

ООО «ДинСофт»

**Специальные запросы  
к WEB-серверу  
программы  
«ДинРобот»**

Евстигнеев Д.В.

Москва, 2018

## 1. Введение

Экранный контент робота представляет собой набор WEB-страниц, реализованных на базе браузера Google Chrome, запущенного в режиме киоска (полноэкранного приложения).

Загружаются WEB-страницы с локального WEB-сервера, реализованного на базе бесплатного приложения DENWER.

Для интеграции WEB-страницы с программой «ДинРобот» используется протокол HTTP. В бортовой программе «ДинРобот» открыт WEB-сервер на порту 5000, на который можно отправлять специальные запросы с WEB-страниц экранного контента.

Если запрос на порт 5000 не соответствует никакому специальному запросу, то загружается документ из папки C:\DynRobot\HTTP.

Запросы можно формировать не обязательно с WEB-страниц локального бортового компьютера робота, но и с удаленных компьютеров, смартфонов и планшетов, что позволяет реализовать WEB-интерфейс для удаленных пультов управления роботом. Так, например, если разместить в папке C:\DynRobot\HTTP документ my.html, то его можно открыть на удаленном компьютере, смартфоне или планшете, подключенного к локальной WiFi-сети робота, с помощью запроса:

**http://<ip-адрес>:5000/my.html**

Здесь: <ip-адрес> – IP-адрес или сетевое имя компьютера робота. Для локального бортового компьютера робота – это адрес 127.0.0.1.

## 2. Специальные запросы к «ДинРобот»

### 2.1. Интеграция WEB-страниц экранного контента и бортовых скриптов iScript

В языке бортовых скриптов робота iScript существует функция управления WEB-страницей (функция `SetBrowserURL(url)`). Данная функция предназначена для открытия WEB-страниц в браузере или для выполнения на них требуемых скриптов на языке JavaScript.

Однако, WEB-контент отображается на экране с помощью браузера Google Chrome. Прямое управление этим браузером из стороннего программного обеспечения невозможно. Управлять им можно только с самих WEB-страниц, которые он загружает. Поэтому в программном комплексе «ДинРобот» предусмотрено, что каждая WEB-страница экранного контента сформирует специальный асинхронный HTTP-запрос на WEB-сервер программы «ДинРобот» с бесконечным таймаутом. Как только в скриптах «ДинРобот» будет вызвана функция `SetBrowserURL(...)`, WEB-сервер программы «ДинРобот» сформирует ответ на этот HTTP-запрос с указанием адреса, куда следует перейти браузеру, или скрипта, который следует исполнить браузеру. После выполнения соответствующих действий WEB-страница должна будет снова сформировать аналогичный асинхронный HTTP-запрос с бесконечным таймаутом и ждать следующей команды.

Для обеспечения работы функции `GetBrowserURL()`, реализованной в языке бортовых скриптов iScript, асинхронный HTTP-запрос, о котором идет речь, передает в «ДинРобот» в качестве параметра URL текущей страницы браузера.

Асинхронный HTTP-запрос с бесконечным таймаутом формируется методом GET на порт 5000 программы «ДинРобот», и имеет следующий вид:

```
http://<ip-адрес>:5000/alive?url=<текущий_URL>
```

Где:

<ip-адрес> – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;

<текущий\_URL> – URL текущей WEB-страницы браузера.

В ответ на данный HTTP-запрос WEB-сервер программы «ДинРобот» может сформировать следующий ответ (не считая HTTP-заголовков):

```
<НОВЫЙ_URL>
```

ИЛИ

```
javascript://<скрипт>
```

Где:

<Новый\_URL > – адрес WEB-страницы, куда следует перейти;

<скрипт> – JavaScript, который следует выполнить.

WEB-сервер может либо вовсе не сформировать никакого ответа, оставив открытым TCP/IP-сокет вплоть до закрытия данной WEB-страницы, либо сформировать ответ по нему, спустя продолжительное время (в зависимости от того, когда будет в iScript вызвана функция SetBrowserURL).

