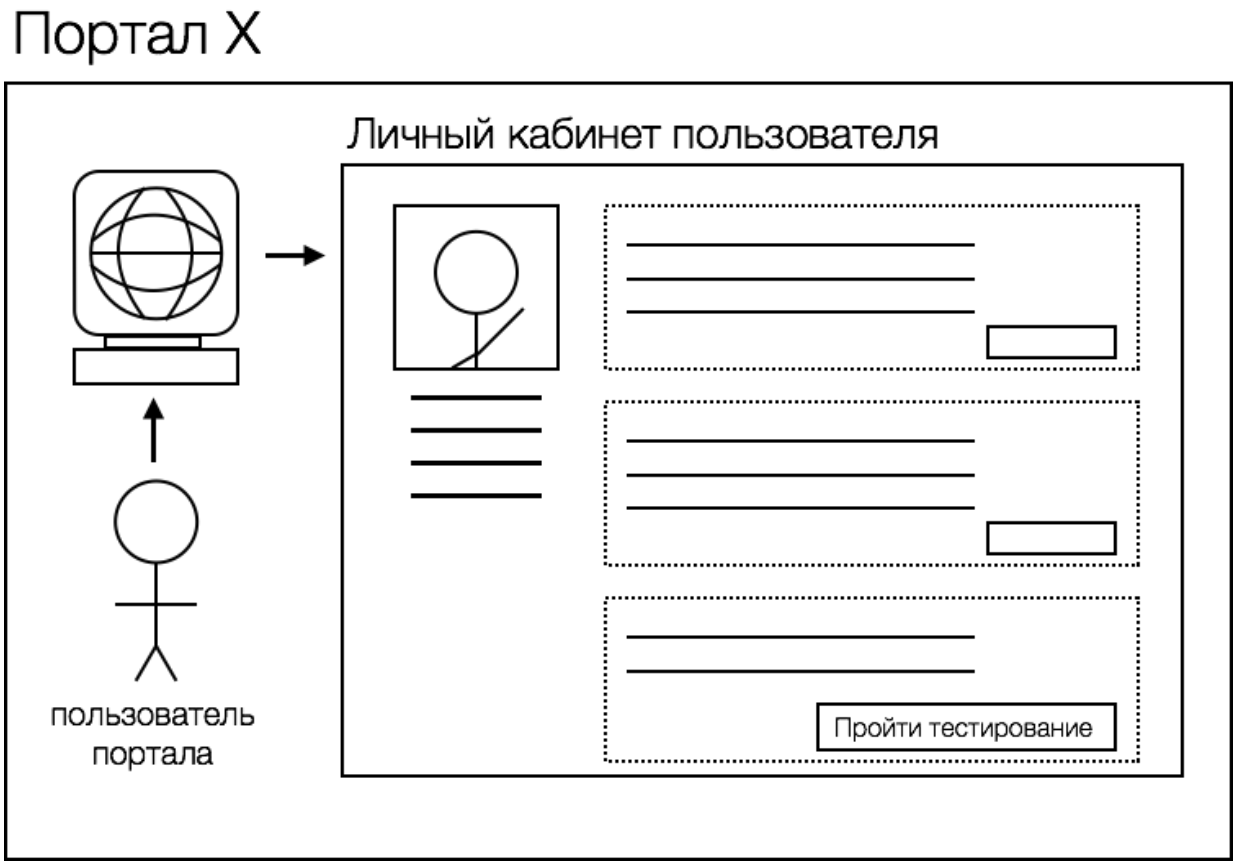

Рисунок 3.2.

Предположим, что портал для конечного пользователя (кто и будет в дальнейшем проходить тестирование) представляет собой личный кабинет. Помимо другого функционала личного кабинета на портале, у этого пользователя есть возможность пройти тестирование. В примере, изображенном на рисунке 3.2, данный функционал представлен кнопкой «Пройти тестирование», нажав на которую пользователь перейдет на страницу с тестовым плеером.

В таком случае типовой алгоритм взаимодействия Портала (П) с системой HT-Line (HTL) будет выглядеть следующим образом:

