

Документация, содержащая описание функциональных характеристик программного обеспечения и информацию, необходимую для установки и эксплуатации программного обеспечения

«Модуль универсального селекторного ядра SEL-CC»

ТЕРМИНЫ И ОБОЗНАЧЕНИЯ

Термин, сокращение	Определение
API ПО	Программный интерфейс взаимодействия ПО с внешними системами для приема-передачи данных.
Опрос (игра)	Совокупность сведений об источнике данных, параметрах принятия решения, фильтрах, системе сбора результатов обработки. Имеет уникальный идентификатор.
ПО	Программное обеспечение
Финальный ответ	Результат обработки вопроса из опроса.
База данных	Единая база данных ПО.
ИС	Информационная система

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Цели разработки ПО

ПО является инструментом создания опросов и обработки массовых пользовательских ответов посредством API ПО.

Ключевой целью разработки ПО является создание единообразного инструмента проведения опросов с заложенными в нём алгоритмами автоматического вынесения вердикта по опросу.

1.2 Идея разработки ПО

В основе ПО лежит идея о том, что опросы и тесты пользователей в задачах без заведомого известного правильного ответа, можно автоматизировать. Автоматизация достигается за счет введения правил кворумирования на большой выборке данных.

Например, порогом необходимой выборки по умолчанию стоит 7 из 10 ответов. В данном случае, с высокой долей вероятности можно предположить, что некоррелируемые пользователи, вошедшие в «семёрку» одинаковых ответов, дали верный ответ – система в этом случае фиксирует данный ответ как финальный.

ПО включает в себя следующие логические узлы:

- Главный модуль принятия решения
- Фильтры фотоизображений
- API библиотека

Создание опросов производится посредством API библиотеки. Главный модуль сохраняет конфигурацию, которая была ему передана (тип источника данных, минимальное и максимальное количество пользователей на один вопрос в рамках опроса, пороговые значения, потребность в фильтрах, пункт назначения результатов опроса, периоды проведения опроса). После получения пачки данных, Главный модуль передает данные в сторону Фильтров изображений (сокрытие личной информации, водяных знаков, признаков поломки и некачественного изображения). Как только наступает время начала проведения опроса, главный модуль формирует очередь вопросов и становится доступен для обращения извне посредством API. При обращении внешней системы, Главный модуль передает вопрос и ожидает на него ответ. После окончания опросов, главный модуль формирует результаты и передает их в систему-потребитель посредством API.

1.3 Функции и сервисы ПО

К функциям и сервисам ПО относятся:

1. Формирование опросов (игр);
2. Загрузка источников данных для опроса (фото данных) в ПО;
3. Выбор настроек игры;
4. Взаимодействие с внешними системами по API ПО;
5. Формирование результатов задания и информирование систем-потребителей результатов опроса.

2 ОПИСАНИЕ ФУНКЦИОНАЛА ПО

2.1 Описание подсистем ПО

2.1.1 Описание подсистемы Главный модуль принятия решения

Подсистема Главный модуль принятия решения обладает следующим функционалом:

- является аккумулярующим звеном, обеспечивающим взаимодействие внутренних подсистем ПО с внешними и внутренними источниками информации, фильтрами, потребителями результатов опроса;
- предполагает работу с большим объемом данных и может быть размещена на распределенном узле серверов, в связи с чем имеет собственный сетевой балансировщик в единой точке входа запросов;
- финальный результат вопросов в рамках опроса передается в API библиотеку.

2.1.2 Описание подсистемы Фильтров фотоизображений

Подсистема Фильтров фотоизображений обладает следующим функционалом:

- фильтрация изображения в части исключения водяных знаков (оверлеев) изображения;
- фильтрация изображения в части исключения фотоматериалов с явным присутствием фиксируемого лица;
- фильтрация изображения в части исключения фотоматериалов с явным присутствием различных силуэтов людей;
- фильтрация изображения в части исключения фотоматериалов с явным присутствием различных государственных регистрационных знаков транспортных средств;
- фильтрация изображения в части исключения фотоматериалов с явным присутствием повреждения, расфокусировки изображения с непригодной для восприятия информационной составляющей.