Пример javascript-файла (с названием scripts.js), реализующего соединение с «ДинРобот»:

```
control_connection = null;

// функция обработки ответа на асинхронный HTTP-запрос
function control_connection_on_data()
{
    if (this.readyState==4)
    {
        if (this.status == 200)
        {
            if (this.responseText!="")
            {
                // в зависимости от ответа сервера
                if (this.responseText.indexOf('javascript://')==0)
                {
                    // выполнить JavaScript
                    var js = control_connection.responseText.substring(13);
                    control_connection = null;
                }
            }
        }
    }
}
```

```

        eval(js);
    }
    else
    {
        // перейти по указанному URL
        location = control_connection.responseText;
    }
}
}
control_connection = null;
}
}

// функция, вызываемая по таймеру каждые 200 мс
function control_connection_on_timer()
{
    // если соединение с ДинРобот еще не открыто, то открыть его
    if (control_connection == null)
    {
        var url="http://127.0.0.1:5000/alive?url="+encodeURIComponent(location);
        control_connection = new XMLHttpRequest();

        control_connection.onreadystatechange=control_connection_on_data;
        control_connection.open("GET", url, true);
        control_connection.send();
    }
}

// запустить таймер
setInterval('control_connection_on_timer()',200);

```

Основная страница сайта при этом выглядит следующим образом:

```

<html>
<body>

<!-- подключение скрипта scripts к WEB-странице -->
<script src='scripts.js'></script>

<h1>Привет мир</h1>
</body>
</html>

```

## 2.2. Запрос к «ДинРобот» на произношение речи с WEB-страницы экранного контента

С WEB-страницы экранного контента можно передать специальный HTTP-запрос на WEB-сервер программы «ДинРобот» для формирования фразы синтезатором речи. Проще говоря, заставить с WEB-страницы сказать робота какую-либо фразу.

Для этого нужно сформировать специальный HTTP-запрос на порт 5000 программы «ДинРобот» методом POST (при Content-Type: application/x-www-form-urlencoded):

```
http://<ip-адрес>:5000/playspeech
```

Здесь:

<ip-адрес> – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1.

В качестве POST-данных HTTP-запросу следует передать параметр text со значением текста произносимой фразы в кодировке UTF-8.

Если запрос правильно сформирован WEB-сервер программы «ДинРобот» отправит в ответ «ОК» (не считая HTTP-заголовков). Однако разработчику WEB-контента, в принципе, нет никого смысла вообще обрабатывать ответ сервера в данном случае.

Пример HTML-страницы (страница должна быть в кодировке UTF-8):

```
<html>
<body>
<h1>Говорящая WEB-страница</h1>

<script>

// функция, говорящая текст через «ДинРобот»
function PlaySpeech(text)
{
  var url = "http://127.0.0.1:5000?playspeech";
  var postData = "text="+encodeURIComponent(text);
  var xhr = new XMLHttpRequest();
  xhr.timeout = 3000;
  xhr.open("POST", url, true);
  xhr.setRequestHeader('Content-Type',
```

```

'application/x-www-form-urlencoded');

xhr.send(postData);
}

// Сказать текст
PlaySpeech("Привет мир!");

</script>
</body>
</html>

```

### 2.3. Запрос на воспроизведение WAVE-файла через «ДинРобот»

С WEB-страницы экранного контента можно передать специальный HTTP-запрос на WEB-сервер программы «ДинРобот» для воспроизведения WAVE-файла.

Для этого нужно сформировать специальный HTTP-запрос на порт 5000 программы «ДинРобот» методом GET:

```
http://<ip-адрес>:5000/playwave?file=<имя_файла>
```

Здесь:

*<ip-адрес>* – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;

*<имя\_файла>* – имя WAVE-файла (если не указан путь, то относительно папки C:\DynRobot).

Если запрос правильно сформирован WEB-сервер программы «ДинРобот» отправит в ответ «ОК» (не считая HTTP-заголовков). Однако разработчику WEB-контента, в принципе, нет никакого смысла вообще обрабатывать ответ сервера в данном случае.

Пример HTML-страницы (страница должна быть в кодировке UTF-8):

```

<html>
<body>
<h1>Звуковая WEB-страница</h1>

<script>
// функция, запускающая звук через «ДинРобот»
function PlayWave(file)
{

```

```

var url = "http://127.0.0.1:5000?playwave?file="+encodeURIComponent(file);
var xhr = new XMLHttpRequest();
xhr.timeout = 3000;
xhr.open("GET", url, true);
xhr.send(postData);
}

// Воспроизвести звук
PlayWave("snap.wav");

</script>
</body>
</html>

```

#### 2.4. Запрос на формирование события в iScript

С WEB-страницы экранного контента можно передать специальный HTTP-запрос на WEB-сервер программы «ДинРобот» для формирования в iScript какого-либо события, обрабатываемого с помощью механизма фреймов.

