

Логирование действий пользователя веб-системе

Акентьев Данила Денисович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной работе рассматривается логирование действий пользователя в веб-системе на основе фреймворка Yii2. Основное внимание уделяется созданию и настройке системы логирования, обеспечению безопасности данных и улучшению пользовательского интерфейса с использованием фреймворка Materialize. Разработанная система позволяет эффективно отслеживать и анализировать действия пользователей, что способствует улучшению функциональности и безопасности веб-приложений.

Ключевые слова: Логирование, веб-система, Yii2, PHP, безопасность данных, пользовательский интерфейс, Materialize, SQL-инъекции.

Logging user actions in the web system

Akentev Danila Denisovich

Sholom-Aleichem Priamursky State University

Student

Abstract

In this paper, we consider logging user actions in a web system based on the Yii2 framework. The main focus is on creating and configuring a logging system, ensuring data security and improving the user interface using the Materialize framework. The developed system allows you to effectively monitor and analyze user actions, which helps to improve the functionality and security of web applications.

Keywords: Logging, Web system, Yii2, PHP, Data security, User interface, Materialize, SQL injection.

1 Введение

1.1 Актуальность

Актуальность исследования обусловлена необходимостью обеспечения безопасности и контроля за действиями пользователей в веб-приложениях. В современных условиях веб-сайты становятся все более сложными и многофункциональными, что требует эффективных инструментов для мониторинга и анализа пользовательской активности. Логирование действий пользователей позволяет не только повысить уровень безопасности, но и улучшить качество обслуживания, анализируя поведение пользователей и устраняя потенциальные проблемы.

1.2 Обзор исследований

М.Е. Кочитов рассмотрел в своей статье применение пагинации на содержимое вебсайтов для разбивки его на страницы, используя язык разметки HTML, язык программирования PHP и готовый шаблон стилей CSS от Materialize [1]. Показал и разработал собственный пример про использование плагина ActiveForm для отображения на веб-странице интерактивных форм в PHP-фреймворке Yii2, автор статьи М.Е. Кочитов [2]. Про использование виджетов GridView рассказал М.Е. Кочитов в своей статье [3]. В.В. Бабенко, И.А. Красникова, А.С. Ленников, А.С. Андропов рассмотрели обоснования внедрения системы логирования, преимущества и недостатки ее разработки [4]. Разработали систему регистрации действий пользователей в информационной системе здравоохранения такие авторы как О. Усатова, Ш. Макиленов, С. Аманжолова, С. Диханбаев, Н. Усатов [5].

1.3 Цель исследования

Целью исследования является разработка и внедрение системы логирования действий пользователей в веб-системе на базе фреймворка Yii2. Это позволит отслеживать и анализировать действия пользователей, обеспечивая высокий уровень безопасности данных и удобство в управлении веб-приложением.

2 Материалы и методы

Для достижения поставленных целей использовались следующие методы и инструменты:

- Фреймворк Yii2 для разработки веб-системы.
- PHP для написания серверной логики.
- HTML и CSS, с использованием Materialize для создания пользовательского интерфейса.
- Методы защиты от SQL-инъекций для обеспечения безопасности.
- Тестирование и отладка кода на различных этапах разработки.

3 Результаты и обсуждения

Для отслеживания действий совершенными пользователями сайта было добавлено логирование. В первую очередь создана таблица «log», которая будет содержать следующие столбцы (рис.1).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id_log	int			Нет	Нет		AUTO_INCREMENT	Ещё ▾
2	id_user	int			Да	NULL			Ещё ▾
3	datez	datetime			Да	NULL			Ещё ▾
4	action	varchar(1000)	utf8mb3_general_ci		Нет	Нет			Ещё ▾
5	category	varchar(255)	utf8mb3_general_ci		Да	NULL			Ещё ▾

Рисунок 1 – Структура таблицы «log»

Далее для добавления в эту таблицу записей понадобился контроллер, который был назван «UserActionLoggerController.php», а также написана функция обрабатывающая запись переменных в таблицу «log» (рис.2).

```
<?php

namespace app\controllers;

use Yii;
use yii\web\Controller;

class UserActionLoggerController extends Controller
{
    public function actionLog($userId, $action, $category)
    {
        // Запись в базу данных лога действия пользователя
        $date = date('Y-m-d H:i:s');
        Yii::$app->db->createCommand()->insert('log', [
            'id_user' => $userId,
            'datez' => $date,
            'action' => $action,
            'category' => $category,
        ]->execute();

        // Вывод каких-то других действий, если нужно
    }
}
```

Рисунок 2 – Код контроллера «UserActionLoggerController.php»

После чего в каждый обработчик действий, который нужно отслеживать, был добавлен следующий код (рис.3). На примере поисковой системы добавленной ранее.

```
// Логируем действие пользователя
$userId = Yii::$app->user->id; // Получаем ID текущего пользователя
$action = 'Поиск'; // Название действия
$category = 'site/search'; // Категория действия

// Формируем строку с данными для сохранения
$searchData = "Поиск по названию: $data, по студенту: $data1, по date: $data2";

// Логируем действие пользователя одной записью
$userActionLoggerController = new UserActionLoggerController(Yii::$app->controller->id, Yii::$app);
$userActionLoggerController->actionLog($userId, $action . ' ' . $searchData, $category);
```