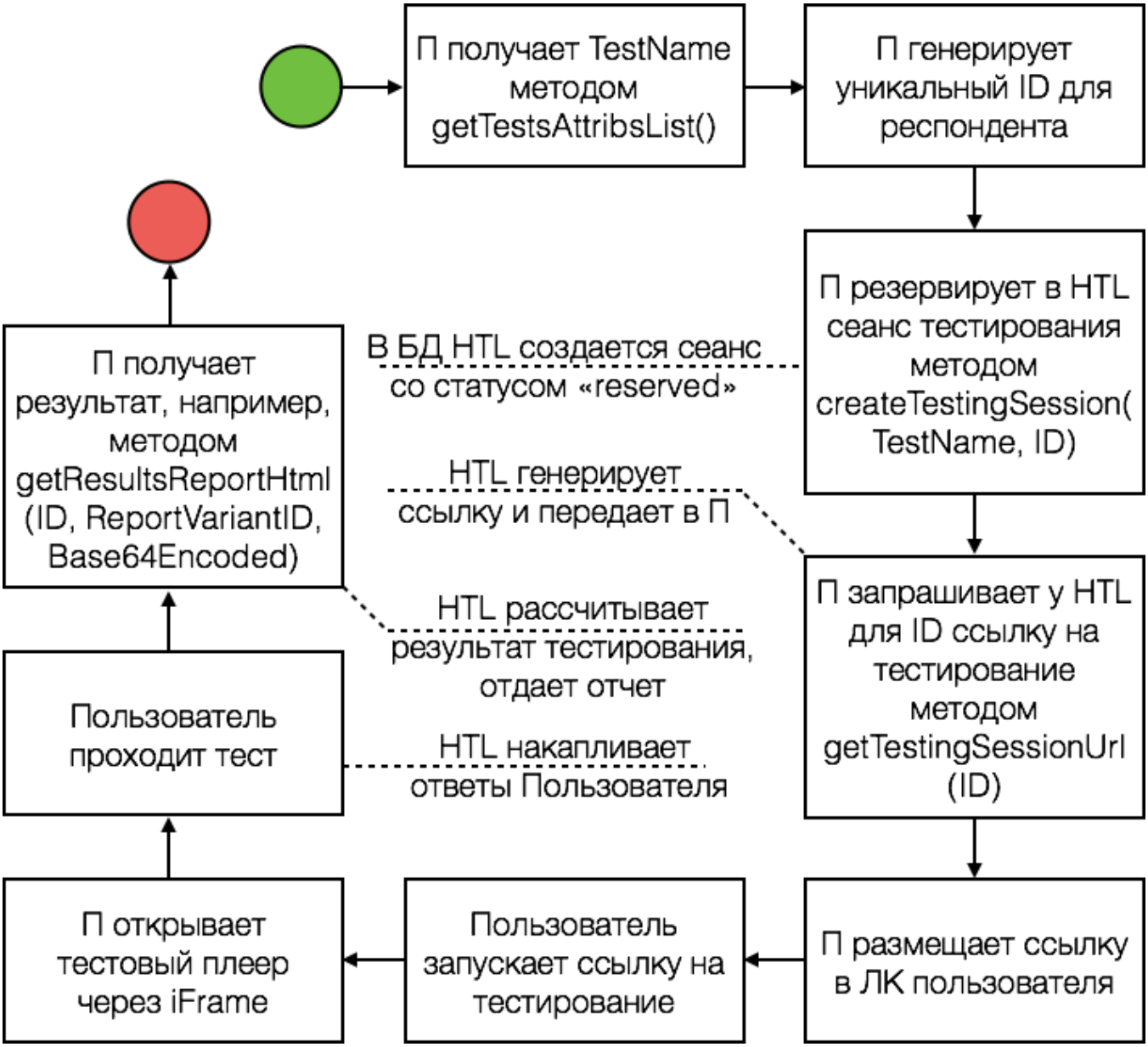

Рисунок 3.3.

4. СПИСОК МЕТОДОВ ИНТЕРФЕЙСА

В состав интерфейса входят методы, которые можно разделить на следующие группы:

- методы для работы с каталогом тестов;
- методы для управления сеансами тестирования;
- методы для работы с протоколами тестирования
- методы доступа к результатам тестирования;
- методы для работы с настройками тестовых методик;
- методы для работы с системой СДО НТ-Line*;
- методы для работы с системой Maintest-6*;
- сервисные и дополнительные методы.

** описание отсутствует в данном документе.*

Методы для работы с каталогом тестов позволяют получать информацию о доступных в системе тестах: количестве тестов, списке тестов, числе доступных лицензий по тестам, списках измерительных шкал для каждого теста.

Методы для управления сеансами тестирования предназначены для работы с сеансами тестирования - создания, запуска и проверки состояния сеансов тестирования.

Методы доступа к результатам тестирования позволяют получать доступ к результатам тестирования в виде числовых значений по шкалам теста (массив чисел), в виде контента документов отчетов (HTML, PDF) или в виде структурированного контента отчета (XML).

В данном руководстве также присутствуют устаревшие методы – методы, которые существовали в первых версиях API, но были исключены в последующих. Такие методы помечаются следующим образом: метод$^{\text{УСТАРЕВШИЙ}}$.

К тому же существуют основные методы (метод $^{\text{ОСНОВНОЙ}}$) – такие методы, которые являются необходимыми или наиболее информативными и эффективными с точки зрения разработки.

4.1. Методы для работы с каталогом тестов

4.1.1. Метод getTestsCount()

Получение числа тестов, доступных для текущего пользователя.