Для этого нужно сформировать специальный HTTP-запрос на порт 5000 программы «ДинРобот» методом GET:

```
http://<ip-адрес>:5000/command?cmd=<команда>
```

Здесь:

<ip-адрес> – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;

<команда> – текст команды (события) для скриптов iScript.

Если запрос правильно сформирован WEB-сервер программы «ДинРобот» отправит в ответ «ОК» (не считая HTTP-заголовков). Однако разработчику WEB-контента, в принципе, нет никого смысла вообще обрабатывать ответ сервера в данном случае.

Пример HTML-страницы, клик по которой заставляет робота здороваться:

```

<html>
<body onclick="OnBodyClick()">
<h1>WEB-страница</h1>
<p>Клигни в меня</p>

```



```

<script>
// функция, отправляющая событие в «ДинРобот»
function SendCmd(cmd)
{
  var url = "http://127.0.0.1:5000?command?cmd="+encodeURIComponent(cmd);
  var xhr  = new XMLHttpRequest();
  xhr.timeout = 3000;
  xhr.open("GET", url, true);
  xhr.send(postData);
}

// функция, вызываемая на клик мышкой по странице
function OnBodyClick()
{
  // отправить событие в «ДинРобот»
  SendCmd(" * ACTIVE");
}
</script>
</body>
</html>

```

На языке iScript для обработки данного события должен быть предусмотрен соответствующий обработчик события. Для приведенного примера обработчик будет выглядеть следующим образом:

```

frameset("commands", 1)
{
  frame(" * ACTIVE")
  {
    PlaySpeech("Привет", "Приветствую Вас");
  }
}
while(true) sync();

```

## 2.5. Запрос с WEB-страницы, включающий автомат

С WEB-страницы экранного контента можно передать специальный HTTP-запрос на WEB-сервер программы «ДинРобот» для включения или отключения автомата.

Для этого нужно сформировать специальный HTTP-запрос на порт 5000 программы «ДинРобот» методом GET:

```
http://<ip-адрес>:5000/automat?state=<состояние>
```

Здесь:

<ip-адрес> – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;

<состояние> – состояние автомата (1 – включено, 0 – выключено).

Если запрос правильно сформирован WEB-сервер программы «ДинРобот» отправит в ответ «ОК» (не считая HTTP-заголовков). Однако разработчику WEB-контента, в принципе, нет никакого смысла вообще обрабатывать ответ сервера в данном случае.

## 2.6. Запрос видеоизображения с камеры робота

Видеоизображение с камеры робота на WEB-странице отображается в виде набора кадров в формате JPEG. На WEB-странице необходимо реализовать таймер, который бы запрашивал очередной кадр видеоизображения с частотой 12-15 кадров в секунду (быстрее не следует, WEB-сервер или браузер не успевают).

Запрос одного кадра изображения формируется следующим GET-запросом по протоколу HTTP:

```
http://<ip-адрес>:5000/video
```

Однако для борьбы с кэшированием браузера желательно делать запросы вида:

```
http://<ip-адрес>:5000/video?rnd=<random>
```

Где:

<ip-адрес> – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;

<random> – случайное число.

В ответ на запрос сервер программы «ДинРобот» возвращает кадр изображения в формате JPEG (не считая HTTP-заголовков).

Пример HTML-страницы:

```
<html>
<body>


<script>
img = document.getElementById('img');
function OnTimer()
{
    img.src = "http://127.0.0.1:5000/video?rnd="+Math.random();
}
}
```

```
setInterval(OnTimer, 80);
</script>
</body>
</html>
```

## 2.7. Запрос фотографии с камеры робота

Робот может сделать с робота фотографию, при необходимости наложить на нее рамку или логотип и сохранить результат в указанный файл в формате JPEG. Для этого в программу «ДинРобот» необходимо отправить специальный HTTP-запрос методом GET на порт 5000:

```
http://<ip-адрес>:5000/photo?photo=<файл_результата>
```

ИЛИ

```
http://<ip-адрес>:5000/photo?photo=<файл_результата>&mask=<файл_рамки>
```

ИЛИ

```
http://<ip-адрес>:5000/photo?photo=<файл_результата>
&mask=<файл_рамки>&analyze=<a>
```

Где:

*<ip-адрес>* – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;

*<файл\_результата>* – полное имя файла в файловой системе робота, куда следует записать результирующее фото. Обычно этот файл размещается в папке, к которой можно получить доступ по HTTP.

