

# **Поднятие из архива сайта moveonow.kz**

## **0. Рабочая среда**

В качестве среды взята виртуальная машина с предустановленной чистой ОС без графической оболочки.

Рабочая (реальная) ОС: Mac OS 10.11.5  
Среда виртуализации: VirtualBox 5.0.26  
Характеристики виртуальной машины: 1 Гб ОЗУ, 20 Гб ПЗУ.  
ОС на виртуальной машине: debian-8.5.0-amd64-netinst.iso

## **1. Настройка окружения для сайта**

**1.1.** Подключаемся к машине по SSH с пользователем root:

```
ssh root@IP_МАШИНЫ
# ВВОДИМ ПАРОЛЬ
```

**1.2.** Убедимся, что стоят последние пакеты:

```
apt-get update
apt-get upgrade
```

**1.3.** Установим/проверим часовой пояс, локаль (en\_US.UTF-8 UTF-8) и время:

```
dpkg-reconfigure tzdata
dpkg-reconfigure locales
date
```

**1.4.** Добавим пользователя, которому будут принадлежать файлы данного сайта. Если на одной машине будет несколько сайтов, желательно создавать отдельного пользователя на каждый:

```
adduser moveonow
# ПРИДУМЫВАЕМ ПАРОЛЬ
```

**1.5.** Загружаем исходный код из архива по адресу moveonow/htdocs в папку /home/moveonow/htdocs через любую визуальную или консольную программу (например, FileZilla). Загружать лучше через созданного пользователя moveonow, иначе потом долго менять права/владельца на загруженные файлы.

**1.6.** Установим web-сервер. Убеждаемся, что порт 80 свободен:

```
netstat -tunlap | grep LISTEN | grep 80
```

Если порт 80 занят, например, Апачем (apache2), удаляем вместе с конфигами:

```
apt-get purge apache2*
```

Если же nginx уже установлен и слушает 80 порт, то дальше нет необходимости его повторной установки — используем его. Иначе, установим nginx:

```
apt-get install nginx
```

Убеждаемся, что nginx занял 80 порт:

```
netstat -tunlap | grep LISTEN | grep 80
```

Заходим через браузер на IP-адрес сервера и убеждаемся, что nginx отдает свою страницу по-умолчанию.

**1.7. Настроим nginx.** Создаем конфиг-файл сайта (здесь и далее nano можно заменить на vim или любой другой текстовый редактор):

```
nano /etc/nginx/sites-available/movenow
```

Содержимое взято с предоставленного разработчикам DEPLOY\_Movenow.MD, подставлены правильные пути и адреса:

```
server {
    listen      80;
    server_name movenow.kz;
    root        /home/movenow/htdocs;

    access_log  off;
    error_log   off;
    charset     utf-8;
    client_max_body_size 100M;

    location / {
        root  /home/movenow/htdocs/frontend/web;
        try_files $uri /frontend/web/index.php?$args;
    }

    location ~* \.php$ {
        try_files $uri /frontend/web$uri =404;
        fastcgi_pass unix:/var/run/movenow-php5-fpm.sock;
        fastcgi_read_timeout 1350;
        include  fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }

    # avoid processing of calls to non-existing static files by Yii
    # (uncomment if necessary)
    location ~* \.(css|js|jpg|jpeg|png|gif|bmp|ico|mov|swf|pdf|zip|rar)$ {
        access_log  off;
        log_not_found  off;
        try_files $uri /frontend/web$uri =404;
    }

    location ~* \.(htaccess|htpasswd|svn|git) {
        deny all;
    }

    location /admin {
        alias  /home/movenow/htdocs/backend/web;
        try_files $uri /backend/web/index.php?$args;

        location = /admin/ {
            return 301 /admin;
        }

        location ~* ^/admin/(.+\.php)$ {
            try_files $uri /backend/web/$1?$args;
        }

    # avoid processing of calls to non-existing static files by Yii
    # (uncomment if necessary)
    location ~* ^/admin/(.+\.(css|js|jpg|jpeg|png|gif|bmp|ico|mov|swf|
pdf|zip|rar))$ {
            try_files $uri /backend/web/$1?$args;
        }
    }
}
```

```

        }

server {
    charset utf-8;
    client_max_body_size 128M;

    listen 80;

    server_name movefuture.kz;
    root      /home/movenow/htdocs/movefuture/web;
    index     index.php;

    location / {
        # Redirect everything that isn't a real file to index.php
        try_files $uri $uri/ /index.php?$args;
    }

    location ~* \.php$ {
        try_files $uri =404;
        fastcgi_pass unix:/var/run/movenow-php5-fpm.sock;
        fastcgi_read_timeout 1350;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
}

