

Техническое задание на доработку системы логирования

1. Общая информация

1.1. Цель работ

Необходимо доработать существующий WordPress-плагин, обеспечивающий обмен данными между сайтом и 1С, с целью внедрения полноценной системы логирования, хранения файлов обмена, мониторинга ошибок и уведомлений о критичных событиях.

Текущая проблема заключается в том, что при синхронизации товаров и заказов отсутствует прозрачная диагностика процессов. В результате: - периодически возникают сбои синхронизации; - товары могут некорректно создаваться или обновляться на сайте; - информация по заказам не всегда корректно выгружается обратно в 1С; - отсутствует оперативное информирование ответственных лиц о критичных ошибках; - невозможно быстро определить причину инцидента без привлечения разработчиков и ручного анализа.

Доработка должна решить указанные проблемы и создать основу для дальнейшего масштабирования интеграции, повышения отказоустойчивости и снижения стоимости сопровождения.

2. Контекст интеграции

На текущий момент: - сайт работает на WordPress; - используется интеграция с 1С; - из 1С на сайт передаются данные по товарам; - обратно в 1С передается информация по заказам; - в рамках обмена используется XML-файл; - 1 раз в сутки передается файл полной синхронизации; - 2 раза в сутки передаются файлы инкрементальной синхронизации.

Необходимо обеспечить логирование всех ключевых этапов данного обмена.

3. Постановка задачи

Разработать и внедрить в существующий WordPress-плагин отдельный модуль логирования, который должен:

1. Фиксировать все события синхронизации между WordPress и 1С.
2. Сохранять оригиналы XML-файлов, поступающих в рамках обмена.
3. Предоставить в административной панели WordPress отдельную страницу для просмотра логов.
4. Обеспечить фильтрацию логов по уровню важности и дате.
5. Позволять настраивать срок хранения логов и файлов обмена.
6. Отображать в админке суммарный размер логов.
7. Отправлять уведомления о важных событиях:
 - в чат;
 - на email;
 - с возможностью настройки минимального уровня важности для отправки уведомлений.

4. Функциональные требования

4.1. Логирование событий синхронизации

Система должна логировать все ключевые события, связанные с обменом между WordPress и 1С.

4.1.1. Обязательные события для логирования

Необходимо фиксировать как минимум следующие события: - запуск полной синхронизации; - завершение полной синхронизации; - запуск инкрементальной синхронизации; - завершение инкрементальной синхронизации; - получение XML-файла от 1С; - начало обработки XML-файла; - завершение обработки XML-файла; - создание нового товара; - обновление существующего товара; - пропуск товара с указанием причины; - ошибка при создании товара; - ошибка при обновлении товара; - начало выгрузки заказов в 1С; - завершение выгрузки заказов в 1С; - успешная выгрузка конкретного заказа; - ошибка выгрузки конкретного заказа; - системные ошибки интеграции; - ошибки валидации данных; - ошибки структуры XML; - ошибки доступа к файлам, БД, API, внешним сервисам; - уведомления о превышении лимитов, таймаутах, прерывании процесса.

4.1.2. Структура записи лога

Каждая запись лога должна содержать минимум следующие поля: - дата и время события; - уровень важности; - источник события; - тип операции; - краткое описание события; - детальное описание / текст ошибки; - идентификатор процесса обмена / сессии синхронизации; - тип синхронизации (полная / инкрементальная / выгрузка заказов); - идентификатор сущности, если применимо: - ID товара; - артикул / внешний код; - ID заказа; - ID файла обмена; - статус события; - ссылка на исходный XML-файл, если событие связано с конкретным файлом.

4.1.3. Уровни важности логов

Необходимо реализовать уровни важности логов, как минимум: - Debug; - Info; - Warning; - Error; - Critical.

Допускается использование внутренней классификации разработчика, но в интерфейсе администратора уровни должны быть понятны и однозначны.

4.2. Хранение оригиналов XML-файлов синхронизации

Необходимо сохранять оригинал каждого XML-файла, поступающего в систему в рамках обмена с 1С.

4.2.1. Требования к сохранению файлов

- сохранять оригинал файла без изменений;
- сохранять дату и время получения файла;
- сохранять тип файла:
 - полная синхронизация;
 - инкрементальная синхронизация;
- обеспечивать связь файла с соответствующей сессией обмена и логами;
- предусмотреть уникальное имя файла, исключающее перезапись;
- предусмотреть безопасное хранение файлов на сервере.

