

**Инструкция по эксплуатации
программного обеспечения "Модуль
Статистика"**

Оглавление

1. Описание и стек технологий	3
2. Запуск ПО Модуль Статистика	4
3. Функции модуля	5
3.1. Добавление статистики просмотров превью объектов в буфер	5
3.2. Добавление статистики просмотров объектов в буфер	6
3.3. Добавление статистики просмотров автора в буфер	7
3.4. Разбор буфера превью объектов	8
3.5. Разбор буфера просмотра объектов	8
3.6. Разбор буфера просмотра авторов	8
3.7. Удаление использованных данных буфера	8
3.8. Вывод статистики автора	8
3.9. Крон скрипты	9
3.9.1. Скрипт разбора буфера превью объектов	9
3.9.2. Скрипт разбора буфера просмотра объектов	9
3.9.3. Скрипт разбора буфера просмотра авторов	9
3.9.4. Скрипт удаления использованных данных буфера	9
3.10. Последовательность выполнения операций	10

1. Описание и стек технологий

Сервис представляет собой скачиваемый архив с необходимым набором файлов.

Версии ПО:

- Nginx 1.10.3
- Apache 2.4
- PHP 7.0.33
- Percona Server MySQL 5.7.32
- Конфигурации подключения к БД хранятся в глобальных include по пути `/etc/php/7.0/include/`

2. Запуск ПО Модуль Статистика

Для запуска достаточно перенести файлы из архива “statistics.zip” в заранее настроенное окружение и создать необходимые таблицы в БД.

Для работы потребуется создать таблицы в БД MySQL из файла DDL_tables.md, который находится в том же архиве. Для вашей системы измените настройки в файле settings.mysql.php согласно требуемым вам настройкам.

3. Функции модуля

3.1. Добавление статистики просмотров превью объектов в буфер

Параметры:

- ajax: `"/analytics/savePreviewStatistics"`

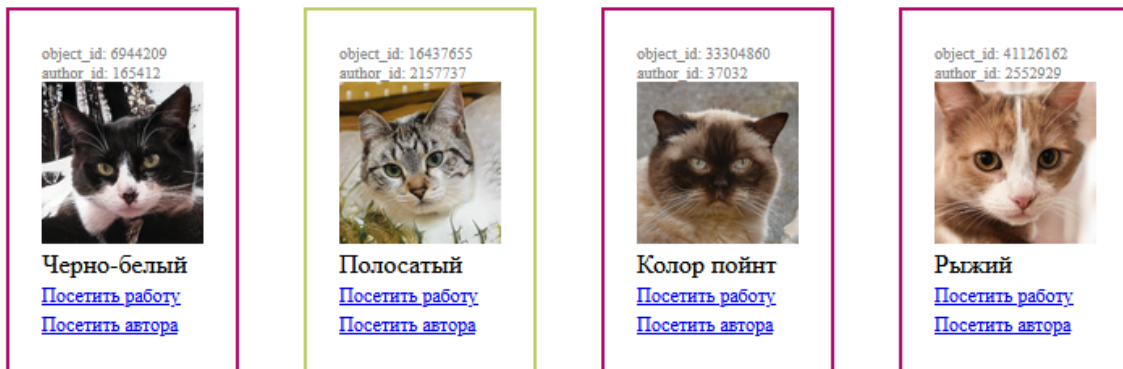
Параметры POST:

- items - двумерный массив (`[[userId => 0, itemId => 0]]`) данных об объекте статистики
- zone - зона статистики

Пример обращения с помощью AJAX:

```
const savePreviewStatistics = (event) => {  
  const xhr = new XMLHttpRequest();  
  const viewedItems = document.querySelectorAll('[viewed=true]');  
  const zonePicker = document.querySelector('#zone-picker');  
  const viewedItemsData = new FormData();  
  const itemsArr = [];  
  viewedItems.forEach(item => {  
    itemsArr.push({itemId: item.dataset.objectId, userId:  
item.dataset.authorId});  
  });  
  viewedItemsData.append('items', JSON.stringify(itemsArr));  
  viewedItemsData.append('zone', zonePicker.value);  
  xhr.open('POST', '/analytics/savePreviewStatistics');  
  xhr.send(viewedItemsData);  
  xhr.onload = xhr.onerror = () => {  
    console.log(xhr.response);  
    location.reload();  
  }  
};
```