*<файл\_рамки>* – полное имя файла в файловой системе робота, где находится рамка для фото. Если данный параметр не указывается или он пустой, то наложение рамки на фотографию не производится.

*<a>* – признак необходимости анализа фото (0 или 1). Анализ фото позволяет оценить число человек в кадре и выдать рекомендации по расположению людей в кадре.

Сервер в ответ на запрос возвращает текстовую строку (не считая HTTP-заголовков):

```
OK
```

или, если признак запроса  $\langle a \rangle$  равен 1,

OK RECOMMENATION= $\langle код \rangle$  PERSONS= $\langle число\_персон \rangle$

Любой другой ответ сервера следует расценивать как ошибку.

Здесь:

$\langle число\_персон \rangle$  – количество лиц, изображенных на фото. Если на фото нет лиц, то  $\langle число\_персон \rangle$  равно 0.

$\langle код \rangle$  – код рекомендации:

- 1 – хорошее фото.
- 2 – на фото не видно лиц.
- 3 – на фото слишком темные лица.
- 4 – на фото есть лица без улыбки (сейчас не работает).
- 5 – все лица в правой части кадра. Нужно сместиться левее.
- 6 – все лица в левой части кадра. Нужно сместиться правее.
- 7 – очень маленькие лица. Нужно ближе подойти.
- 8 – очень большие лица. Нужно подальше отойти.
- 9 – все лица очень низко. Нужно ближе подойти.
- 10 – все лица очень высоко. Нужно назад отойти.

Файл рамки должен быть выполнен в виде прозрачного PNG-файла.

При наложении рамки на фотографию производится оценка размеров прозрачной зоны, а сама фотография масштабируется так, чтобы она могла без полей вписаться в эту прозрачную зону рамки. Причем, фотография масштабируется до размеров рамки, а не наоборот. Поэтому файл рамки должен иметь размер по длине и ширине не ниже требуемого разрешения.

Кроме того, если совместно с роботом используется принтер, то рекомендуется у файла рамки делать такое соотношение сторон, которое бы соответствовало соотношению сторон бумаги принтера. Обычно фотобумага имеет соотношение сторон 3:2. Поэтому рекомендуется делать фоторамку с размерами 2900x2000 пикселей (чуть меньше размера кадра).

Пример HTML-страницы:

```

<html>
<body>
<center>
  
</center>

<center>
  <form>
    <input type=button value="ФОТО" onclick="Snap()">
  </form>
</center>

<script>
img = document.getElementById('img');
mode = 0; // режим: 0 - предпросмотр видео, 1 - отображение фото

// таймер отображение видео
function OnTimer()
{
  if (mode == 0)
    img.src = "http://127.0.0.1:5000/video?rnd="+Math.random();
}
setInterval(OnTimer, 80);

// функция, вызываемая на нажатие кнопки «ФОТО»
function Snap()
{
  mode = 1; // установить фото режим

  // придумать название фото
  var t = (new Date()) - 0; // время в наносекундах
  var shortFileName = t + ".jpg"; // краткое название файла
  // полное название файла (путь к файлу на диске робота)
  var fullFileName =
"c:\\WebServices\\home\\mysite\\www\\photo\\"+
      shortFileName;
  // полное имя файла с фото рамкой
  var maskFile = "c:\\WebServices\\home\\mysite\\www\\mask.png";

  // URL-запроса
  var url = "http://127.0.0.1:5000/photo?photo=" +
    encodeURIComponent(fullFileName) + "&mask=" + encodeURIComponent(maskFile);

  // сформировать асинхронный HTTP-запрос
  var xhr = new XMLHttpRequest();
  xhr.timeout = 10000; // таймаут 10 секунд
  xhr.onreadystatechange = function()
  {
    // функция обработки результата асинхронного запроса
    if (this.readyState==4)
    {
      if (this.status == 200)
      {
        if (this.responseText.indexOf("OK")>=0)
        {