```

**1.8.** Подключаем конфиг nginx в работу:

```
ln -s /etc/nginx/sites-available/movenow /etc/nginx/sites-enabled/movenow
```

Убеждаемся что в конфиге нет ошибок:

```
nginx -t
```

Перегружаем nginx:

```
service nginx restart
```

**1.9.** Установим PHP:

```
apt-get install php5-fpm php5-mysql
```

Убеждаемся что поставили PHP 5.6.\*

```
php -v
```

**1.10.** Настроим PHP. Создаем файл из шаблона:

```
cp /etc/php5/fpm/pool.d/www.conf /etc/php5/fpm/pool.d/movenow.conf
nano /etc/php5/fpm/pool.d/movenow.conf
```

Указываем и проверяем основные параметры, по примеру ниже:

```
...
; pool name ('www' here)
[movenow]
...
;
user = movenow
group = movenow
...
listen = /var/run/movenow-php5-fpm.sock
...
```

**1.11.** Перегружаем php5-fpm, чтобы конфиг ушел в работу:

```
service php5-fpm restart
```

Убеждаемся, что есть процессы php-fpm под пользователем movenow:

```
ps -ax | grep php
```

**1.12.** Установим СУБД MySQL. Убеждаемся, что пока ее нет:

```
mysql  
# bash: mysql: command not found
```

Если запустится интерпретатор MySQL, пропускаем этот пункт. Если нет, устанавливаем:

```
apt-get install mysql-server  
# ПРИДУМЫВАЕМ ПАРОЛЬ ДЛЯ ПОЛЬЗОВАТЕЛЯ СУБД root
```

**1.13.** Подключимся к СУБД и настроим БД:

```
mysql -uroot -p  
# ВВОДИМ ПРИДУМАННЫЙ ПАРОЛЬ  
create database if not exists movenow;  
create user 'movenow'@'localhost' identified by 'ПАРОЛЬ';  
# ПРИДУМЫВАЕМ ПАРОЛЬ ДЛЯ ПОЛЬЗОВАТЕЛЯ СУБД movenow  
grant all on movenow.* to 'movenow'@'localhost';  
flush privileges  
exit
```

**1.14.** Загружаем архив БД на сервер в текущую директорию (домашнюю для текущего пользователя, например /root), затем считываем архив в СУБД:

```
gunzip dreamvision-backup-2016-07-22.sql.gz  
mysql -umovenow -p movenow < dreamvision-backup-2016-07-22.sql  
# ВВОДИМ ПРИДУМАННЫЙ ДЛЯ movenow ПАРОЛЬ
```

**1.15.** Создаем хранимые процедуры, отсутствующие в архиве БД. Подключаемся к СУБД:

```
mysql -umovenow -p movenow  
# ВВОДИМ ПАРОЛЬ ПОЛЬЗОВАТЕЛЯ БД movenow
```