Входные параметры: отсутствуют

Выходные параметры:

Параметр	Тип	Описание
TestsCount	integer	число тестов, доступных для текущего пользователя
OperationStatus	TypeOperationStatus	результат выполнения операции

Внимание! Для оптимизации работы со списком тестов рекомендуется использовать метод getTestsAttribsList(), который в результате выполнения одного запроса выдает полный список тестов со всеми атрибутами.

4.1.2. Метод getTestName()

Получение системного имени теста (по индексу теста в массиве).

Входные параметры:

Параметр	Тип	Описание
TestIndex	integer	индекс теста в массиве тестов (0 <= TestIndex<TestCount)

Выходные параметры:

Параметр	Тип	Описание
TestName	string	системное имя теста
OperationStatus	TypeOperationStatus	результат выполнения операции

Внимание! Для оптимизации работы со списком тестов рекомендуется использовать метод getTestsAttribsList(), который в результате выполнения одного запроса выдает полный список тестов со всеми атрибутами.

4.1.3. Метод getTestAttribs()

Получение атрибутов теста (по индексу теста в массиве).

Входные параметры:

Параметр	Тип	Описание
TestIndex	integer	индекс теста в массиве тестов (0 <= TestIndex<TestCount)

Выходные параметры:

Параметр	Тип	Описание
TestAttribs	TypeTestAttribs	структура с атрибутами теста
OperationStatus	TypeOperationStatus	результат выполнения операции

Внимание! Для оптимизации работы со списком тестов рекомендуется использовать метод getTestsAttribsList(), который в результате выполнения одного запроса выдает полный список тестов со всеми атрибутами.

4.1.4. Метод getTestScalesList()

Получение списка измерительных шкал теста (по имени теста).

Входные параметры:

Параметр	Тип	Описание
TestName	string	системное имя теста
ReportVariantId	integer	идентификатор варианта отчета (0 –получить все шкалы)

Выходные параметры:

Параметр	Тип	Описание
TestsScalesList	TypeTestsScalesList	список атрибутов для шкал теста
OperationStatus	TypeOperationStatus	результат выполнения операции

4.1.5. Метод getTestLicensesCount()

Получение числа доступных лицензий для теста (по имени теста).

Входные параметры:

Параметр	Тип	Описание
TestName	String	системное имя теста

Выходные параметры:

Параметр	Тип	Описание
LicensesCount	integer	число доступных лицензий для теста
OperationStatus	TypeOperationStatus	результат выполнения операции

Внимание! Для оптимизации работы со списком тестов рекомендуется использовать метод getTestsAttribsList(), который в результате выполнения одного запроса выдает полный список тестов со всеми атрибутами.

4.1.6. Метод getTestsAttribsList()[ОСНОВНОЙ]

Получение списка атрибутов для всех тестов.

Входные параметры:отсутствуют

Выходные параметры:

Параметр	Тип	Описание
TestsAttribsList	TypeTestsAttribsList	список атрибутов доступных тестов
OperationStatus	TypeOperationStatus	результат выполнения операции

4.2. Методы для управления сеансами тестирования

4.2.1. Метод createTestingSession()[ОСНОВНОЙ]

Создание сеанса тестирования с заданным идентификатором сеанса.

Входные параметры:

Параметр	Тип	Описание
TestName	string	системное имя теста
ExternalSessionGuid	string	уникальный идентификатор сеанса тестирования (генерируется на стороне клиента)
IsAnonymousSession	Integer	параметр запроса личных данных в начале тестирования (0 – не запрашивать, 1 - запрашивать)

Выходные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	String	уникальный идентификатор созданного сеанса тестирования
OperationStatus	TypeOperationStatus	результат выполнения операции

4.2.2. Метод getTestingSessionUrl()[основной]

Получение адреса страницы для сеанса тестирования.

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования

Выходные параметры:

Параметр	Тип	Описание
TestingSessionUrl	string	адрес страницы сеанса тестирования
OperationStatus	TypeOperationStatus	результат выполнения операции

4.2.3. Метод getTestingSessionUrlEx()[основной]

Получение адреса страницы для сеанса тестирования (с учетом дополнительных параметров).

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования
IsCompactMode	integer	параметр компактного режима тестирования (0 - нет, 1 - да)
FrameWidth	integer	ширина фрейма тестирования в пикселях (0 - по умолчанию)
FrameHeight	integer	высота фрейма тестирования в пикселях (0 - по умолчанию)

Выходные параметры:

Параметр	Тип	Описание
TestingSessionUrl	string	адрес страницы сеанса тестирования
OperationStatus	TypeOperationStatus	результат выполнения операции

4.2.4. Метод isTestingSessionComplete()[основной]

Проверка состояния завершенности сеанса тестирования*.

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования

Выходные параметры:

Параметр	Тип	Описание
IsSessionComplete	boolean	признак завершенности сеанса тестирования
OperationStatus	TypeOperationStatus	результат выполнения операции

4.2.4.1. Механизм Callback

Существует более эффективная альтернатива периодическому использованию метода isTestingSessionComplete().

