

Приложение № 5  
к требованиям к вычислительной мощности,  
используемой провайдером хостинга, для  
проведения уполномоченными  
государственными органами, осуществляющими  
оперативно-разыскную деятельность или  
обеспечение безопасности Российской  
Федерации, в случаях, установленных  
федеральными законами, мероприятий в целях  
реализации возложенных на них задач,  
утвержденным приказом Министерства  
цифрового развития, связи и массовых  
коммуникаций Российской Федерации  
от 1 ноября 2023 г. № 935

**Перечень сервисных типов и соответствующих им запросов  
для описания схемы данных технических средств, обеспечивающих  
выполнение установленных действий при проведении оперативно-разыскных  
мероприятий**

№ п/п	Описание типа	Сервисный тип	Запрос	Описание запроса
1	Схема данных, описанная на языке SDL	Отсутствует	<pre> type Query {   """   Запрос схемы данных   """   getSchema: String! } </pre>	Запрос схемы данных, описанной на языке SDL GraphQL
2	Идентификатор отложенного запроса	Отсутствует	<pre> type Query {   """   Запрос результата выполнения отложенного запроса   """   getOfflineRequest(requestID: String!, offset: Int, limit: Int, cursor: String): QueryMessage! } </pre>	Запрос результата выполнения отложенного запроса по его идентификатору либо страничное получение результатов с использованием аргументов «offset», «limit» и «cursor» согласно подпункту 2 пункта 9 приложения № 2 к требованиям, утвержденным настоящим приказом. Ответ (результат

			<p>выполнения запроса) предоставляется в виде данных в формате JSON в соответствии с изначальным GraphQL запросом. Если отложенный запрос с указанным идентификатором не найден, то должно быть возвращено сообщение об ошибке GraphQL с кодом REQUEST_NOT_FOUND.</p> <p>Если отложенный запрос еще не выполнен или результаты его выполнения уже удалены, то должно быть возвращено сообщение об ошибке GraphQL с кодом NO_REQUEST_RESULT.</p>	
			<pre> type Query {   """   Удаление результатов   отложенного запроса   """   delOfflineRequest(requestID: String!):   Boolean! } </pre>	<p>Удаление результатов отложенного запроса по его идентификатору. Если удаление произошло успешно, то должен быть возвращен ответ True.</p> <p>Если удаление не удалось выполнить, то должен быть возвращен ответ False и описание</p>

				<p>ошибки GraphQL. Если отложенный запрос с указанным идентификатором не найден, то должно быть возвращено сообщение об ошибке GraphQL с кодом REQUEST_NOT_FOUND.</p>
			<pre> type Query {     """     Отмена выполнения отложенного запроса     """     cancelOfflineRequest(requestID: String!): Boolean! } </pre>	<p>Прерывание (отмена) выполнения отложенного запроса по его идентификатору. Если прерывание произошло успешно, то должен быть возвращен ответ True. Если прерывание не удалось выполнить, то должен быть возвращен ответ False и описание ошибки GraphQL. Если отложенный запрос с указанным идентификатором не найден, то должно быть возвращено сообщение об ошибке GraphQL с кодом REQUEST_NOT_FOUND.</p>
3	Информация о статусе выполнения отложенного запроса	<pre> """ Сервисный тип для информации о статусе выполнения отложенного запроса """ type RequestInfo { </pre>	<pre> type Subscription {     """     Запрос статуса отложенного запроса     """     statusOfflineRequest(requestID: String!): RequestInfo! } </pre>	<p>Запрос статуса отложенного запроса по его идентификатору. Ответ предоставляется в</p>

