

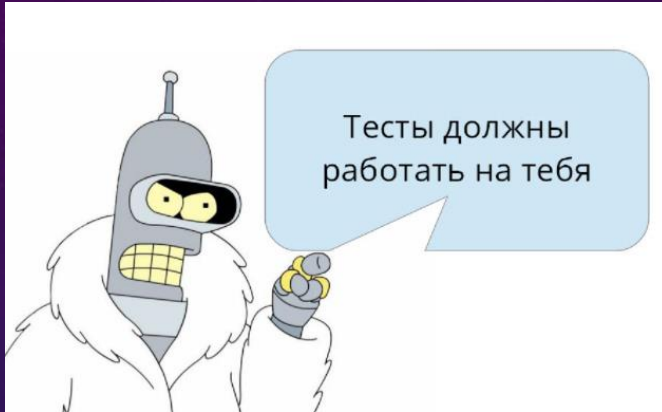
РГАТУ
ВКР на тему

«Разработка плагина системы регистрации действий
пользователя в браузере с генерацией кода для
автоматизации тестирования»

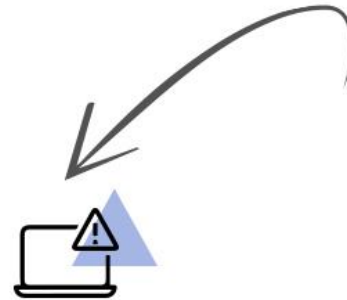
09.03.04 Программная инженерия
студент ипб-17 Куликова Э. В.
руководитель Смирнов Н. В.

2. АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ

2



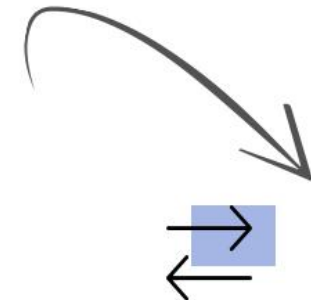
Цели автоматизации тестирования



Хотим избежать
людских ошибок



Хотим тестировать
быстрее



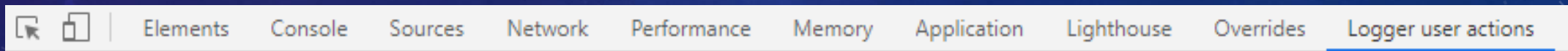
Хотим тестировать
чаще

3. ЦЕЛИ И ЗАДАЧИ

Цель — упростить процесс автоматизации тестирования, сократить время на написания автоматизированных тестов и увеличить количество тестов.

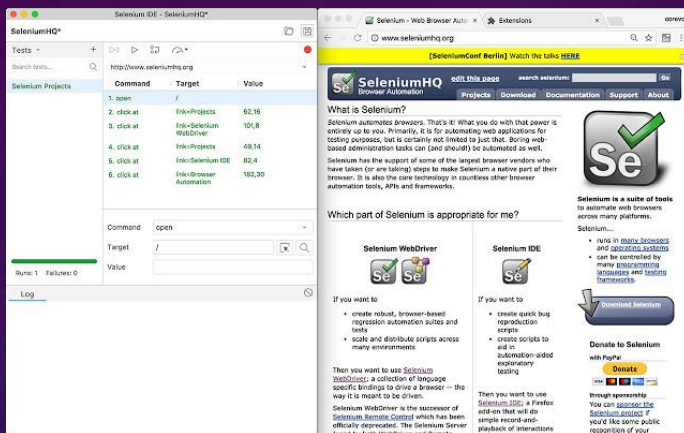
Задачи:

- Создать плагин, который будет регистрировать действия пользователей в браузере;
- Определять по каким ui-компонентам web-приложения совершено действие;
- Добавить генерацию java-кода тестов (шаблонов) для ui-компонентов.