4.2.2. Требования к доступу к файлам

В административной панели должно быть возможно: - видеть список сохраненных файлов обмена; - видеть дату загрузки файла; - видеть тип синхронизации; - видеть размер файла; - открыть / скачать файл при наличии соответствующих прав доступа; - перейти из записи лога к связанному XML-файлу.

4.3. Отдельная страница логов в админке WordPress

В существующем плагине необходимо создать отдельную административную страницу для просмотра логов синхронизации.

4.3.1. Требования к интерфейсу страницы логов

На странице должно отображаться: - список событий синхронизации; - дата и время; - уровень важности; - источник / модуль; - тип синхронизации; - краткое описание; - статус; - ссылка на подробный просмотр записи; - ссылка на связанный XML-файл, если применимо.

4.3.2. Фильтры и поиск

Необходимо реализовать: - фильтр по уровню важности; - фильтр по дате / периоду; - фильтр по типу синхронизации; - поиск по тексту сообщения, ID заказа, ID товара, артикулу, внешнему коду, идентификатору сессии.

4.3.3. Детальная карточка записи лога

При открытии записи должно отображаться: - полное описание события; - технические детали ошибки; - связанные сущности; - идентификатор сессии обмена; - информация о файле обмена; - стек ошибки или служебная информация, если доступно; - дата и время; - уровень важности.

4.4. Настройки хранения и очистки логов

Необходимо реализовать в админке настройки хранения логов и файлов обмена.

4.4.1. Настраиваемые параметры

Администратор должен иметь возможность задать: - количество дней хранения логов; - количество дней хранения XML-файлов; - минимальный уровень логирования, если такая настройка допустима архитектурно; - минимальный уровень важности для отправки уведомлений в чат; - минимальный уровень важности для отправки уведомлений на email; - email-адрес или список email-адресов для уведомлений; - параметры интеграции с чатом.

4.4.2. Очистка данных

Необходимо реализовать автоматическую очистку устаревших данных: - удаление логов старше заданного количества дней; - удаление XML-файлов старше заданного количества дней; - удаление должно выполняться безопасно и по расписанию; - необходимо логировать сам факт очистки и ее результат.

4.4.3. Отображение размера логов

В административной панели необходимо отображать: - общий размер хранимых логов; - общий размер хранимых XML-файлов; - при возможности — количество записей логов и количество сохраненных файлов.

Это позволит контролировать нагрузку на хранилище и своевременно корректировать политику хранения.

4.5. Механизм уведомлений

Необходимо реализовать механизм отправки уведомлений о важных событиях.

4.5.1. Уведомления в чат

Система должна отправлять уведомления в чат при возникновении событий выше или равных уровню важности, заданному в настройках.

Требования: - возможность указать параметры интеграции с чатом в админке; - возможность включить / отключить уведомления; - возможность задать минимальный уровень важности; - отправка краткого, но информативного сообщения.

Пример состава уведомления: - дата и время; - уровень важности; - тип синхронизации; - краткое описание ошибки / события; - идентификатор сессии; - при необходимости — ID заказа / товара; - ссылка на страницу логов в админке.

4.5.2. Email-уведомления

Система должна отправлять email-уведомления при возникновении событий выше или равных уровню важности, заданному в настройках.

Требования: - возможность указать один или несколько email-адресов; - возможность включить / отключить уведомления; - возможность задать минимальный уровень важности; - отправка понятного сообщения с техническими деталями.

4.5.3. Надежность уведомлений

Необходимо предусмотреть: - логирование факта отправки уведомления; - логирование ошибок отправки уведомлений; - защиту от дублирования уведомлений в рамках одного и того же события, если это возможно без усложнения архитектуры.

5. Нефункциональные требования

5.1. Производительность

- Логирование не должно существенно замедлять процесс синхронизации.
- Сохранение XML и запись логов должны быть реализованы с учетом работы с большими файлами и большим количеством записей.
- Интерфейс просмотра логов должен поддерживать пагинацию.

5.2. Безопасность

- Доступ к странице логов и файлам должен быть только у пользователей с соответствующими правами.
- Необходимо исключить публичный доступ к сохраненным XML-файлам.
- Следует предусмотреть защиту от записи потенциально опасных данных в интерфейс без экранирования.

5.3. Надежность

- При сбое логирования основной процесс обмена не должен полностью блокироваться, если это не критично.
- При невозможности сохранить лог или файл должна фиксироваться хотя бы минимальная системная ошибка доступным способом.