```

```

        // Отобразить фото.
        img.src = "photo\\"+shortFileName;
        return;
    }
}
// отобразить фото с ошибкой
img.src = "error.jpg";
}

};
xhr.open("GET", url, true);
xhr.send();
}
</script>
</body>
</html>

```

## 2.8. Запросы реализации сервиса «Робо-Аватар»

Сервис «Робо-Аватар» – это развлекательный контент типа «Сфоткай себя и ты станешь роботом». Робот выделяет лицо на кадре видеоизображения с камеры робота, обводит его рамкой и, при формировании команды на съемку, ставит это лицо в качестве своей аватарки (виртуального лица робота). Лицо держится на экране несколько секунд, а затем меняется на штатное лицо робота.

Сервис «Робо-Аватар» имеет смысл использовать только на роботах, имеющих аватарку (виртуальное лицо).

Для реализации этого сервиса на WEB-странице необходимо отображать последовательность кадров видеоизображения. Кадры следует запрашивать по таймеру с частотой 5-10 кадров в секунду (быстрее не справляется алгоритм выделения лиц).

Для запроса кадра изображения для сервиса «Робо-Аватар» следует сформировать следующий HTTP-запрос методом GET:

```
http://<ip-адрес>:5000/avatar?snap=<snap>
```

Однако для борьбы с кэшированием браузера имеет смысл запрашивать документ следующим образом:

```
http://<ip-адрес>:5000/avatar?snap=<snap>&rnd=<random>
```

Где:

- <ip-адрес>* – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;
- <snap>* – сигнал на захват лица для формирования из него аватарки робота (0 или 1);
- <random>* – случайное число.

Сигнал *<snap>* необходимо держать в состоянии равным 1 в течение одной секунды (в течение несколько кадров), чтобы программной «ДинРобот» гарантировано не был пропущен запрос на захват аватара.

Пример WEB-страницы:

```

<html>
<body>

<form>
<input type="button" value="ФОТО" onclick="Snap()">
</form>

<script>
img = document.getElementById('img');
snap = 0;

// функция вызываемая по таймеру каждые 100 мс
function OnTimer()
{
    img.src = "http://127.0.0.1:5000/avatar?snap="+ (snap>0?1:0)+
            "&rnd="+Math.random();
    if (snap>0) snap-=100;
}

// установить таймер
setInterval(OnTimer, 100);

// функция, запускаемая при нажатии кнопки «ФОТО»
function Snap()
{
    snap = 1000; // завести таймер съемки на 1000 мс
}
</script>
</body>
</html>

```

## 2.9. Запросы реализации сервиса «Фото-виньетки»

Сервис «Фото-виньетки» – это развлекательный сервис, позволяющий выделить лицо человека с камеры робота и подставить его в фотографию другого персонажа (виньетку).

Виньетки должны представлять собой PNG-файлы, в которых прозрачным цветом вырезано лицо. Важно, чтобы лицо в этих фотографиях не было наклонено или повернуто.

Для реализации этого сервиса на WEB-странице необходимо отображать последовательность кадров видеоизображения. Кадры следует запрашивать по таймеру с частотой 5-10 кадров в секунду. При формировании в запросе дополнительного параметра кадр можно сохранить на диск робота в виде JPEG-файла.

Для запроса кадра изображения для сервиса «Фото-виньетки» следует сформировать следующий HTTP-запрос методом GET:

```
http://<ip-адрес>:5000/vignette?mask=<виньетка>&rnd=<random>
```

или:

```
http://<ip-адрес>:5000/vignette?mask=<виньетка>&rnd=<random>&save=<файл>
```

Где:

- <ip-адрес> – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;
- <виньетка> – полное имя PNG-файла в файловой системе робота, в котором изображение виньетки;
- <random> – случайное число;
- <save> – полное имя файла, в который следует сохранить JPEG-файл с результатом. Может быть пустым, если сохранять файл не нужно.

Параметр <save> следует формировать только у последнего результирующего кадра. Живое видео следует запрашивать без установленного параметра <save>.

В ответ на запрос возвращается JPEG-файл с наложенным в виньетку лицом.