Создаем процедуры (код предоставлен разработчиками сайта в файле movenow\_function\_procedure.txt, однако в него внесены изменения, так как в исходном виде код не запускался):

```
DELIMITER $$  
  
DROP FUNCTION IF EXISTS AccessToBrandSite $$  
  
CREATE FUNCTION AccessToBrandSite(`user_id` INT)  
RETURNS TINYINT(4)  
DETERMINISTIC  
BEGIN  
    DECLARE access TINYINT;  
    SELECT COUNT(id) INTO access  
    FROM mevius_user  
    WHERE  
        (mevius_user.id = user_id)  
        AND  
        (reg_type_code = 1 AND status_reason_code = 3)  
        OR  
        (reg_type_code = 1 AND status_reason_code = 4)  
        OR  
        (reg_type_code = 3 AND status_reason_code = 4)
```

```

        OR
        (reg_type_code = 3 AND status_reason_code = 9)
        OR
        (reg_type_code = 4 AND status_reason_code = 5)
        OR
        (reg_type_code = 1 AND status_reason_code = 4)
        OR
        (reg_type_code = 1 AND doc_status_code = 3));
    RETURN access;
END $$

DELIMITER ;

DELIMITER $$

DROP PROCEDURE IF EXISTS `??` $$

CREATE PROCEDURE `??`(
    IN `t1` DATETIME,
    IN `t2` DATETIME
)
BEGIN

    SELECT baseTable.period AS 'Date',
        'RAPPMevius' AS 'Inserted By',
        baseTable.SynchPoint AS IntegrationPoint,
        'Работает' AS 'Статус работы',
        CASE
            WHEN baseTable.period = DATE(t1)
                THEN TIME (t1)
            ELSE TIME ('2015-01-01 00:00:00')
        END AS 'Период с_',
        CASE
            WHEN baseTable.period = DATE(t2)
                THEN TIME (t2)
            ELSE TIME ('2015-01-01 23:59:59')
        END AS 'Период по_',
        q_Number.q AS 'Количество запросов',
        '' AS 'Количество запросов создания УЗ (001, 001_1)',
        '' AS 'Количество запросов обновления УЗ (001, 001_1)',
        q_Err.q AS 'Количество ошибок в результате запроса'
    FROM (
        SELECT DISTINCT date(request_time) as period, 'int_web_001' AS SynchPoint
        FROM mevius_integration_int_web_001_transactions
        WHERE request_time BETWEEN t1 and t2
    UNION
        SELECT DISTINCT date(created_at), 'int_web_004'
        FROM mevius_integration_int_web_004_transactions
        WHERE created_at BETWEEN t1 AND t2
    UNION
        SELECT DISTINCT date(created_at), 'int_web_001_1'
        FROM mevius_integration_int_web_001_1_transactions
        WHERE created_at BETWEEN t1 AND t2
    UNION
        SELECT DISTINCT date(created_at), 'int_web_003'
        FROM mevius_integration_int_web_003_transactions

```

```

        WHERE created_at BETWEEN t1 AND t2

    UNION

    SELECT DISTINCT date(created_at), 'int_web_007'
    FROM mevius_integration_int_web_007_transactions
    WHERE created_at BETWEEN t1 AND t2
) AS baseTable
LEFT JOIN (
    SELECT COUNT(*) AS q, date(request_time) AS period, 'int_web_001' AS
SynchPoint, 'Количество запросов' AS description
    FROM mevius_integration_int_web_001_transactions
    WHERE request_time BETWEEN t1 AND t2
    GROUP BY date(request_time)

    UNION

    SELECT COUNT(*), date(created_at), 'int_web_004', 'Количество
запросов'
    FROM mevius_integration_int_web_004_transactions
    WHERE created_at BETWEEN t1 AND t2
    GROUP BY date(created_at)

    UNION

    SELECT COUNT(*), date(created_at), 'int_web_001_1', 'Количество
запросов'
    FROM mevius_integration_int_web_001_1_transactions
    WHERE created_at BETWEEN t1 AND t2
    GROUP BY date(created_at)

    UNION

    SELECT COUNT(*), date(created_at), 'int_web_003', 'Количество
запросов'
    FROM mevius_integration_int_web_003_transactions
    WHERE created_at BETWEEN t1 AND t2
    GROUP BY date(created_at)

    UNION

    SELECT COUNT(*), date(created_at), 'int_web_007', 'Количество
запросов'
    FROM mevius_integration_int_web_007_transactions
    WHERE created_at BETWEEN t1 AND t2
    GROUP BY date(created_at)

)
AS q_Number ON q_Number.period=baseTable.period AND
q_Number.SynchPoint=baseTable.SynchPoint

LEFT JOIN (
    SELECT
        COUNT(response_message) AS q, DATE(request_time) AS period,
        'int_web_001' AS SynchPoint, 'Количество ошибок' AS description
    FROM
        mevius_integration_int_web_001_transactions
    WHERE
        request_time BETWEEN t1 and t2
        AND IFNULL(response_message,'')<>'Success'
        group by DATE(request_time)