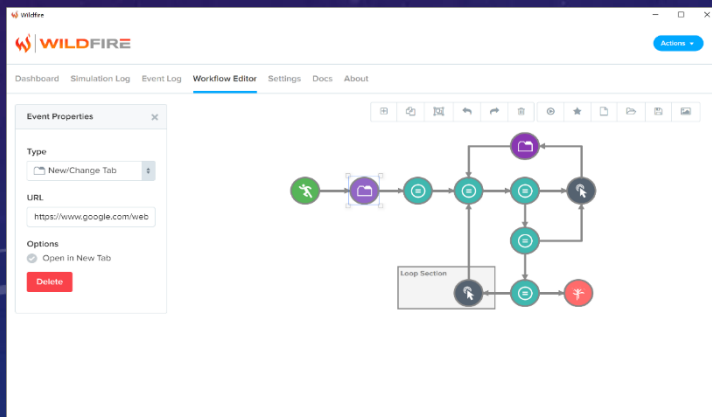


4. АНАЛОГИ

Selenium IDE



Wildfire.ai



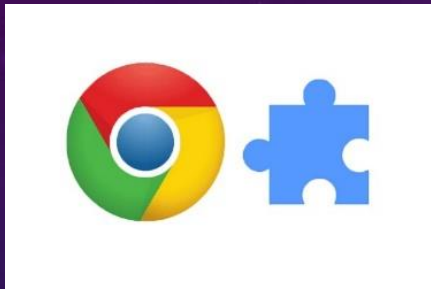
Сравнительные критерии	Selenium IDE	Wildfire.ai
Поддерживает браузеры Google Chrome и Firefox	+	+
Записывает взаимодействия пользователя с веб-страницей	+	+
Запись и воспроизведение теста	+	+
Сохранение и загрузка скриптов для последующего воспроизведения	+	+
Возможность просматривания и моделирования журнала симуляции	—	+
Использование инструмента Selenium	+	+
Генерация кода теста	—	—

5. ОСНОВНЫЕ ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

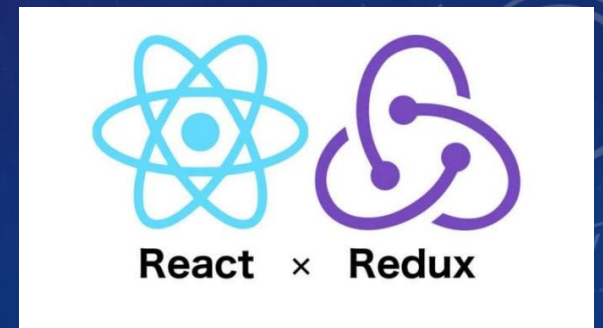
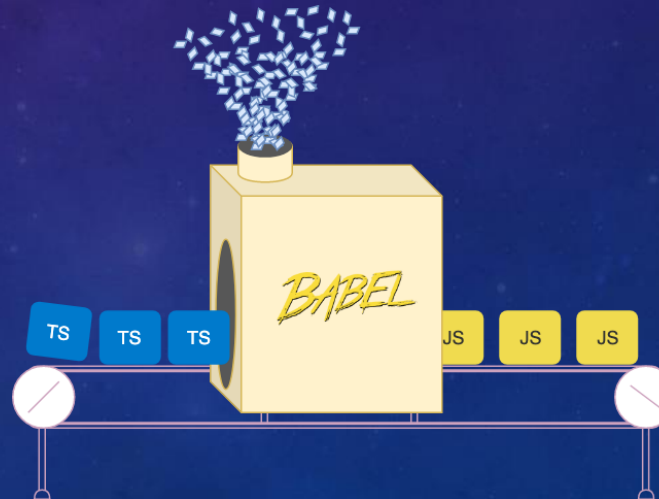
- Регистрировать действия пользователей при работе в веб-приложении на странице браузера;
- Анализировать и обрабатывать полученные события с веб-страницы приложения;
- Определять с какими ui-компонентами и элементами веб-приложения взаимодействовал пользователь;
- Сохранять собранные данные о событиях в клиентское хранилище IndexedDB для последующей с ними работы;
- Превращать собранные действия в код на языке программирования java, адаптированный под корпоративную библиотеку автоматизации тестирования autotest-lib;
- Предоставлять пользователю через графический интерфейс следующие функции:
 - Начать генерацию кода автоматизированного теста;
 - Закончить генерация кода автоматизированного теста;
 - Обновить лог событий;
 - Скопировать сгенерированный код;
 - Посмотреть сгенерированный код автотеста и лог событий выбранной сессии.

6. ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ

Расширение в
браузере Google
Chrome



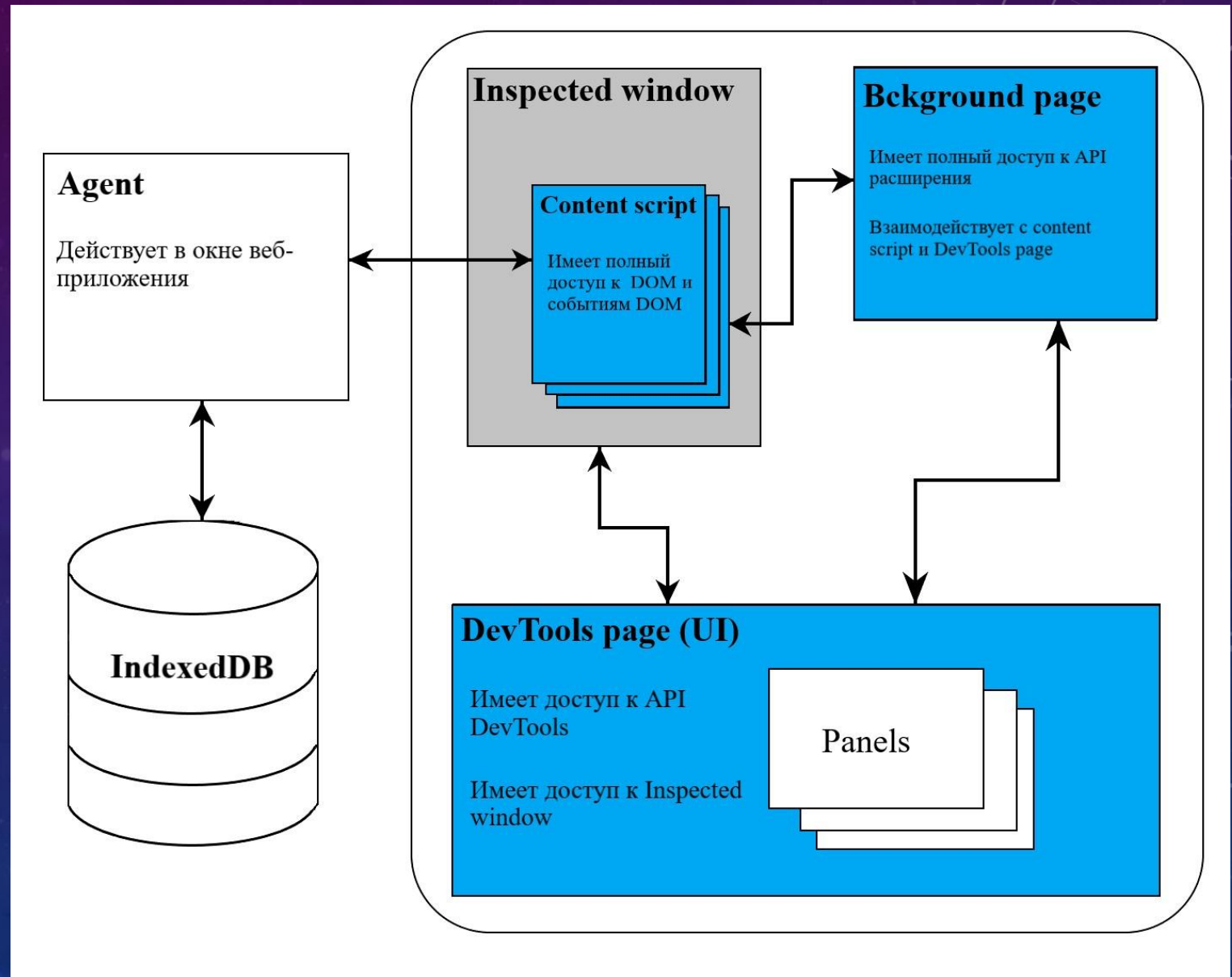
Инструменты
разработчика
DevTools



7. АРХИТЕКТУРА

Agent и DevTools общаются между собой посредством отправки сообщений через content-script и background:

Agent ↔ *content-script* ↔ *background* ↔ *UI*.