По умолчанию для контроля текущего состояния сеанса тестирования нужно периодически вызывать метод API isTestingSessionComplete(), который возвращает текущий статус сеанса тестирования - "завершен" или "не завершен". Однако, при большом числе одновременных сеансов тестирования данный подход может повлечь за собой неоправданную избыточную нагрузку, как на сервер, так и на клиента API.

Для того чтобы оптимизировать решение задачи уведомления клиента API о завершении сеанса тестирования, на сервере API реализован механизм колбека - обратного вызова, который передается клиентской системе по окончании сеанса тестирования.

Механизм колбека позволяет настроить автоматический обратный вызов метода в информационной системе клиента, который будет вызываться каждый раз при завершении сеанса тестирования. В качестве такого обратного вызова может быть использован, например, HTTP-запрос (GET или POST) указанного клиентом URI или SOAP-запрос к серверу клиента.

Настройка обратного вызова под клиента требует участия разработчиков системы HT-Line, поэтому нет жестких ограничений на тип и состав обратного вызова - могут быть использованы как упомянутые выше вызовы HTTP и SOAP, так какие-то более специфические варианты (это зависит от особенностей информационной системы клиента, которая интегрируется с системой HT-Line).

В качестве простейшего решения для колбека мы предлагаем по умолчанию использовать HTTP-запрос, который отправляет в систему клиента (методом GET или POST) идентификатор завершенного сеанса тестирования - ExternalSessionGuid. Для защиты передаваемых данных можно подписывать сообщение, используя дополнительный параметр, содержащий дайджест отправляемых данных (например, "хеш с солью" от ExternalSessionGuid).

Таким образом, для того чтобы при интеграции с системой HT-Line можно было задействовать механизм колбека, разработчики клиента API должны обратиться к разработчикам HT-Line и передать информацию о методе обратного вызова (HTTP, SOAP и т.д.), адресе (URI) и списке передаваемых параметров (по умолчанию - ExternalSessionGuid и подпись). В ответ на такое обращение разработчики системы HT-Line реализуют и настроят соответствующий обратный вызов в системе HT-Line.

4.2.5. Метод isTestingSessionInterrupted()

Проверка досрочного завершения сеанса тестирования.

Внимание! Используется только для тестов с пороговым фильтром. Метод был разработан для специальной версии теста и не является основным.

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования

Выходные параметры:

Параметр	Тип	Описание
isSessionInterrupted	boolean	Признак досрочного завершения сеанса тестирования: *true* - сеанс тестирования является завершенным, но сработал пороговый фильтр; *false* - сеанс тестирования является завершенным, фильтр (если он установлен) - не срабатывал.
OperationStatus	TypeOperationStatus	результат выполнения операции

4.3. Методы доступа к результатам тестирования

4.3.1. Метод getResultsReportUrl()[основной]

Получение адреса страницы отчета по результатам тестирования.

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования

Выходные параметры:

Параметр	Тип	Описание
ResultsReportUrl	string	адрес страницы отчета с результатами тестирования
OperationStatus	TypeOperationStatus	результат выполнения операции

Примечание. Метод отдает адрес страницы для варианта отчета по умолчанию - как правило, в качестве такого варианта выступает отчет для респондента.

Внимание! По прохождении тестирования в системе HT-Line формируется отчет. Отчет может быть в нескольких вариантах. Например «Отчет для респондента» и «Отчет для руководителя». Контент отчетов разных вариантов отличается. Для каждого вариант отчета есть свой идентификатор ReportVariantId.

4.3.2. Метод getResultsReportXml()[ОСНОВНОЙ]

Получение структурированного контента отчета в формате XML.

Внимание! В случае, когда отчет генерируется средствами ИС заказчика, то метод getResultsReportXML() является самым удобным в использовании. Позволяет одним вызовом получить контент всех вариантов отчетов в удобном для парсинга XML формате.

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования
ReportVariantId	integer	идентификатор варианта отчета (0 – получить все варианты отчетов)
Base64Encoded	integer	признак кодирования контента в кодировке Base64 (0 - нет, 1 - да)

Выходные параметры:

Параметр	Тип	Описание
ReportContentXml	string	структурированный контент отчета в формате XML
OperationStatus	TypeOperationStatus	результат выполнения операции

4.3.3. Метод getResultsReportHtml()[основной]

Получение контента отчета в формате HTML.

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования
ReportVariantId	integer	идентификатор варианта отчета (0 - вариант по умолчанию)
Base64Encoded	integer	признак кодирования контента в кодировке Base64 (0 - нет, 1 - да)

Выходные параметры:

Параметр	Тип	Описание
ReportContentHtml	string	контент отчета в формате HTML
OperationStatus	TypeOperationStatus	результат выполнения операции

4.3.4. Метод getResultsReportHtmlEx()

Получение контента отчета в формате HTML (с подстановкой личных данных).