```

```

UNION

SELECT
count(IFNULL(response_status, 1)), DATE(created_at), 'int_web_004',
'Количество ошибок'
FROM
mevius_integration_int_web_004_transactions
WHERE
(created_at BETWEEN t1 AND t2)
AND response_status <> 1
group by DATE(created_at)

UNION

SELECT
count(IFNULL(error_message, 1)), DATE(created_at), 'int_web_001_1',
'Количество ошибок'
FROM
mevius_integration_int_web_001_1_transactions
WHERE
created_at BETWEEN t1 AND t2
AND IFNULL(error_message,'')<> ''
group by DATE(created_at)

UNION

SELECT DISTINCT COUNT(*), DATE(created_at), 'int_web_003', 'Количество
ошибок'
FROM
mevius_integration_int_web_003_transactions
WHERE
created_at BETWEEN t1 AND t2 AND response_status <> 1
GROUP BY DATE(created_at)

UNION

SELECT DISTINCT COUNT(*), DATE(created_at), 'int_web_007', 'Количество
ошибок'
FROM
mevius_integration_int_web_007_transactions
WHERE
created_at BETWEEN t1 AND t2 AND response_status <> 1
GROUP BY DATE(created_at)

) AS q_Err ON q_Err.period=baseTable.period AND
q_Err.SynchPoint=baseTable.SynchPoint
ORDER BY baseTable.SynchPoint ASC, baseTable.period ASC;

END $$

DELIMITER ;

```

## 2. Настройка системы управления сайта

**2.1.** Указываем имя пользователя `movenow` и придуманный для него пароль для подключения к БД в конфиге:

```
nano /home/movenow/htdocs/common/config/main-local.php
```