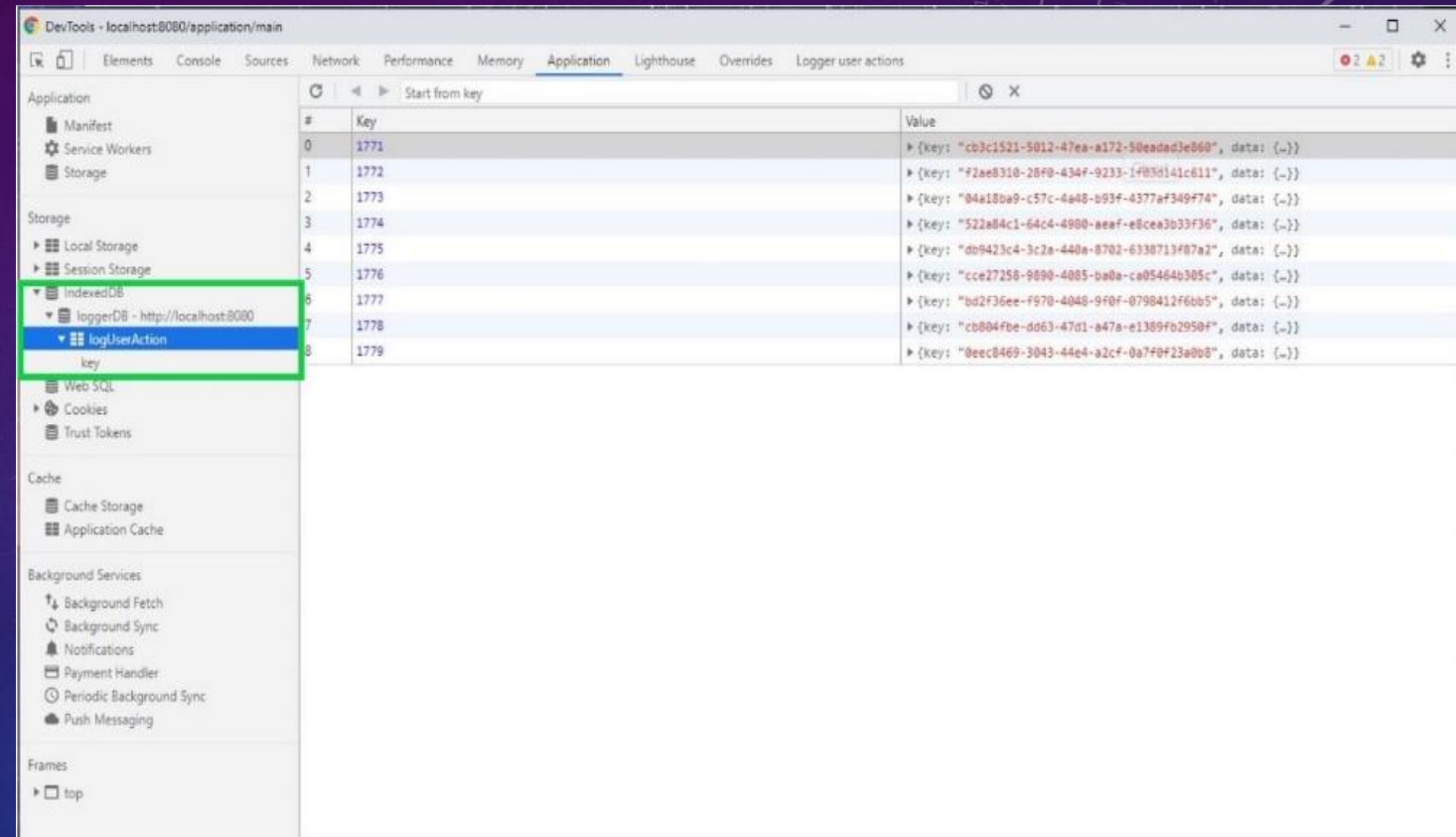


8. БАЗА ДАННЫХ INDEXEDDB

8

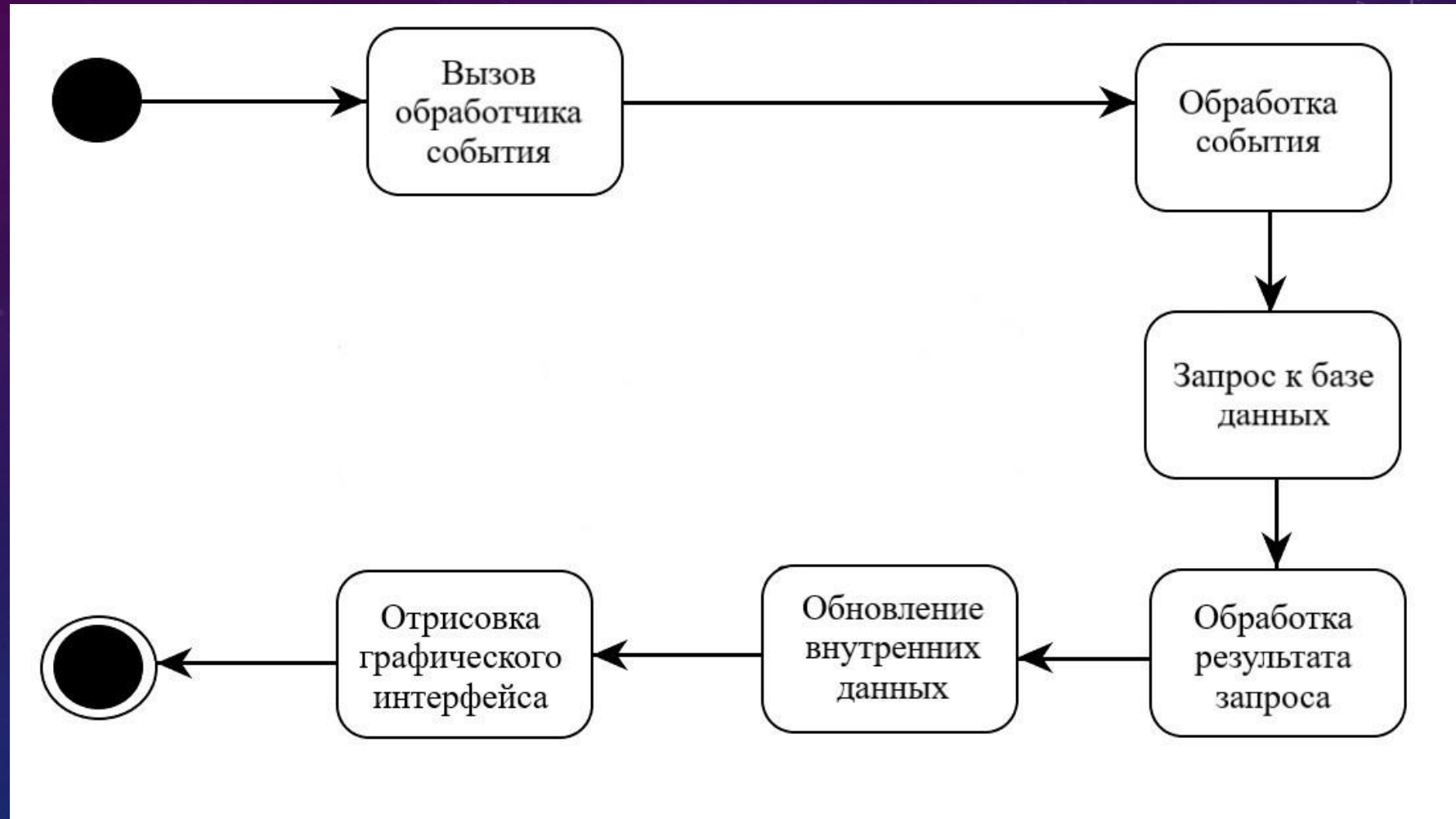
Сохранённый объект данных в IndexedDB

```
▼ {key: "04a18ba9-c57c-4a48-b93f-4377af349f74", data: {...}}
  ▼ data:
    bigButton: undefined
    date: 1620726751258
    filterCaption: "Предметный класс"
    filterToolbar: undefined
    grid: undefined
    gridColumn: undefined
    gridColumnRow: undefined
    gridToolbar: undefined
    isFilter: true
    modalWindow: undefined
    modalWindowButton: undefined
    nameForm: "Конфигурация пакетной замены"
    quickFilter: undefined
    tab: "Конфигурация пакетной замены"
    tabContainer: "TCD_Filters"
    type: "change"
    value: ""
    xpath: "/HTML[1]/BODY[1]/TABLE[1]/TBODY[1]/TR[2]/TD[1]/DIV[1]/DIV[1]/DIV..."
    key: "04a18ba9-c57c-4a48-b93f-4377af349f74"
```