2.1.3 Описание подсистемы API библиотека

Подсистема API библиотека обладает следующим функционалом:

- возможность программного создания опросов
- возможность программного получения статусов опросов
- возможность получения вопроса для принятия решения пользователем
- возможность отправки ответа на вопрос пользователем
- возможность программного получения финальных ответов

Подсистема является точкой входа и выхода в процессе работы пользователя с ПО.

3 РАБОТА С ПО

3.1 Создание опроса (игры)

Пример запроса:

```
POST http://{{host}}:{{port}}/api/v1/game
```

Тело запроса:

```
{
  "extSystemId": "test", // внешняя система
  "name": "game_finished", //имя игры
  "answerType": 2, //тип ответа
  "gameType": 1, //тип игры
  "startDate": "1680001600", //дата старта
  "finishDate": "1682758800", //дата завершения
  "activityPeriod": "17:00-23:59", //период активности
  "question": "Контейнер (бункер-накопитель) размещен вне специально
оборудованной площадки?", //формулировка задания
  "options": "нет нарушения,есть нарушение,нет ответа", //опции для
категориального типа ответа (answerType=2)
  "filterToggles": {
    "cropProcessingOn": true, //кроп фильтр
    "faceFilterOn": true, //фильтр лиц
    "silhouetteFilterOn": false, //фильтр силуэтов
    "spkvFilterOn": true //СПКВ
  }, //фильтры
  "config": {
    "actualImageAmount": 100, //порог актуальных скриншотов в игре
    "screenshotSupplyLimit": 20, //кол-во выдач одного актуального
скриншота пользователям
    "fakeImageProportion": 0.01, //флаг включения подмешивания (с
пропорцией кол-ва фейков относительно кол-ва актуальных)
    "fakeSupplyLimit": 1000, //кол-во выдач одного подмешанного скриншота
пользователям
    "lessThanActiveUsersPositiveFakeProbability": 0.8, //коэффициент
выдачи подмешанного скриншота с нарушением при недостижении порога активных
пользователей
    "moreThanActiveUsersPositiveFakeProbability": 1, //коэффициент выдачи
подмешанного скриншота с нарушением по достижению порога активных
пользователей
    "activeUsersThreshold": 0, //порог активных пользователей
    "answersRequiredToComputeFinal": 10, //кол-во ответов, необходимых
для формирования ФО
    "definedAnswerThreshold": 7, //кол-во однозначных ответов,
необходимых для принятия решения
    "silhouetteVboxThresholdPercentage": 2, //максимальный процент ббокса
с силуэтом от площади изображения
    "useFinalAnswerAsDecisive": false, //использование ФО в качестве ИО
(по умолчанию false -> ИО = ЭО)
    "setFinalAnswerAfterGameActivityFlag": false //флаг актуальности
скриншотов в течение одного дня (по ум. false -> скриншоты актуальны всегда)
  } //конфигурация
}
```

Пример ответа:

```
{
  "code": 200,
  "success": true,
  "data": {
    "gameId": "someId"
  }
}
```

```
}  
}
```

3.2 Получение информации об опросах

GET

/api/v1/game?extSystemId=someid&sourceType=schedule&from=1624324242&to=1624324242&finished=true

Примечание:

В случае отсутствия в запросе "finished" – данными ответа является список всех игр в данном временном диапазоне.

Параметр	Описание	Тип данных	Обязательный	Значение по умолчанию
extSystemId	Идентификатор внешней системы	String	Да	NULL
From	Фильтр по времени получения результатов анализа (от) – unix timestamp	Integer	Нет	День и время первой игры
To	Фильтр по времени получения результатов анализа (до) – unix timestamp	Integer	Нет	Текущий день
Finished	Фильтр по статусу игры: завершена или нет (true false)	String	Нет	true false
sourceType	Тип источника данных: schedule file url	String	Нет	Все источники
answerType	Фильтр по типу ответа	String	Нет	Все типы