**2.2.** Настраиваем запуск фоновых задач через Cron:

```
crontab -e -u movenow
```

В открывшемся редакторе вводим и сохраняем следующие строки (содержимое взято с предоставленного разработчикам файла dreamvision.txt):

```
### WEB 001 1 ###

# Add new transaction period
*/5 * * * * php ~/htdocs/yii int-web-0011/add-period >> ~/htdocs/common/
modules/integration/log/intweb0011_addperiod.log 2>&1

# Parsing transactions
*/5 * * * * php ~/htdocs/yii int-web-0011/parse-queue 350 >> ~/htdocs/
common/modules/integration/log/intweb0011_parse.log 2>&1

# Get Customers via SOAP and save them into file
*/5 * * * * php ~/htdocs/yii int-web-0011/get-customers >> ~/htdocs/
common/modules/integration/log/intweb0011_getcustomers.log 2>&1

# Load saved xml with Customers to DB
*/5 * * * * php ~/htdocs/yii int-web-0011/load-xml-to-db >> ~/htdocs/
common/modules/integration/log/intweb0011_loadxmltodb.log 2>&1

### WEB 001 ###
*/15 * * * * php ~/htdocs/yii int-web-001/create-queue 1000 >> ~/htdocs/
common/modules/integration/log/intweb001_create_queue.log 2>&1
*/25 * * * * php ~/htdocs/yii int-web-001/parse-queue 1000 >> ~/htdocs/
common/modules/integration/log/intweb001_parse_queue.log 2>&1

### WEB 003 ###
*/5 * * * * php ~/htdocs/yii int-web-003/parse-queue 500 >> ~/htdocs/
common/modules/integration/log/intweb003_parse_queue.log 2>&1

### WEB 004 ###
*/5 * * * * php ~/htdocs/yii int-web-004/create-queue >> ~/htdocs/common/
modules/integration/log/intweb004_create_queue.log 2>&1
*/15 * * * * php ~/htdocs/yii int-web-004/parse-queue 500 >> ~/htdocs/
common/modules/integration/log/intweb004_parse_queue.log 2>&1

### GetDict ###

### Brands ###
00 00 * * * php ~/htdocs/yii get-dict/product >> ~/htdocs/common/modules/
integration/log/get_dict.log 2>&

### Cities ###
01 00 * * * php ~/htdocs/yii get-dict/city >> ~/htdocs/common/modules/
integration/log/get_dict.log 2>&

### Activity Types ###
02 00 * * * php ~/htdocs/yii get-dict/activity-type >> ~/htdocs/common/
modules/integration/log/get_dict.log 2>&

### Gifts ###
03 00 * * * php ~/htdocs/yii get-dict/gift >> ~/htdocs/common/modules/
integration/log/get_dict.log 2>&

### Gifts Types ###
04 00 * * * php ~/htdocs/yii get-dict/gift-type >> ~/htdocs/common/
modules/integration/log/get_dict.log 2>&

### Campaigns ###
05 00 * * * php ~/htdocs/yii get-dict/campaign >> ~/htdocs/common/modules/
integration/log/get_dict.log 2>&

### Status Codes ###
```

```
06 00 * * * php ~/htdocs/yii get-dict/status >> ~/htdocs/common/modules/integration/log/get_dict.log 2>&

### Status Case Codes ###  
07 00 * * * php ~/htdocs/yii get-dict/status-case >> ~/htdocs/common/modules/integration/log/get_dict.log 2>&

###Send Push Notifications###

00 17 * * 2,5 php ~/htdocs/yii push/send-full-register-invites >> ~/htdocs/push/logs/full-register-invites.log 2>&  
00 19 * * 2,5 php ~/htdocs/yii push/send-points-summary >> ~/htdocs/logs/push/points-summary.log 2>&  
00 20 * * 2,5 php ~/htdocs/yii push/send-new-filter-usage-notification >> ~/htdocs/logs/push/new-filter-usage-notification.log 2>&
```

После сохранения выходим из редактора, Cron возьмет новые команды в работу автоматически.

### 3. Резюме

Доступные для проверки страницы публичной и административной частей визуально работают, однако полноценно протестировать не представляется возможным, так как административная часть очень велика, а страницы публичной части сосредоточены в личном кабинете, который сам по себе не содержит данных и нуждается в полноценной проверке согласно специальным процедурам, которые должны быть предоставлены разработчиком или заказчиком.

Подготовил Конаныхин Илья, ikonanykhin@neolabs.kz

## Приложение: снимки экранов главной страницы и одной из страниц административной панели

The screenshot shows the main page of the Mevius app. At the top, there's a navigation bar with links for MOVIE, FUTURE, FRIENDS, MY MOVIES, MEVIUS, FUTURE SHOP, GALLERY, and PHOTO AWARDS. A search bar and user profile icons are also present. The main content area features a large circular graphic with two people taking a selfie, overlaid with the text 'KEEP CALM and MOVE'. Below this, there are three promotional sections: 'Сфотографируй свое увлечение' (Take a photo of your hobby), 'Используй фильтр в клиенте KEEPCALMandMOVE\_' (Use the filter in the KEEPCALMandMOVE\_ app), and 'Сделай реальное селфи и получи +100 баллов' (Take a real selfie and get +100 points). To the right, there's a call-to-action button 'СОЗДАЙ МОТИВАТОР' (Create a motivator) and another section about winning a hoverboard. Below this, there's a grid of 12 user posts from various users like Almack77, LilWayne, GRIFDESAN, Mayra, Pavlik99, DAMA, and others, each featuring a 'KEEP CALM and MOVE' filter. At the bottom, there's a footer note about smoking and a disclaimer in Kazakh.

