

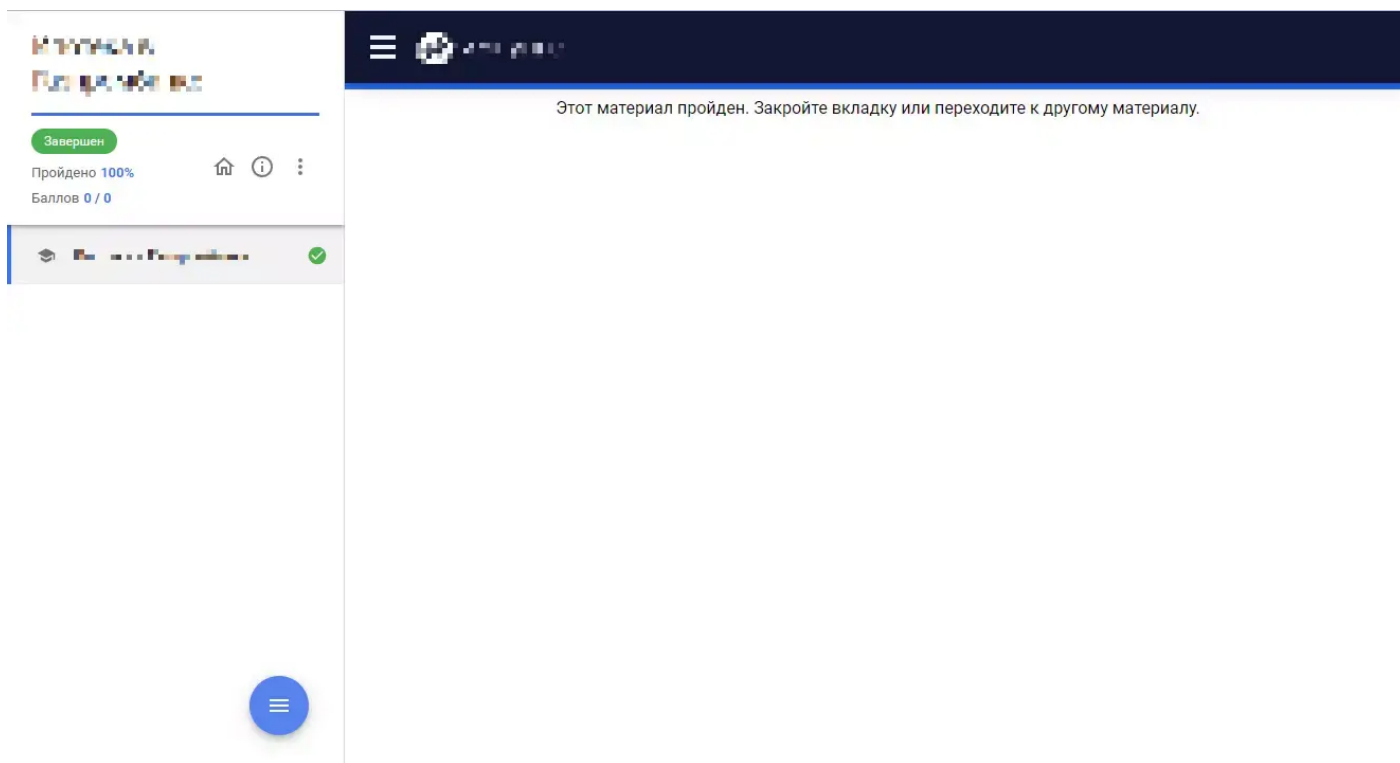
WebTutor

- Корректно завершаем курсы Scroll в Webtutor
- Отключение принудительной загрузки файла из ресурсов базы

Корректно завершаем курсы Scroll в Webtutor

Проблема

Если вы загружали курс из Scroll в Webtutor, вы могли столкнуться с такой проблемой: слушатель проходит курс, нажимает кнопку завершения обучения, курс закрывается, и появляется финальное сообщение. При этом встроенный плеер курсов WebTutor остаётся открытым, хотя это от него уже не нужно.