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования
ReportVariantId	integer	идентификатор варианта отчета (0 - вариант по умолчанию)
TestingDateTime	string	дата тестирования
ResponderFullName	string	полное имя респондента
ResponderGender	string	пол респондента
ResponderBirthDate	string	дата рождения респондента
ResponderAge	string	возраст респондента
Base64Encoded	integer	признак кодирования контента в кодировке Base64 (0 - нет, 1 - да)

Выходные параметры:

Параметр	Тип	Описание
ReportContentHtml	string	контент отчета в формате HTML
OperationStatus	TypeOperationStatus	результат выполнения операции

4.3.5. Метод getResultsReportPdf()

Получение контента отчета в формате PDF.

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования
ReportVariantId	integer	идентификатор варианта отчета (0 - вариант по умолчанию)
Base64Encoded	integer	признак кодирования контента в кодировке Base64 (0 - нет, 1 - да)*

*Метод работает только со значением параметра Base64Encoded = 1.

Выходные параметры:

Параметр	Тип	Описание
ReportContentPdf	string	контент отчета в формате PDF
OperationStatus	TypeOperationStatus	результат выполнения операции

4.3.6. Метод getResultsReportPdfEx()

Получение контента отчета в формате PDF (с подстановкой личных данных).

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования
ReportVariantId	integer	идентификатор варианта отчета (0 - вариант по умолчанию)
TestingDateTime	string	дата тестирования
ResponderFullName	string	полное имя респондента
ResponderGender	string	пол респондента
ResponderBirthDate	string	дата рождения респондента
ResponderAge	string	возраст респондента
Base64Encoded	integer	признак кодирования контента в кодировке Base64 (0 - нет, 1 - да)*

*Метод работает только со значением параметра Base64Encoded = 1.

Выходные параметры:

Параметр	Тип	Описание
ReportContentPdf	string	контент отчета в формате PDF
OperationStatus	TypeOperationStatus	результат выполнения операции

4.3.7. Метод getResultsReportDocx()

Получение контента отчета в формате DOCX.

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования
ReportVariantId	integer	идентификатор варианта отчета (0 - вариант по умолчанию)
Base64Encoded	integer	признак кодирования контента в кодировке Base64 (0 - нет, 1 - да)*

*Метод работает только со значением параметра Base64Encoded = 1.

Выходные параметры:

Параметр	Тип	Описание
ReportContentDocx	string	контент отчета в формате DOCX
OperationStatus	TypeOperationStatus	результат выполнения операции

4.3.8. Метод getResultsReportDocxEx()

Получение контента отчета в формате DOCX (с подстановкой личных данных).

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования
ReportVariantId	integer	идентификатор варианта отчета (0 - вариант по умолчанию)
TestingDateTime	string	дата тестирования
ResponderFullName	string	полное имя респондента
ResponderGender	string	пол респондента
ResponderBirthDate	string	дата рождения респондента
ResponderAge	string	возраст респондента
Base64Encoded	integer	признак кодирования контента в кодировке Base64 (0 - нет, 1 - да)*

*Метод работает только со значением параметра Base64Encoded = 1.

Выходные параметры:

Параметр	Тип	Описание
ReportContentDocx	string	контент отчета в формате DOCX
OperationStatus	TypeOperationStatus	результат выполнения операции

4.3.9. Метод getResultsScales()[УСТАРЕВШИЙ]

Получение результатов тестирования в виде значений по измерительным шкалам теста.

Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования
ReportVariantId	integer	идентификатор варианта отчета (0 - вариант по умолчанию)

Выходные параметры:

Параметр	Тип	Описание
ScalesValues	string	строка результатов по шкалам теста (разделитель - символ ";")
OperationStatus	TypeOperationStatus	результат выполнения операции

Метод getResultsScales() исключен в версии API 1.6.0

4.3.10. Метод getReportVariantsList()

Получение списка вариантов отчета для теста.

Входные параметры:

Параметр	Тип	Описание
TestName	string	системное имя теста

Выходные параметры:

Параметр	Тип	Описание
ReportVariantsList	TypeReportVariantsList	список атрибутов вариантов отчета для теста
OperationStatus	TypeOperationStatus	результат выполнения операции

4.3.11. Метод getResultsValuesList()[основной]

Запрос списка значений результатов тестирования.
Входные параметры:

Параметр	Тип	Описание
ExternalSessionGuid	string	идентификатор созданного ранее сеанса тестирования
ReportVariantId	integer	идентификатор варианта отчета (0 – получить список значений по всем шкалам)
ValuesMetrics	string	метрикадляScaleValue ("sten", "source", "percent", "hit", "iq", "custom"; поумолчанию "sten"

Выходные параметры:

Параметр	Тип	Описание
ResultsValuesList	TypeResultsValuesList	список значений результатов тестирования по шкалам теста
OperationStatus	TypeOperationStatus	результат выполнения операции

4.3.12. Метод getResultsReportByProtocolIdXml()

Получение структурированного контента отчета в формате XML по идентификатору протокола. Метод позволяют получить информацию о любом протоколе тестирования респондентов текущего пользователя, независимо от того, присвоен ли сеансу ExternalSessionGuid илинет.