Ответ:

```
{  
  "success": true,  
  "data": {  
    "games": [  
      {  
        "gameId": "f004fce2-8593-453f-a09b-98303e526edd",  
        "name": "SomeName",  
        "gameType": 1 | 2,  
        "sourceType": "schedule",  
        "status": "finished" | "inProcess" | "notStarted",  
        "startDate": "1604582153",  
        "finishDate": "1604582155",  
        "activityPeriod": {  
          "from": "14:00",  
          "to": "17:00"  
        },  
      },  
      "screenshotsCount": 123  
    ],  
  },  
}
```

```
    { ... }  
  ]  
}  
}
```

Поля ответа:

Поле	Описание	Тип данных
Success	Индикатор успешности запроса	Boolean
data.games[].gameId	Идентификатор игры	String
data.games[].name	Наименование игры	String
data.games[].gameType	Тип игры	Integer
data.games[].sourceType	Тип источника данных: schedule file url	String
data.games[].status	Статус игры	String
data.games[].startDate	Время начала игры в Unix timestamp	String
data.games[].finishDate	Время окончания игры в Unix timestamp	String
data.games[].activityPeriod.from	Начало периода активности (hh:mm)	String
data.games[].activityPeriod.to	Конец периода активности (hh:mm)	String
data.games[].screenshotsCount	Количество задач (скриншотов)	Integer

3.3 Получение новой задачи для пользователя

GET /api/v1/game/{gameId}/screenshot?extSystemId=someid&userId=someuser

Параметр	Описание	Тип данных	Обязательный	Значение по умолчанию
extSystemId	Идентификатор внешней системы	String	да	NULL
gameId	Идентификатор игры	String	да	NULL
userId	Идентификатор пользователя внешней системы	String	да	NULL

Ответ:

```
{
  "success": true,
  "screenshot": {
    "screenshotId": "08769582-6bb9-4385-a8a2-4606500b7ca0",
    "imageUrl": "https://address:port/image.jpg"
  }
}
```

Поля ответа:

Поле	Описание	Тип данных
success	Индикатор успешности запроса	Boolean
screenshot.screenshotId	Идентификатор задачи	String
screenshot.imageUrl	Ссылка на изображение	String

3.4 Отправка ответа пользователя

POST /api/v1/game/{gameId}/screenshot/{screenshotId}/answer

Параметр	Описание	Тип данных	Обязательный	Значение по умолчанию
screenshotId	Идентификатор задачи	String	Да	NULL
gameId	Идентификатор игры	String	Да	NULL

Данные запроса:

```
{
  "extSystemId": "f00866ba-4d82-11ea-b24d-0242ac130003",
  "userId": "someuser",
  "answer": ...
}
```


Поля запроса:

Поле	Описание	Тип данных
extSystemId	Идентификатор внешней системы	String
userId	Идентификатор пользователя внешней системы	String
answer	Зависит от типа данных, определяемого из screenshotId -> gameId	Answer

Ответ:

```
{
  "success": true,
  "data": {
    "finished": true | false,
    "userResult": "right" | "wrong" | "undefined" | "inProcess",
    "answers": [
      {
        "userId": "someid",
        "answer": ...,
        "result": "right" | "wrong" | "undefined" | "inProcess"
      },
      { ... }
    ]
  }
}
```

Поля ответа:

Поле	Описание	Тип данных
Success	Индикатор успешности запроса	Boolean
data.finished	Флаг получения финального ответа	Boolean
data.userResult	Результат ответа пользователя, по которому сделан запрос: "right": верный ответ "wrong": неверный ответ "undefined": финальный ответ неоднозначен "inProcess": финальная ответ не получена	String
data.answers[].userId	Идентификатор одного из ответивших пользователей	String
data.answers[].answer	Зависит от типа данных, определяемого из screenshotId -> gameId	Answer

<code>data.answers[].result</code>	Результат ответа пользователя из списка: “right”: верный ответ “wrong”: неверный ответ “undefined”: финальный ответ неоднозначен “inProcess”: финальный ответ не получен	String
------------------------------------	--	--------