После обработки действия пользователя, формируется объект данных для сохранения его в хранилище IndexedDB, чтобы потом превратить его в строки java-кода автоматизированного теста, который будет проверять UI-компонент, с которым работал пользователь.

9. АЛГОРИТМ РАБОТЫ ПРИЛОЖЕНИЯ



Общая схема работы приложения

10. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ПОЛЬЗОВАТЕЛЯ

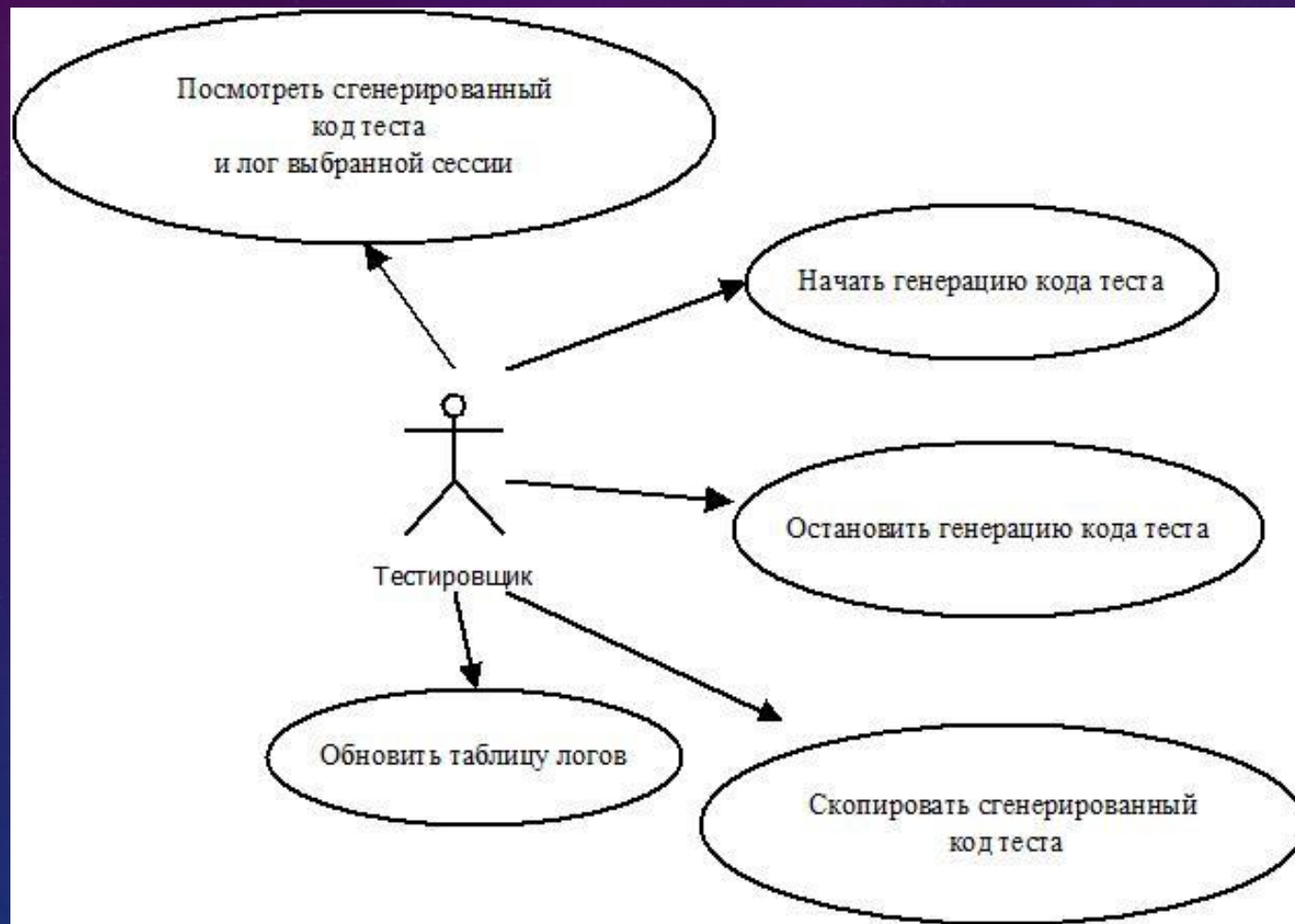


Диаграмма вариантов использования системы

11. ПРОЦЕСС РАБОТЫ ГЕНЕРАЦИИ АВТОТЕСТА

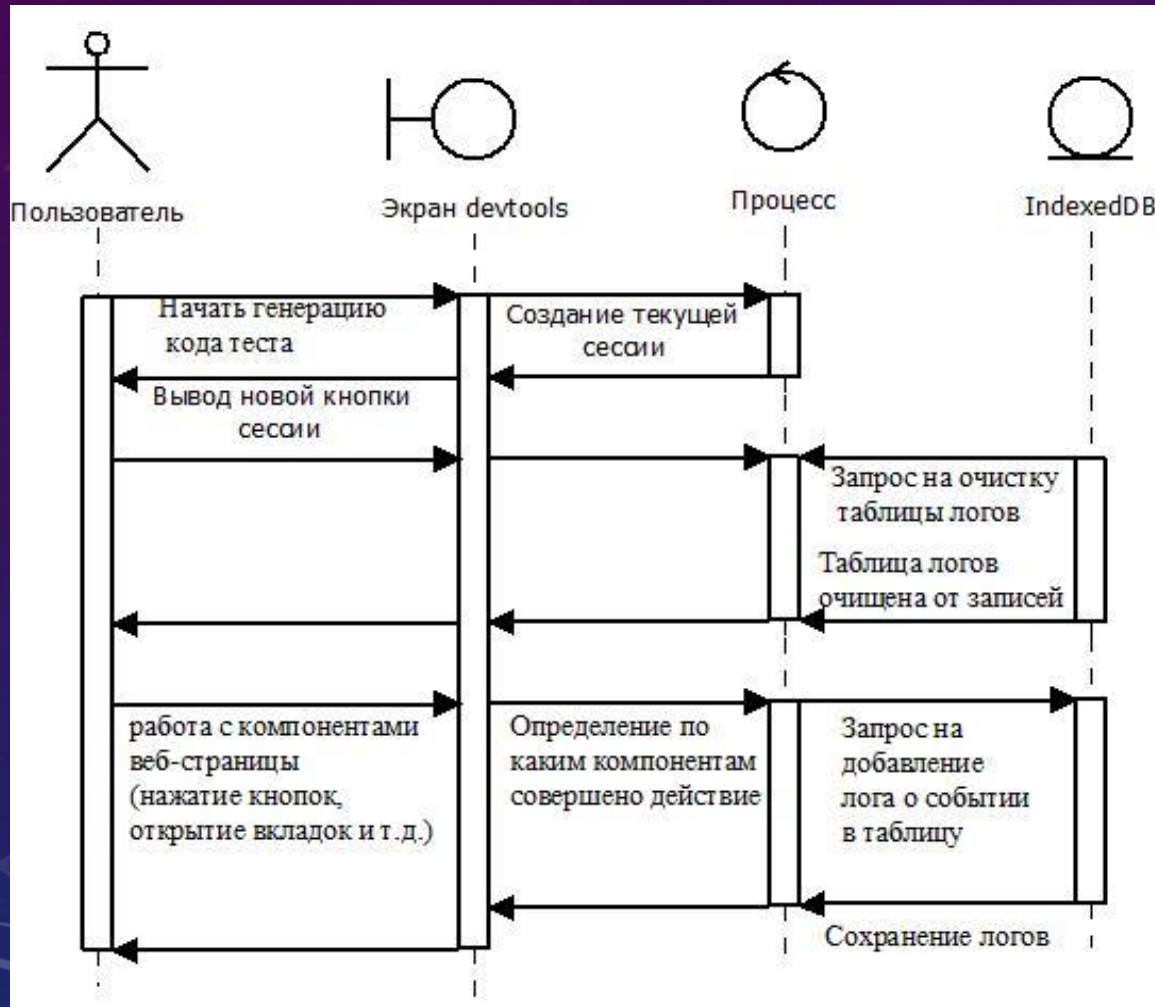


Диаграмма последовательности старта генерации автотеста

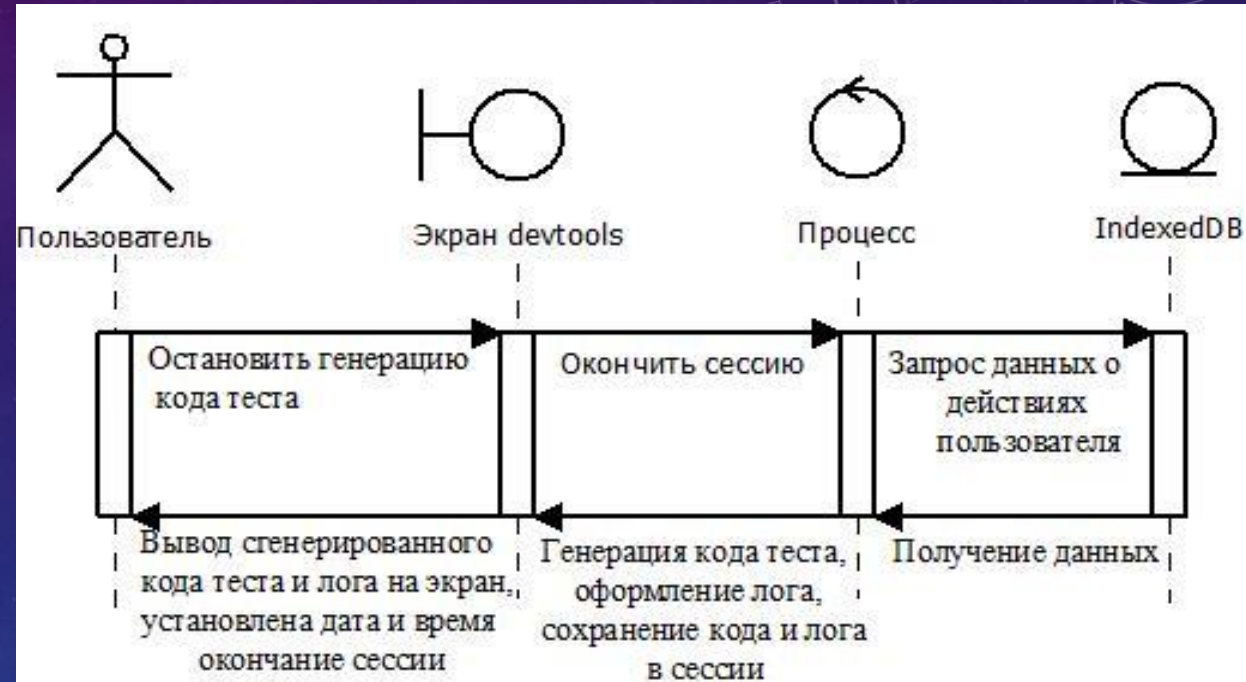


Диаграмма последовательности окончания генерации автотеста

12. ТЕСТИРОВАНИЕ ПРОГРАММЫ

The screenshot shows the Chrome DevTools interface with the 'Logger user actions' panel active. The panel is divided into three sections: 'Сессии' (Sessions), 'Код' (Code), and 'Лог' (Log).

Сессии (Sessions): This section displays a list of sessions with timestamps. The middle session is highlighted in green.

Timestamp
11-05-2021 12:46:53
11-05-2021 12:46:12
11-05-2021 12:44:51

Код (Code): This section displays the generated test script code.

```
ToolbarUtils.clickButton(form,"Детализация");
FormUtils.activateTab(mainPage,"Прикрепленные файлы");
Panel panel = form.getTabControl("TCD_Details").activateTab("Прикрепленные файлы");
ToolbarUtils.clickButton(panel,"Показать историю обработки");
Grid grid = GridUtils.getGridByTabName(form,"Прикрепленные файлы");
GridToolbox toolbox = grid.toolbox();
toolbox.refresh();
```