Рисунок 3 – Код для логирования действий функции

Для удобства администратору при отслеживании логов. Был создан отдельный интерфейс «Логи» (рис.4). В данный интерфейс входит таблица с логами и поиск по ним.


```

public function actionLog($iduser='', $category='', $dateLog='')
{
    $data = $iduser;
    $data1 = $category;
    $data2 = $dateLog;

    //////////////// Инъекции SQL
    $data = trim($data);
    $data1 = trim($data1);
    $data2 = trim($data2);

    $white = ['ALTER', "DATABASE", "BETWEEN", "CASE", "CHECK", "DATABASE", "INDEX", "REPLACE", "PROCEDURE", "UNIQUE", "VIEW", "DEFAULT", "DELETE", "DESC", "DISTINCT"];
    foreach($white as $key)
    {
        $b1 = [strtoupper($data), strtoupper($data1), strtoupper($data2)];
        if(str_contains($b1[0], $key) || str_contains($b1[1], $key) || str_contains($b1[2], $key)) $this->redirect('?r=site/search');
    }

    static $pattern = "/[!@#%$^&*,?~_]{\v+|\(\)\*\|\;\|\}\|\|\<|>|'"/;
    if(preg_match($pattern, $data, $matches, PREG_OFFSET_CAPTURE) || preg_match($pattern, $data1, $matches, PREG_OFFSET_CAPTURE)) $this->redirect('?r=site/search');
    ////////////////

    $query = Log::find();
    if(!empty($data)) $query->andWhere(['LIKE', 'id_user', '%'. $data. '%', false]);
    if(!empty($data1)) $query->andWhere(['LIKE', 'category', '%'. $data1. '%', false]);
    if(!empty($data2)) $query->andWhere(['LIKE', 'datez', '%'. $data2. '%', false]);

    $models = $query->all();

    \Yii::$app->getView()->params['iduser'] = $data;
    \Yii::$app->getView()->params['category'] = $data1;
    \Yii::$app->getView()->params['datez'] = $data2;

    $dataProvider = new ArrayDataProvider([
        'allModels' => $models, // Массив моделей для отображения
        'pagination' => [
            'pageSize' => 10, // Количество записей на странице
        ],
    ]);

    return $this->render('log', [
        'pages' => $pagination,
        'models' => $models,
        'dataProvider' => $dataProvider,
    ]);
}

```

Рисунок 6 – Функция, обрабатывающая запросы поисковой системы логов

Вывод логов в виде таблицы обусловлен структурой их хранения. После проверки способов выводов опытным путем был сделан вывод, что виджет GridView идеально подходит для данной задачи (рис.7).

```

<br>
<?= GridView::widget([
    'dataProvider' => $dataProvider,
    'columns' => [
        // Колонки GridView,
        'id_log',
        'id_user',
        'datez',
        'action',
        'category',
    ],
    'dataProvider' => new \yii\data\ArrayDataProvider([
        'allModels' => $models,
        'pagination' => [
            'pageSize' => 10,
        ],
        'sort' => [
            'attributes' => ['id_log', 'id_user', 'datez', 'action', 'category'], // Поля для сортировки
            'defaultOrder' => ['datez' => SORT_DESC], // Сортировка по умолчанию по полю datez в обратном порядке
        ],
    ]),
]); ?>
</div>

```

Рисунок 7 – Код для вывода логов

4 Выводы

В ходе исследования была разработана и внедрена система логирования действий пользователей в веб-системе на основе фреймворка Yii2. Основные результаты включают:

- Эффективная система логирования: Созданная система позволяет отслеживать и записывать все важные действия пользователей, что

способствует улучшению контроля за функционированием веб-приложения.

- Обеспечение безопасности данных: Реализация методов защиты от SQL-инъекций и других уязвимостей обеспечивает высокий уровень безопасности данных пользователей и самого веб-приложения.
- Улучшенный пользовательский интерфейс: Использование фреймворка Materialize позволило создать современный и удобный интерфейс для администрирования логов, что облегчает анализ и управление данными.
- Практическая значимость: Разработанная система логирования может быть адаптирована для различных веб-приложений, требующих отслеживания и анализа пользовательской активности. Это делает данный подход полезным для широкого круга разработчиков и проектов, улучшая функциональность и безопасность веб-систем.

В целом, результаты исследования подтверждают, что интеграция системы логирования на базе Yii2 с использованием современных технологий и методов безопасности способствует созданию надежных и эффективных веб-приложений.

Библиографический список

1. Кочитов М.Е. Применение пагинации на содержимое вебсайтов // Постулат. 2018. Т.7. №. 33. С.27.
2. Кочитов М.Е. Использование плагина ActiveForm для отображения на веб-странице интерактивных форм в PHP фреймворке yii2 // Постулат. 2020. Т. 1. №. 51. С. 70
3. Кочитов М.Е. Использование виджетов sqlDataProvider и GridView для отображения таблицы данных в PHP фреймворке yii2 // Постулат. 2021. Т. 1. №. 63.
4. Бабенко В.В., Красников И.А., Ленников А.С. О необходимости внедрения системы логирования в автоматизированные системы технического диагностирования // Сборник докладов научно-технической конференции молодых ученых и специалистов. Санкт-Петербург. 2024. С. 20-23
5. Ussatova O., Makilenov Sh., Amanzholova S. Development of a system for logging user actions in a health information system // Вестник казахской академии транспорта и коммуникаций им. М. Тынышпаева. 2024. Т. 130. №. 1. С. 332-343.