		<pre> Идентификатор отложенного запроса """ requestID: String! """ Статус отложенного запроса """ status: RequestStatus """ Дата и время начала выполнения отложенного запроса """ startsAt: DateTimeBase }  """ Возможные статусы выполнения отложенного запроса """ enum RequestStatus { """ Запрос еще не выполняется """ NOTSTARTED """ Запрос выполняется """ RUNNING """ Запрос выполнен """ READY """ Выполнение запрос прервано ТС OPM """ ABORTED """ Запрос отменен ПУ """ CANCELED } </pre>	}	<p>виде сообщения типа RequestInfo. Если отложенный запрос с указанным идентификатором не найден, то должно быть возвращено сообщение об ошибке GraphQL с кодом REQUEST_NOT_FOUND.</p>
4	HTTP URI для неформатированных данных	<pre> """ Сервисный тип для неформатированных данных """ type Media { """ HTTP URI для неформатированных данных в полном формате согласно RFC7230 http(s)://&lt;IP-адрес или доменное имя ТС OPM&gt;:&lt;порт&gt;/download/&lt;у никальный путь к файлу&gt; """ </pre>	Отсутствует	Отсутствует

		<pre> link: String! } </pre>		
5	Сигналы о функционировании ТС ОРМ	<pre> """ Сервисный тип для сигналов о функционировании ТС ОРМ """ type Trap {   """   Тип сигнала   """   type: TrapType!   """   Описание сигнала   """   message: String   """   Дата и время возникновения сигнала   """   startsAt: DateTimeBase!   """   Дополнительная информация о предупреждениях для контролируемых метрик. Только для сигнала типа METRICALERTS   """   metricalerts: MetricAlerts }  """ Возможные типы сигналов """ enum TrapType {   """   Перезапуск ПО   """   RESTARTDB   """   Попытка несанкционированного доступа   """   UNAUTHORIZEDACCESS   """   Критическая ошибка ПО   """   CRITICALERROR   """   Серьезная ошибка ПО   """   MAJORERROR   """   Незначительная ошибка ПО   """   MINORERROR </pre>	<pre> type Subscription {   """   Запрос получения сигналов о функционировании ТС ОРМ   """   trap: Trap! } </pre>	Запрос получения сигналов о функционировании ТС ОРМ. Ответ предоставляется в виде сообщения типа Trap. Если тип сигнала «Предупреждение о проблеме с контролируемыми параметрами (METRICALERTS)», то ТС ОРМ должны возвращать дополнительную информацию о предупреждениях в поле «metricalerts».

```

    """
    Изменение схемы
    данных
    """
    SCHEMACHANGED
    """
    Предупреждение о
    проблеме с
    контролируемой метрикой
    """
    METRICALERTS
  }

  """
  Дополнительная
  информация о
  предупреждении
  """
  type MetricAlerts {
    """
    Ключ, определяющий
    группу предупреждений
    """
    groupKey: String!
    """
    Статус предупреждения
    """
    status: AlertStatus!
    """
    Определяет получателя
    предупреждения
    """
    receiver: String!
    """
    Метки, по которым
    сгруппированы
    предупреждения
    """
    groupLabels:
    [KeyValueRef!]!
    """
    Общие метки для всех
    предупреждений
    """
    commonLabels:
    [KeyValueRef!]!
    """
    Общие описания для
    всех предупреждений
    """
    commonAnnotations:
    [KeyValueRef!]!
    """
    URL менеджера
    предупреждений
    """
    externalURL: String!
    """
    Список предупреждений
    """
    alerts: [Alert!]!
  }

```

		<pre> """ Возможные статусы предупреждений """ enum AlertStatus {     """     Проблема решена     """     RESOLVED     """     Проблема не решена     """     FIRING }  """ Предупреждение о проблеме с контролируемой метрикой """ type Alert {     """     Статус предупреждения     """     status: AlertStatus!     """     Метки для     предупреждения     """     labels:     [KeyValueRef!]!     """     Описания для     предупреждения     """     annotations:     [KeyValueRef!]!     """     Дата и время     возникновения     предупреждения     """     startsAt:     DateTimeBase!     """     Дата и время решения     проблемы, в результате     которой возникло     предупреждение     """     endsAt: DateTimeBase!     """     URL сущности, в     которой возникла     проблема     """     generatorURL: String!     """     Идентификатор     предупреждения     """     fingerprint: String! } </pre>		
--	--	--	--	--

		<pre>""" Ключ-значение """ type KeyValueRef {     """     Ключ     """     key: String!     """     Значение     """     value: String! }</pre>		
--	--	---	--	--