Демонстрационный пример:



3.2. Добавление статистики просмотров объектов в буфер

Параметры:

- ajax: `"/analytics/saveViewStatistics"`

Параметры POST:

- `userId` - идентификатор пользователя, кому принадлежит
- `itemId` - идентификатор объекта
- `zone` - зона статистики
- `currentUserId` - идентификатор текущего пользователя для запрета накрутки своей статистики

Пример обращения с помощью AJAX:

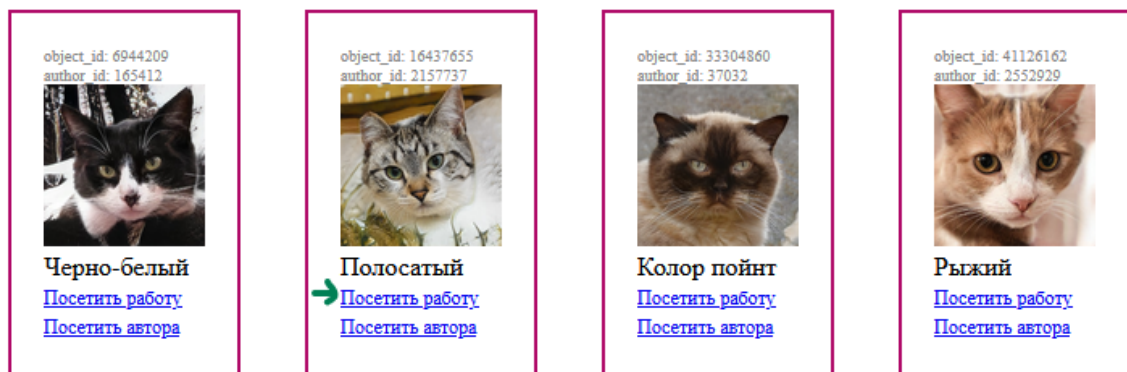
```
const saveViewStatistics = (event) => {  
  const xhr = new XMLHttpRequest();  
  const viewedItem = event.currentTarget;  
  const viewerId = document.querySelector('#user-id').value;  
  const zonePicker = document.querySelector('#zone-picker');  
  const viewedItemData = new FormData();  
  viewedItemData.append('itemId',  
viewedItem.parentElement.dataset.objectId);  
  viewedItemData.append('userId', viewerId);  
  viewedItemData.append('zone', zonePicker.value);  
  xhr.open('POST', '/analytics/saveViewStatistics');  
  xhr.send(viewedItemData);  
  xhr.onload = xhr.onerror = () => {
```

```

        console.log(xhr.response);
        location.reload();
    }
};

```

Демонстрационный пример:



3.3. Добавление статистики просмотров автора в буфер

Параметры:

- ajax: `"/analytics/saveAuthorViewStatistics"`

Параметры POST:

- `userId` - идентификатор гостя
- `authorId` - идентификатор автора

Пример обращения с помощью AJAX:

```

const saveAuthorViewStatistics = (event) => {
    const xhr = new XMLHttpRequest();
    const viewedItem = event.currentTarget;
    const viewerId = document.querySelector('#user-id').value;
    const viewedItemData = new FormData();
    viewedItemData.append('userId', viewerId);
    xhr.open('POST', '/analytics/saveAuthorViewStatistics');
    viewedItemData.append('authorId',
viewedItem.parentElement.dataset.authorId);
    xhr.send(viewedItemData);
    xhr.onload = xhr.onerror = () => {
        console.log(xhr.response);
    }
}

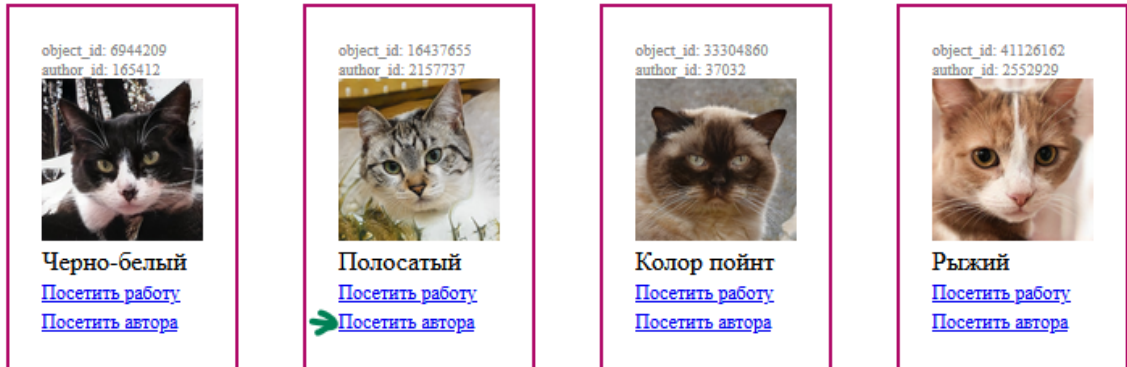
```

```

        location.reload();
    }
};

```

Демонстрационный пример:



3.4. Разбор буфера превью объектов

Описание см. п. 2.9.1

- крон: "cron/cron_service_stat_update_objects_preview_data_lock.php"

3.5. Разбор буфера просмотра объектов

Описание см. п. 2.9.2

- скрипт: "cron/cron_service_stat_update_objects_data_lock.php"

3.6. Разбор буфера просмотра авторов

Описание см. п. 2.9.3

- скрипт: "cron/cron_service_stat_update_authors_data_lock.php"

3.7. Удаление использованных данных буфера

Описание см. п. 2.9.4

- скрипт: "cron/cron_service_stat_delete_used_data_lock.php"

3.8. Вывод статистики автора

Параметры:

- ajax: "/analytics/author"

Пример реализации:

```

const getAuthorStat = (event) => {
    const xhr = new XMLHttpRequest();
    const authorIdSelect = document.querySelector('#select-author-id');;

```

```

const periodSelect = document.querySelector('#select-period');;
const authorData = new FormData();
const resultTextarea = document.querySelector('#author-stat');
authorData.append('author_id', authorIdSelect.value);
authorData.append('period', periodSelect.value);
xhr.open('POST', '/analytics/author');
xhr.send(authorData);
xhr.onload = xhr.onerror = () => {
    console.log(xhr.response);
    resultTextarea.value = xhr.response;
    resultTextarea.value +=
explainStatJson(JSON.parse(xhr.response))
}
};

```

3.9. Крон скрипты

В обычных условиях скрипты рекомендуется установить на исполнение каждую минуту с блокировкой двойного запуска: например, воспользоваться crontab.

3.9.1. Скрипт разбора буфера превью объектов

Выполняет получение данных превью объектов из буфера и запись в таблицы ежедневной и общей статистики

3.9.2. Скрипт разбора буфера просмотра объектов

Выполняет получение данных просмотров объектов из буфера и запись в таблицы ежедневной и общей статистики

3.9.3. Скрипт разбора буфера просмотра авторов

Выполняет получение данных просмотра авторов из буфера и запись в таблицы ежедневной и общей статистики

3.9.4. Скрипт удаления использованных данных буфера

Выполняет удаление данных из буфера, если они были полностью использованы скриптами 2.9.1-2.9.3

3.10. Последовательность выполнения операций

1. Сбор статистики
2. Выполнение крон-скриптами агрегации данных
3. Вывод агрегированных данных