5.4. Поддерживаемость

- Решение должно быть реализовано в рамках архитектуры существующего плагина.
 - Код должен быть структурирован так, чтобы в дальнейшем можно было расширить:
 - список каналов уведомлений;
 - состав логируемых событий;
 - аналитику по обменам;
 - экспорт логов.
-

6. Предполагаемая реализация

6.1. Разделы в админке

Рекомендуется реализовать в плагине следующие разделы: - **Логи синхронизации** — список и просмотр событий; - **Файлы обмена** — список сохраненных XML-файлов; - **Настройки логирования** — параметры хранения, очистки и уведомлений.

6.2. Хранение данных

На усмотрение разработчика, но с учетом надежности и удобства поддержки, допускается реализация через: - отдельные таблицы в БД WordPress для логов и метаданных файлов; - файловое хранилище для XML; - cron-задачи WordPress для очистки устаревших данных.

При выборе реализации необходимо учитывать объем данных, скорость выборки и удобство фильтрации.

7. Этапы выполнения работ

Этап 1. Аналитика и проектирование

- анализ текущего механизма обмена с 1С;
- определение точек интеграции для логирования;
- проектирование структуры логов, файлов и настроек;
- согласование перечня событий и уровней важности.

Этап 2. Реализация серверного логирования

- внедрение механизма записи логов;
- внедрение идентификаторов сессий обмена;
- логирование операций по товарам, заказам и файлам;
- обработка ошибок и системных исключений.

Этап 3. Реализация хранения XML-файлов

- сохранение оригиналов XML;
- привязка файлов к сессиям и логам;
- реализация безопасного доступа к файлам.

Этап 4. Реализация интерфейса в админке

- создание страницы логов;
- создание страницы файлов обмена;
- реализация фильтров, поиска, пагинации и детального просмотра.

Этап 5. Реализация настроек хранения и очистки

- создание страницы настроек;
- настройка сроков хранения;
- реализация автоматической очистки;
- вывод размера логов и файлов.

Этап 6. Реализация уведомлений

- интеграция с чатом;
- интеграция с email;
- настройка порогов уведомлений;
- логирование отправки уведомлений.

Этап 7. Тестирование и сдача

- функциональное тестирование логирования;
- тестирование полной и инкрементальной синхронизации;
- тестирование выгрузки заказов;
- тестирование уведомлений;
- тестирование очистки и хранения;
- подготовка инструкции для администратора.

8. Ожидаемый результат

По итогам выполнения работ заказчик должен получить: - прозрачную систему логирования обмена между WordPress и 1С; - возможность быстро находить причины ошибок синхронизации; - сохраненные оригиналы XML-файлов для последующего анализа; - удобный интерфейс просмотра логов в админке WordPress; - автоматическую очистку устаревших данных; - контроль объема занимаемого логами пространства; - оперативные уведомления о критичных проблемах в чат и на email.

9. Критерии приемки

Работы считаются выполненными при соблюдении следующих условий:

1. В админке WordPress создана отдельная страница логов в составе плагина.
2. На странице отображаются события синхронизации с фильтрами по уровню важности и дате.

3. Логируются ключевые этапы обмена товарами и заказами.
4. Сохраняются оригиналы XML-файлов полной и инкрементальной синхронизации.
5. Из интерфейса можно просмотреть информацию о файлах и перейти к связанным логам.
6. В настройках можно задать срок хранения логов и XML-файлов в днях.
7. Реализована автоматическая очистка устаревших логов и файлов.
8. В админке отображается размер логов и файлов.
9. Реализованы уведомления в чат и на email с настраиваемым порогом уровня важности.
10. Все ключевые действия по логированию, очистке и отправке уведомлений работают стабильно и документированы.

10. Дополнительно рекомендуется

В рамках текущей задачи можно ограничиться базовым функционалом, однако архитектуру желательно заложить с возможностью дальнейшего развития. В перспективе это позволит без существенной переработки добавить: - дашборд по состоянию обменов; - статистику по ошибкам за период; - экспорт логов в CSV/Excel; - ручной повтор отправки уведомлений; - ручной повтор обработки конкретного файла обмена; - интеграцию с внешними системами мониторинга.

Такой подход повысит ценность доработки не только как инструмента диагностики, но и как полноценного контура контроля интеграции, что особенно важно при росте ассортимента, заказов и нагрузки на обмен с 1С.