К чему хотим прийти

Чаще всего при нажатии на кнопку завершения курса должно происходить одно из следующих событий:

- Заккрытие курса вместе с плеером WT с переходом на учебный портал;
- Заккрытие курса вместе с плеером WT с переходом по заданной ссылке;

- Переход на следующий модуль обучения (если структура курса состоит из нескольких модулей).

Решение

1. Публикуем курс из Scroll и распаковываем полученный SCORM. Ищем папку «**js**». Там открываем файл **scorm_2004_libs.js**.
2. Вставляем в конце файла этот скрипт

```
// Функция, которая будет вызываться при появлении нужного текста
function handleTextChange(mutationsList, observer) {
  for (let mutation of mutationsList) {
    if (mutation.type === 'childList') {
      // Проверяем каждое добавление узла в DOM
      mutation.addedNodes.forEach(node => {
        // Проверяем, содержит ли добавленный узел нужный текст
        if (node.textContent.includes('Этот материал пройден. Закройте вкладку или
переходите к другому материалу.')) {
          // Код для запуска другого скрипта
          console.log("Найден нужный текст! Запускаем другой скрипт.");
        }
      });
    }
  }
}

// Создаем новый экземпляр MutationObserver с указанной функцией обратного вызова:
const observer = new MutationObserver(handleTextChange);

// Опции наблюдения: следим за изменениями внутри элемента body и его потомков:
const observerOptions = {
  childList: true,
  subtree: true
};

// Начинаем наблюдение за изменениями в DOM с указанными опциями:
observer.observe(document.body, observerOptions);
```

3. Вместо комментария «**// Код для запуска другого скрипта**» из кода выше вставляем нужный код

```
//Переключить на следующий модуль
const playerApp = window.parent.document.querySelector('#app');
if (playerApp !== null) playerApp.__vue__.$store.dispatch('learning/goNextPart');

или

//Закрыть вкладку с курсом
const playerApp = window.parent.document.querySelector('#app');
if (playerApp !== null) playerApp.__vue__.$store.dispatch('app/close');

или

//Закрыть вкладку с курсом и сделать редирект (не забудьте прописать ссылку для редиректа)
const playerApp = window.parent.document.querySelector('#app');
if (playerApp !== null) playerApp.__vue__.$store.dispatch('app/redirect', "https://ya.ru");

или

//Завершить курс (статус «пройден» или «не пройден» будет зависеть от настроек условий
успешного прохождения. Этот вариант я пока не тестировал)
const playerApp = window.parent.document.querySelector('#app');
if (playerApp !== null) playerApp.vue.$store.dispatch('learning/finishCourse');
```

4. Упаковываем файлы обратно в ZIP-архив и скамливаем его вашему WebTutor.

Какая логика работы?

Этот скрипт отслеживает изменения в DOM-дереве страницы.

После нажатия на кнопку завершения курса шаблон Scroll выводит фразу «Этот материал пройден...». Это — триггер для скрипта, который выполняет нужное вам действие: переключает модуль, закрывает вкладку, закрывает вкладку с редиректом и т. д. (эта часть кода, которую мы выше вставили в большой скрипт).

Теперь ваши курсы будут завершаться одновременно с плеером курса WebTutor либо переключение между разными модулями будет бесшовным.

Для любознательных

- Это решение работает до тех пор, пока Scroll не добавит в кнопки возможность вставлять кастомный JS. Тогда вы можете выбросить большой скрипт и повесить на кнопку только маленькую часть с конкретным действием (закрытие вкладки, закрытие с редиректом, включение следующего модуля).
- Это решение работает только в актуальном плеере курсов WT.
- Если формулировка фразы в курсе после завершения курса изменится, в скрипте нужно также обновить фразу-триггер.
- Скрипт надо вставлять при каждом новом экспорте архива из Scroll. Удобно сделать шаблон в отдельном файле и каждый раз копировать его оттуда.

Отключение принудительной загрузки файла из ресурсов базы

Проблема

Все стандартные ссылки на материалы ресурса базы автоматически загружают файлы на ПК пользователя. Это не всегда удобно, например, когда хочется чтобы pdf файл открылся сразу в браузере на просмотр.

Решение

Если ссылка на ресурс типа /download_file.html... то припишите к ней &no_disposition=1.