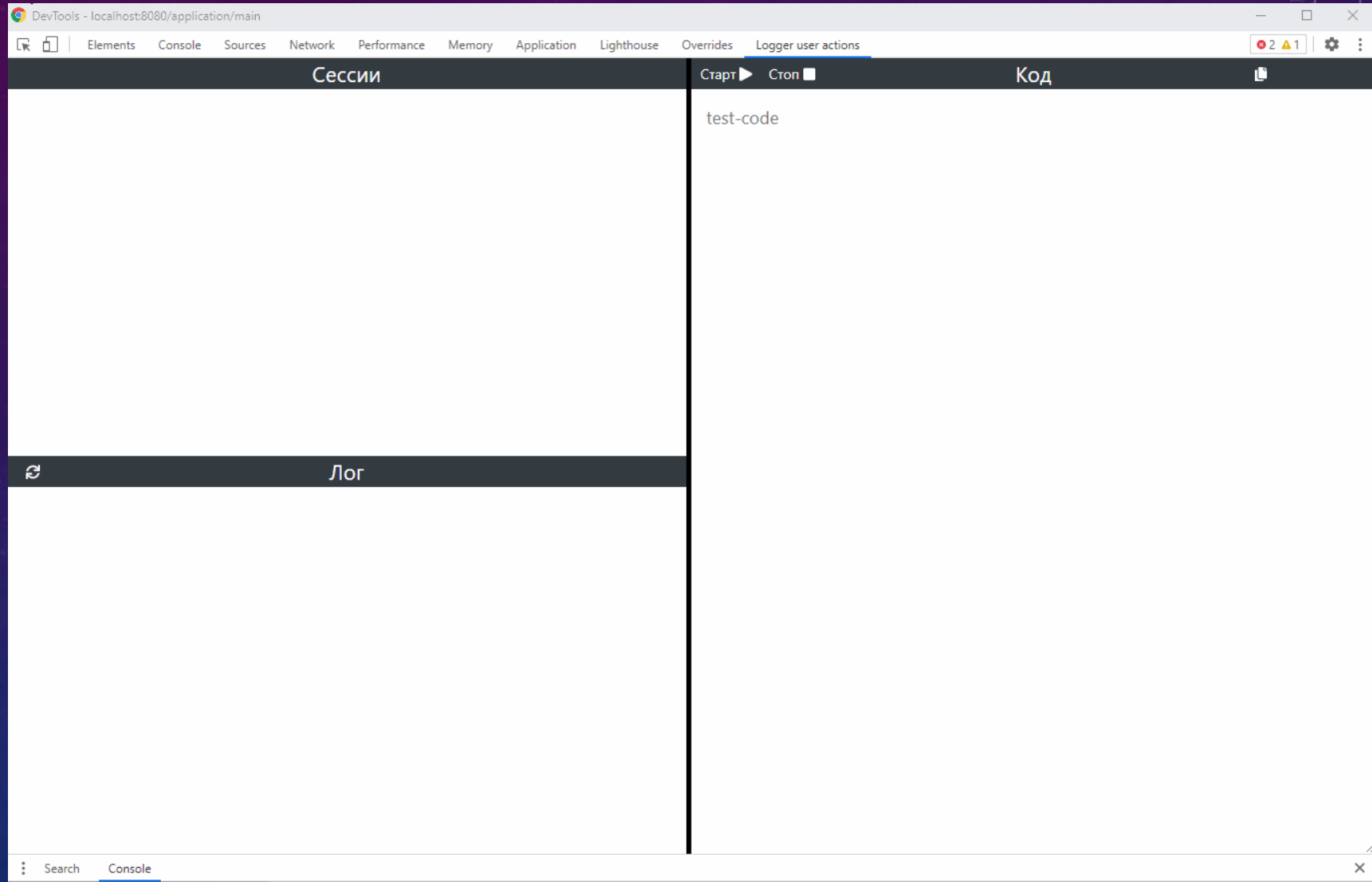
Лог (Log): This section displays a list of log entries with timestamps and descriptions.

Timestamp	Description
11-05-2021 12:45:58	Нажатие на кнопку деталей BigButton titl...
11-05-2021 12:46:02	Переход на вкладку формы Tab "Прикре...
11-05-2021 12:46:05	Клик по BigButton title = "Показать истор...
11-05-2021 12:46:09	Клик по GridToolbar title = "Обновить да...

Пример сгенерированного кода автоматизированного теста

13. ДЕМОНСТРАЦИЯ РАБОТЫ ПЛАГИНА

13



ЗАКЛЮЧЕНИЕ

- ✓ Спроектирован и разработан плагин (расширение) для браузера, который регистрирует действия пользователя в веб системах «НПО «Криста», для последующего преобразования в программный код, способный повторить эти действия в автоматизированном режиме с учетом различных тонкостей, разрабатываемых предприятием программных продуктов;
- ✓ Полученная система может использоваться специалистами предметной области для ускорения разработки функциональных автоматизированных тестов для интерфейса веб-приложений НПО «Криста».

СПАСИБО ЗА ВНИМАНИЕ!