Пример WEB-страницы:

```
<html>  
<body>
```



```


<form>
<input type="button" value="ФОТО" onclick="Snap()">
</form>

<script>
img = document.getElementById('img');
save = '';
mode = 0; // режим: 0 - живое видео, 1 - фото с конеч.результатом

// функция вызываемая по таймеру каждые 100 мс
function OnTimer()
{
    if (mode == 0)
    {
        img.src = "http://127.0.0.1:5000/vignettes"+
            "?mask= c:\\vignette\\1.png&save="+encodeURIComponent(save)+
            "&rnd="+Math.random();

        if (save!="")
        {
            save = "";
            mode = 1;
        }
    }
}

// установить таймер
setInterval(OnTimer, 100);

// функция, запускаемая при нажатии кнопки «ФОТО»
function Snap()
{
    // формируем имя файла для конечного результата
    var t = (new Date()) - 0; // время в наносекундах
    save = "c:\\WebServices\\home\\MySite\\www\\photo\\"+t+".jpg";
}
</script>
</body>
</html>

```

## **2.10. Запросы реализации сервиса добавления лиц в базу данных**

Сервис добавления лиц в базу данных работает только при использовании системы распознавания лиц (не путать с системой детектирования лиц).

Запросы данного сервиса позволяют добавить в базу данных лиц программы «ДинРобот» новое лицо, регистрируемое в данный момент системой распознавания лиц, под определенным идентификатором.

Перед тем, как добавить лицо в базу данных, необходимо, чтобы система распознавания лиц была включена, а также распознавала детали лица (пол и возраст). Это можно сделать через iScript. Однако рекомендуется включать распознавание, путем HTTP-запроса кадров изображений с камеры специальным запросом (метод GET):

```
http://<ip-адрес>:5000/video?facedetails=1&facerecognizer=1&rnd=<random>
```

Где:

<ip-адрес> – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;

<random> – случайное число для борьбы с кэшированием.

В ответ на данный запрос WEB-сервер программы «ДинРобот» формирует кадр видеоизображения в формате JPEG (не считая соответствующих HTTP-заголовков).

Запрос кадров видеоизображения в указанном формате необходимо повторять по таймеру в течение нескольких секунд. За это время пользователь сможет увидеть себя на экране робота и правильно расположить лицо в кадре.

Непосредственно запрос сам HTTP-запрос на добавление лица в базу данных имеет вид (метод GET):

```
http://<ip-адрес>:5000/addperson?id=<ID>&temp=<temp>
```

Где:

<ip-адрес> – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1.

<ID> – текстовый идентификатор лица в базе данных (соответствует названию директории, в которую будет размещено фото лица и его биометрический шаблон). Кодировка UTF-8.

<temp> – временное добавление (1) или постоянное (0).

Идентификатор лица должен состоять только из символов, разрешенных к использованию в названиях директорий. При использовании русских букв следует аккуратно следить за кодировкой символов.

Временное добавление лица в базу данных распространяется только на один сеанс работы программы «ДинРобот». После перезапуска программы «ДинРобот» информация о лице будет забыта. Постоянное добавление лица приводит к созданию на диске робота в папке C:\DynRobot\FaceDB директории, название которой совпадает с идентификатором лица. В данную директорию «ДинРобот» размещает фотография лица и его биометрический шаблон.

Рекомендуется разработать WEB-страниц таким образом, чтобы можно было бы ассоциировать идентификатор и имя человека. Например, с использованием базы данных MySQL.

В ответ на запрос на добавление лица в базу данных WEB-сервер программы «ДинРобот» формирует следующие ответы (не считая HTTP-заголовков).

В случае успеха:

```
OK person=<ID> gender=<пол> age=<возраст>
```

Где:

<ID> – идентификатор лица (UTF-8).

<пол> – пол (1 – «Муж», 2 – «Жен», 0 – не определено).

<возраст> – примерный возраст (целое число).

В случае, если добавляемое лицо уже идентифицируется какое либо лицо, то ответ WEB-сервера «ДинРобот» будет иметь вид:

```
EXISTS <existsID>
```

Где:

<existsID> – идентификатор обнаруженного в базе данных лица (UTF-8).

В случае ошибки добавления лица в базу данных «ДинРобот» возвращает текст ошибки.