Входные параметры:

Параметр	Тип	Описание
ProtocolId	integer	числовой идентификатор протокола тестирования
ReportVariantId	integer	идентификатор варианта отчета (0 - вариант по умолчанию)
Base64Encoded	integer	признак кодирования контента в кодировке Base64 (0 - нет, 1 - да)

Выходные параметры:

Параметр	Тип	Описание
ReportContentXml	string	структурированный контент отчета в формате XML
OperationStatus	TypeOperationStatus	результат выполнения операции

4.4. Методы для работы с настройками тестовых методик

4.4.1. Метод getTestSettingsInterrupts()

Получение настроек прерывания, установленных для тестовой методики.

Входные параметры:

Параметр	Тип	Описание
TestName	string	системное имя теста

Выходные параметры:

Параметр	Тип	Описание
IsInterruptEnabled	boolean_Тип_данных_TypeReportVariantsList	признак установки режима прерывания для тестовой методики
InterruptString	string	строка установленных настроек прерывания тестовой методики. Разделители **:** и **;**
OperationStatus	TypeOperationStatus	результат выполнения операции

4.5. Сервисные и дополнительные методы

4.5.1. Метод getApiVersionNumber()

Получение номера текущей версии интерфейса.

Входные параметры: отсутствуют

Выходные параметры:

Параметр	Тип	Описание
VersionNumber	string	номер текущей версии интерфейса
OperationStatus	TypeOperationStatus	результат выполнения операции

4.6. Методы для работы с протоколами тестирования

4.6.1. Метод getResponderProtocolsList()[УСТАРЕВШИЙ]

Получение списка протоколов по данным анкеты респондента.Метод позволяют получить информацию о любом протоколе тестирования респондентов текущего пользователя, независимо от того, присвоен ли сеансу ExternalSessionGuid или нет.

Входные параметры:

Параметр	**Тип**	**Описание**
FieldNamesArray	string	Массив полей анкеты в формате строки "xN;xN;...;xN;", гдехNизсписка (FirstName, MiddleName, LastName, Name, Age, BirthDate, Gender, Email, Phone, Field1, Field2, Field3, Field4, Field5)
FieldValuesArray	string	Массив значений полей анкеты в формате строки "xN;xN;...;xN;"
PartialMatching	int	Полнотекстовый поиск. 1 - включен, 0 - выключен. По умолчанию - 0.

Параметр PartialMatching не влияет на поле анкеты Age.

Значение поляанкеты BirthDate передается в формате ГГГГ-ММ-ДД. При PartialMatching = 1 допустимо передавать в качестве значения год рождения в формате ГГГГ.

В качестве значения поля анкеты Namепередается краткое имя респондента.

Выходные параметры:

Параметр	Тип	Описание
ResponderProtocolsList	TypeResponderProtocolsList	Двумерныймассив, элементамикоторогоявляются (ProtocolId, TestName, TestTitle, FirstName, MiddleName, LastName, Name, Age, BirthDate, Gender, Email, Phone, Field1, Field2, Field3, Field4, Field5)
OperationStatus	TypeOperationStatus	результат выполнения операции

Метод getResponderProtocolsList () исключен в версии API 1.6.0

5. СЛОЖНЫЕ ТИПЫ ДАННЫХ

В методах интерфейса используются как простые, так и сложные (комплексные) типы данных. Как правило, в качестве аргументов методов используются простые типы данных (строки и целые числа), а в качестве возвращаемых результатов – сложные (структуры и массивы).

Например, результат выполнения любого метода интерфейса содержит структуру данных OperationStatus сложного типа TypeOperationStatus, которая содержит такие поля, как числовой код результата выполнения метода (номер ошибки), мнемонический код результата (мнемокод ошибки) и текстовое описание результата (текст сообщения об ошибке). Значение полей этой структуры можно использовать для протоколирования номера ошибки и для вывода пользователю системы текста сообщения о возникшей ошибке.

5.1. Тип данных TypeOperationStatus

Этот тип данных предназначен для описания результата выполнения метода интерфейса. В случае, если произошла ошибка при выполнении метода, в структуре данных этого типа (OperationStatus) содержится числовое и текстовое описание ошибки.

Описаниетипа:

```
      <xsd:complexType name="TypeOperationStatus">
      <xsd:sequence>
      <xsd:elementminOccurs="1" maxOccurs="1" name="ErrorNumber" type="xsd:int"/>
      <xsd:elementminOccurs="1" maxOccurs="1" name="ErrorCode" type="xsd:string"/>
      <xsd:elementminOccurs="1" maxOccurs="1" name="ErrorMessage"
type="xsd:string"/>
      </xsd:sequence>
      </xsd:complexType>
```

Назначение полей:

- ErrorNumber - числовой код выполнения операции (номер ошибки);
- ErrorCode - мнемонический код выполнения операции (мнемокод ошибки);
- ErrorMessage - текстовое описание ошибки.