MOVE NOW

БОЮС...

ИСКАТЬ

MEVIUS FUTURE SHOP GALLERY PHOTO AWARDS

СОЗДАЙ МОТИВАТОР

У тебя есть любимое занятие, которое захочет двигаться вперед? Сфотографируй свое увлечение, создай хобби-мотиватор с помощью фильтра и выиграй приз месяца с Mevius!

Сфотографируй свое увлечение

Используй фильтр в клиенте KEEPCALMandMOVE\_

Сделай реальное селфи и получи +100 баллов

Участуй в розыгрыше крутого гироборда

Все участники Мой друзья Победители

Almack77 20.05.2016 KEEP CALM MOVE

LilWayne 20.05.2016 KEEP CALM MOVE

GRIFDESAN 20.05.2016 KEEP CALM MOVE

GRIFDESAN 20.05.2016 KEEP CALM MOVE

Mayra 21.05.2016 KEEP CALM MOVE

Mayra 21.05.2016 KEEP CALM MOVE

Pavlik99 21.05.2016 KEEP CALM MOVE

DAMA 21.05.2016 KEEP CALM MOVE

DAMA 21.05.2016 KEEP CALM MOVE

Moli 21.05.2016 KEEP CALM MOVE

VASTEAH952 21.05.2016 KEEP CALM MOVE

Официальный партнер конкурса

ТЕМЕКІ ШЕГУ ТЕРІНІ ЕРТЕ КАРТАЙТАДЫ. КУРЕНІЕ ВЫЗЫВАЕТ ПРЕЖДЕВРЕМЕННОЕ СТАРЕНИЕ КОЖИ

Mevius Admin

Walter

Search... Q

- Страны
- Пользователи
- Магазин
- Акции
- Эффекты
- Галерея
- Ред. уведомлений
- Настройки
- Обратная связь
- Жалобы
- Отчеты

## Статистика

Количество фотографий помещено

Период	Количество фотографий
2015 Сентябрь	~2500
2015 Октябрь	~4000
2015 Ноябрь	~4500
2015 Декабрь	~5000
2016 Январь	~6500
2016 Февраль	~5500
2016 Март	~4500
2016 Апрель	~7000
2016 Май	~6500
2016 Июнь	~8500

### Авторизация

- Успешные авторизации: Количество уникальных успешных авторизаций на сайте.
- Неуспешные: Количество уникальных неуспешных авторизаций на сайте.
- Авторизация-через приложение: Количество уникальных логинов через приложение.

### Регистраций

- Полные регистрации: Всего зарегистрировалось на сайте, пройдя полную регистрацию.
- Регистрации на MeviusApp: Количество пользователей которые зарегистрировались, используя MeviusApp.
- Импортированные по 1:1: Пользователи, которые были импортированы из CRM через MeviusApp.

### MeviusApp activity - Users

- Пользователи с минимум 1 фото за весь период: Уникальных пользователей, который загрузил минимум 1 фото.
- Пользователи с минимум 1 фото через App за весь период: Уникальных пользователей, который загрузил минимум 1 фото через App.
- Пользователи с минимум 1 загруженным 1 фото через за весь период: Уникальных пользователей, у которых было загружено минимум 1 фото.

### MeviusApp activity - Photos

- Фотографии загружено/загружено всего: Количество загруженных фотографий.
- Всего фотографий загружено/загружено всего через сайт: Количество загруженных фотографий через сайт.
- Всего фотографий загружено/загружено всего через App: Количество загруженных фотографий через App.

### Contest activity

- Всего/загружено конкурсных фото: Количество загруженных конкурсных фотографий.
- Акционные фотографии загружено/загружено всего через сайт: Количество загруженных фотографий с любым из акционных фильтров.
- Акционные фотографии загружено/загружено всего через App: Количество загруженных фотографий с любым из акционных фильтров через App.

Affiliates