Пример HTML-страницы.

```
<html>
<body>
<h1>Дайте мне на вас взглянуть</h1>

<img src='http://127.0.0.1/video?facedetails=1&facerecognizer=1'
id=img>
<br>

<form name=main>
Введите ваше имя: <input name=userID>
<input type=button value="Добавить" onclick="DoAdd()">
</form>

<script>
img = document.getElementById('img');

// функция, вызываемая по таймеру каждые 100 мс
function OnTimer()
{
    // запрос изображения
    img.src='http://127.0.0.1/'+

'video?facedetails=1&facerecognizer=1&rnd='+Math.random();
}
// установить таймер
timer = setInterval(OnTimer, 100);

// функция, вызываемая при нажатии кнопки «Добавить»
function DoAdd()
{
    var userID = document.forms['main'].elements['userID'].value;
    if (userID == '')
    {
        alert("Введите ваше имя!");
        return;
    }

    // запрос на добавление лица в базу данных
    xhr = new XMLHttpRequest();
    xhr.timeout = 4000;
    xhr.onreadystatechange=function()
    {
        // функция обработки ответа сервера
        if (this.readyState==4 && this.status == 200)
        {
            var g = this.responseText.match(/EXISTS (\s+)/);
            if (g)
            {
                alert('А я Вас знаю, вы '+g[1]);
            }
        }
    }
}
```

```

        return;
    }

    g = this.responseText.match(/OK
person=(\s+)\s+gender=([^\s+])\s+age=([^\s+])/);
    if (g)
    {
        alert('Очень хорошо, '+g[1]+' , я вас запомнила');
        // перейти на другую страницу
        location = "index.html";

    }
    else
    {
        // ошибка
        alert(this.responseText);
    }
}
};

var url =
    "http://127.0.0.1:5000/addperson?id="+userID+"&temp=0";
xhr.open("GET", url, true);
xhr.send();
}
</script>
</body>
</html>

```

## 2.11. Запросы развлекательного сервиса гадания

Развлекательный сервис «гадание» реализован на базе идентификации пола и возраста людей системой распознавания лиц. После система выдает среднестатистический прогноз, соответствующий полу и возрасту человека, что в большинстве случаев совпадает с проблемами и ожиданиями людей в этом возрасте.

Для активизации сервиса необходимо организовать процесс запроса видеокладов изображения с помощью следующего HTTP-запроса методом GET:

```
http://<ip-адрес>:5000/gadanie?active=1&rnd=<random>
```

Где:

<ip-адрес> – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;

<random> – случайное число для борьбы с кэшированием.

В ответ на данный запрос WEB-сервер программы «ДинРобот» формирует кадр видеоизображения в формате JPEG (не считая соответствующих HTTP-заголовков).

После нескольких секунд запроса кадров видеоизображения необходимо сформировать HTTP-запрос методом GET на получение результатов гадания:

```
http://<ip-адрес>:5000/gadanieresult&rnd=<random>
```

Где:

<ip-адрес> – адрес робота, для локальной WEB-страницы этот адрес будет 127.0.0.1;

<random> – случайное число для борьбы с кэшированием.

В ответ на запрос WEB-сервер программы «ДинРобот» формирует следующий JSON-документ (не считая HTTP-заголовков):

```
{
  "x": <x>,
  "y": <y>,
  "width": <width>,
  "height": <height>,
  "age": <возраст>,
  "gender": <пол>
}
```

Где:

<x>, <y>, <width>, <height> – положение, длина и ширина лица (в пикселях) на последнем кадре видеоизображения.

<возраст> – возраст лица, определенный системой распознавания лиц (целое число).

<gender> – пол лица, определенный системой распознавания лиц (1 – «Муж», 2 – «Жен», 0 – не определено).

Для придания эффектности рекомендуется реализовать WEB-страницу следующим образом. Сразу же после получения результатов гадания (а это происходит почти мгновенно) увеличить кадр видеоизображения с наведением на лицо человека. Одновременно с

увеличением реализовать плавный переход в видеоролик с эффектным процессом гадания, снабдив все соответствующей музыкой. По окончании видеоролика сделать плавный переход из анимации в станицу с результатами гадания. Выдаваемый роботом прогноз должен быть построен случайным образом, но соответствовать полу и возрасту человека. Красиво одновременно с этим показать на экране вырезанное из кадра лицо человека, его пол и возраст. Все вышеописанное легко реализуется на базе HTML5, и уже было реализовано автором программы «ДинРобот».