В случае отсутствия ошибок при выполнении метода номер ошибки равен нулю (ErrorNumber=0), а мнемокод ошибки содержит строку "OK" (ErrorCode="OK").

5.2. Тип данных TypeTestAttribs

Этот тип данных предназначен для описания атрибутов одного из тестов из каталога тестов. В полях этой структуры описаны все атрибуты теста, которые могут потребоваться при работе с интерфейсом: идентификатор теста, название теста, статус теста, число доступных лицензий и число собранных ранее протоколов тестирования.

Описаниетипа:

```
<xsd:complexType name="TypeTestAttribs">
<xsd:sequence>
<xsd:element name="TestId" type="xsd:int"/>
<xsd:element name="TestUid" type="xsd:string"/>
<xsd:element name="TestName" type="xsd:string"/>
<xsd:element name="MaintestName" type="xsd:string"/>
<xsd:element name="TestTitle" type="xsd:string"/>
<xsd:element name="TestEnabled" type="xsd:boolean"/>
<xsd:element name="IsLibraryTest" type="xsd:boolean"/>
<xsd:element name="QuestionsCount" type="xsd:int"/>
<xsd:element name="LicensesCount" type="xsd:int"/>
<xsd:element name="ProtocolsCount" type="xsd:int"/>
</xsd:sequence>
</xsd:complexType>
```

Назначение полей:

- TestId - уникальный числовой идентификатор экземпляра теста;
- TestUid - уникальный строковый идентификатор экземпляра теста;
- TestName - системное имя теста;
- TestTitle - название теста;
- TestEnabled - статус теста (0 - отключен, 1 - включен);
- IsLibraryTest - принадлежит ли тест сервису Maintest-5i (0 – тест не является одним из тестов Maintest-5i, 1 - является);
- MaintestName - системное имя теста в формате Maintest-4;
- QuestionsCount – число вопросов в тесте;
- LicensesCount - число доступных для использования лицензий;
- ProtocolsCount - число накопленных протоколов тестирования.

5.3. Тип данных TypeTestsAttribsList

Этот тип предназначен для описания атрибутов сразу всех доступных для пользователя тестов из каталога тестов. Он содержит массив структур типа TypeTestAttribs, описанных выше.

Описание типа:

```
    <xsd:complexType name="TypeTestsAttribsList">
    <xsd:sequence>
    <xsd:element name="TestAttribs" type="tns:TypeTestAttribs"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    </xsd:complexType>
```

Назначение полей:

- TestAttribs - атрибуты теста в виде структуры типа TypeTestAttribs.

5.4. Тип данных TypeTestScale

Этот тип данных предназначен для описания одной из измерительных шкал теста. В полях этой структуры содержится название измерительной шкалы, а также такие ее атрибуты, как тип шкалы, число полюсов и порядковый номер шкалы в тесте.

Описаниетипа:

```
<xsd:complexType name="TypeTestScale">
<xsd:sequence>
<xsd:element name="ScaleNumber" type="xsd:int"/>
<xsd:element name="ScaleTitle" type="xsd:string"/>
<xsd:element name="ScaleType" type="xsd:string"/>
<xsd:element name="ScalePoles" type="xsd:int"/>
</xsd:sequence>
</xsd:complexType>
```

Назначение полей:

- ScaleNumber - порядковый номер шкалы в тесте;
- ScaleTitle - наименование измерительной шкалы;
- ScaleType - мнемокод типа измерительной шкалы (внутренне понятие системы HT-Line);
- ScalePoles - число полюсов шкалы (1 или 2).

5.5. Тип данных TypeTestsScalesList

Этот тип данных предназначен для описания полного списка измерительных шкал теста. Он содержит массив структур типа TypeTestScale, описанных выше.

Описание типа:

```
    <xsd:complexType name="TypeTestsScalesList">
    <xsd:sequence>
    <xsd:element name="TestScale" type="tns:TypeTestScale"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    </xsd:complexType>
```

Назначение полей:

- TestScale - описания шкалы теста в виде структуры типа TypeTestScale.

5.6. Тип данных TypeReportVariant

Этот тип предназначен для описания атрибутов варианта отчета теста.

Описаниетипа:

```
<xsd:complexType name="TypeReportVariant">
<xsd:sequence>
<xsd:element name="VariantId" type="xsd:int"/>
<xsd:element name="VariantName" type="xsd:string"/>
<xsd:element name="VariantTitle" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
```

Назначение полей:

- VariantId - номер варианта отчета;
- VariantName - системное имя варианта отчета;
- VariantTitle - название варианта отчета.

5.7. Тип данных TypeReportVariantsList

Этот тип данных предназначен для описания полного списка вариантов отчета теста. Он содержит массив структур типа TypeReportVariant, описанных выше.

Описание типа:

```
    <xsd:complexType name="TypeReportVariantsList">
    <xsd:sequence>
    <xsd:element name="ReportVariant" type="tns:TypeReportVariant"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    </xsd:complexType>
```

Назначение полей:

- ReportVariant - описание варианта в виде структуры типа TypeReportVariant.

5.8. Тип данных TypeResponderProtocol

Этот тип содержит данные о протоколе и респонденте: идентификатор протокола, имя и название теста, данные анкеты респондента.

Описаниетипа:

```
<xsd:complexType name="TypeResponderProtocol">
<xsd:sequence>
<xsd:element name="ProtocolId" type="xsd:string"/>
<xsd:element name="TestName" type="xsd:string"/>
<xsd:element name="TestTitle" type="xsd:string"/>
<xsd:element name="FirstName" type="xsd:string"/>
<xsd:element name="MiddleName" type="xsd:string"/>
<xsd:element name="LastName" type="xsd:string"/>
<xsd:element name="Name" type="xsd:string"/>
<xsd:element name="Age" type="xsd:string"/>
<xsd:element name="BirthDate" type="xsd:string"/>
<xsd:element name="Gender" type="xsd:string"/>
```

```
<xsd:element name="Email" type="xsd:string"/>
<xsd:element name="Phone" type="xsd:string"/>
<xsd:element name="Field1" type="xsd:string"/>
<xsd:element name="Field2" type="xsd:string"/>
<xsd:element name="Field3" type="xsd:string"/>
<xsd:element name="Field4" type="xsd:string"/>
<xsd:element name="Field5" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
```

Назначение полей:

- ProtocolId – уникальный числовой идентификатор протокола.
- TestName – идентификатор теста.
- TestTitle – название теста
- FirstName – имя респондента.
- MiddleName – отчество респондента.
- LastName – фамилия респондента.
- Name – краткое имя респондента.
- Age – возраст респондента
- BirthDate – дата рождения респондента в формате ГГГГ-ММ-ДД
- Gender – пол респондента: m – мужской, f – женский.
- Email – emailреспондента
- Phone – телефон респондента
- Field1…Field5 – настраиваемые поля анкеты респондента

5.9. Тип данных TypeResponderProtocolsList

Этот тип данных предназначен для описания полного списка данных о протоколе и респонденте. Он содержит массив структур типа TypeResponderProtocol,описанных выше.

Описание типа:

```
<xsd:complexType name="TypeResponderProtocolsList">
<xsd:sequence>
<xsd:element name="ResponderProtocolsList" type="tns:TypeResponderProtocol"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
```

Назначениеполей:

- ResponderProtocolsList – описание протокола и респондента в виде структуры типа TypeResponderProtocol.

5.10. Тип данных TypeResultsScaleValues

Этот тип содержит данные о значениях результатов тестирования по шкале.

Описание типа:

```
<xsd:complexType name="TypeResultsScaleValues">
	<xsd:sequence>
		<xsd:element name="ScaleNumber" type="xsd:int"/>
		<xsd:element name="ScaleTitle" type="xsd:string"/>
		<xsd:element name="ScaleValue" type="xsd:double"/>
		<xsd:element name="StenValue" type="xsd:double"/>
		<xsd:element name="SourceValue" type="xsd:int"/>
		<xsd:element name="PercentValue" type="xsd:int"/>
		<xsd:element name="HitValue" type="xsd:int"/>
		<xsd:element name="IqValue" type="xsd:int"/>
		<xsd:element name="CustomValue" type="xsd:double"/>
	</xsd:sequence>
</xsd:complexType>
```

Назначение полей:

- ScaleNumber – номер шкалы;
- ScaleTitle –наименование шкалы;
- ScaleValue – значение по шкале;
- StenValue – результат по шкале в стенах;
- SourceValue – результат по шкале в сырых баллах;
- PercentValue – результат по шкале в процентах;
- HitValue – результат по шкале в виде числа ключевых ответов;
- IqValue – результат по шкале в iq – баллах.
- CustomValue – значение по настраиваемой шкале.

5.11. Тип данных TypeResultsValuesList

Этот тип данных предназначен для описания полного списка данных о численных значениях результата тестирования по шкале. Он содержит массив структур типа TypeResultsScaleValues, описанных выше.

Описание типа:

```
  <xsd:complexType name="TypeResultsValuesList">
	<xsd:sequence>
		<xsd:element name="ScaleValues" type="tns:TypeResultsScaleValues" minOccurs="0" maxOccurs="unbounded"/>
	</xsd:sequence>
  </xsd:complexType>
```

Назначениеполей:

- ScaleValues – описание значений результатов тестирования по шкале в виде структуры типа TypeResultsScaleValues.

6. СЛОВАРЬ ТЕРМИНОВ И ОБОЗНАЧЕНИЙ