



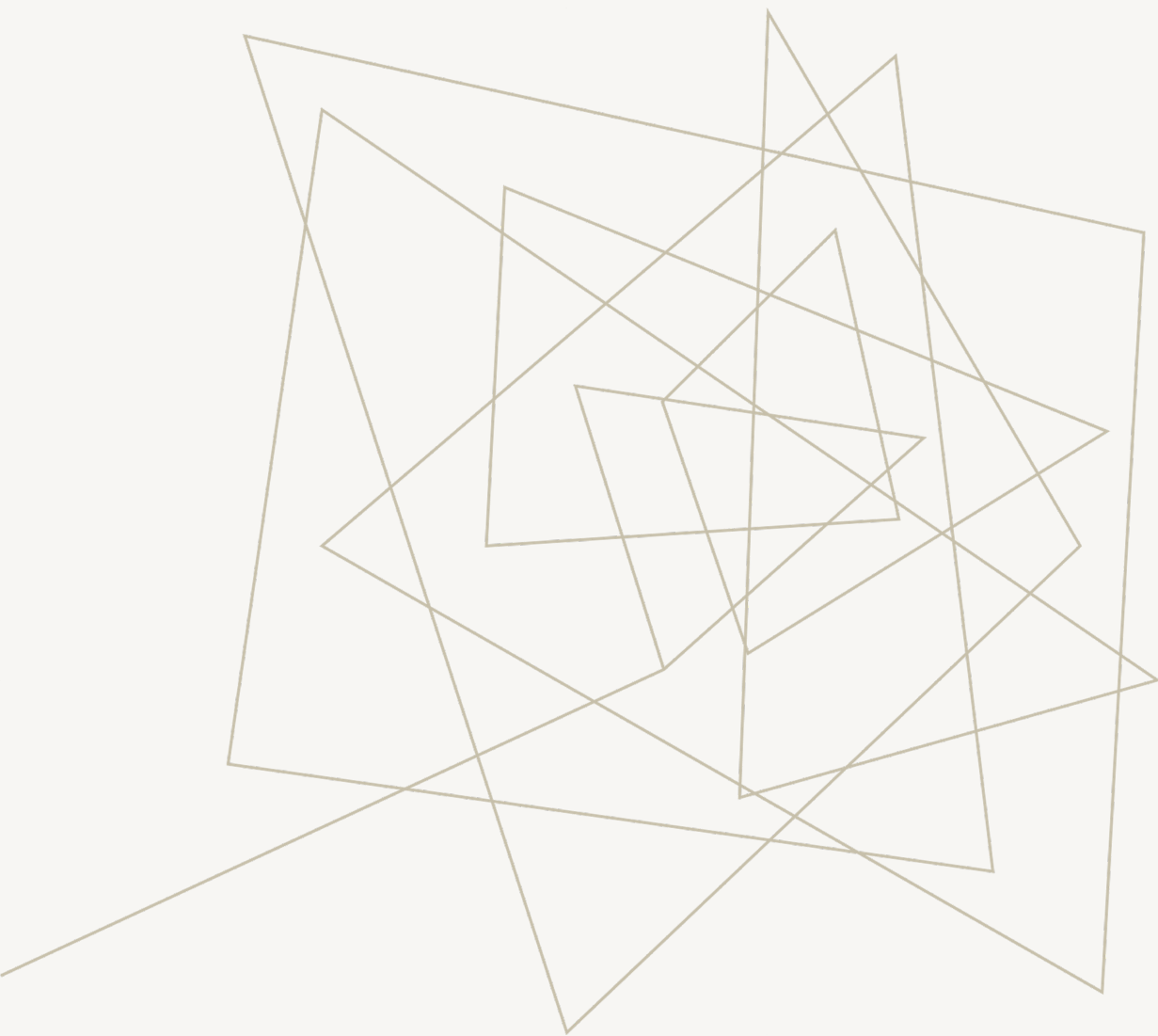
УСКОРЯЕМ БАГ-РЕПОРТИНГ С ПОМОЩЬЮ САМОПИСНОГО БРАУЗЕРНОГО РАСШИРЕНИЯ

Анастасия Нестерова

ОБО МНЕ

- 10+ лет в IT
- 9+ лет в тестировании ПО
- Внедрила автоматизацию с нуля
- Отвечаю за весь процесс тестирования: от планирования до завершения
- Занимаюсь ручным и автоматическим тестированием
- Веду канал в телеграмм **QA or not QA?**





СИТУАЦИИ

СИТУАЦИИ

Описываем шаги для тикета



СИТУАЦИИ

Описываем шаги для тикета

Проверяем списки опций, ссылок, ...

СИТУАЦИИ

Описываем шаги для тикета

Проверяем списки опций, ссылок, ...

Не открыты DevTools

СИТУАЦИИ

Описываем шаги для тикета

Проверяем списки опций, ссылок, ...

Не открыты DevTools

Время, когда возникла проблема, чтобы быстро
найти в логах

СИТУАЦИИ

Описываем шаги для тикета

Проверяем списки опций, ссылок, ...

Не открыты DevTools

Время, когда возникла проблема, чтобы быстро
найти в логах

Отчет о сессии исследовательского тестирования

**И ТАК
СОЙДЕТ**



А ПОТОМ...

Смотреть запись



Собеседование QA под нейросетью: когда ИИ говорит «Да»

ИИ уже активно меняет формат собеседований: от генерации заданий и оценки кандидатов до попыток заменить интервьюеров. Во время доклада покажем реальные примеры ИИ-агентов, расскажем, как сами кандидаты используют ИИ (не всегда честно), и дадим практические советы, как использовать технологии в свою пользу и не попасться на уловки.



Михаил Новотарский
Сбер



Евгений Гребенщиков
Сбер

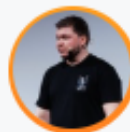


Смотреть запись



Браузерные инструменты: пишем свое расширение

Иногда для тестирования не хватает одной кнопки в интерфейсе... и тогда мы можем добавить ее сами. Расскажу, как создать собственное браузерное расширение под задачи QA. Обсудим автозаполнение форм, отправку HTTP-запросов, работу с DOM, быстрые проверки и сохранение настроек.



Алексей Иванов
2ГИС




А ПОТОМ...

Vadim Nikitenko
Райффайзен Банк
Москва, Россия

Язык доклада: 

Генерация автотестов с помощью MCP
+ LLM

Секционный доклад (40 мин)

Материалы: 




Евгений Гребенщиков
СБЕР
Москва, Россия

Язык доклада: 

Генерация тест-кейсов на основе
требований с помощью AI: От идеи до
мультиагентной системы

Секционный доклад (40 мин)

Материалы: 





ИДЕЯ



ИДЕЯ

Браузер видит все мои действия

Нужны хранилища с действиями и ошибками

Тогда я смогу из этих хранилищ получить всю информацию, необходимую для описания проблемы

ИДЕЯ

Браузерное расширение **QA Trace**

- Отслеживает действия пользователя
- Мониторит ошибки (консоль, network, UI)
- Собирает промпт для LLM





**КАК РАБОТАЮТ
РАСШИРЕНИЯ?**

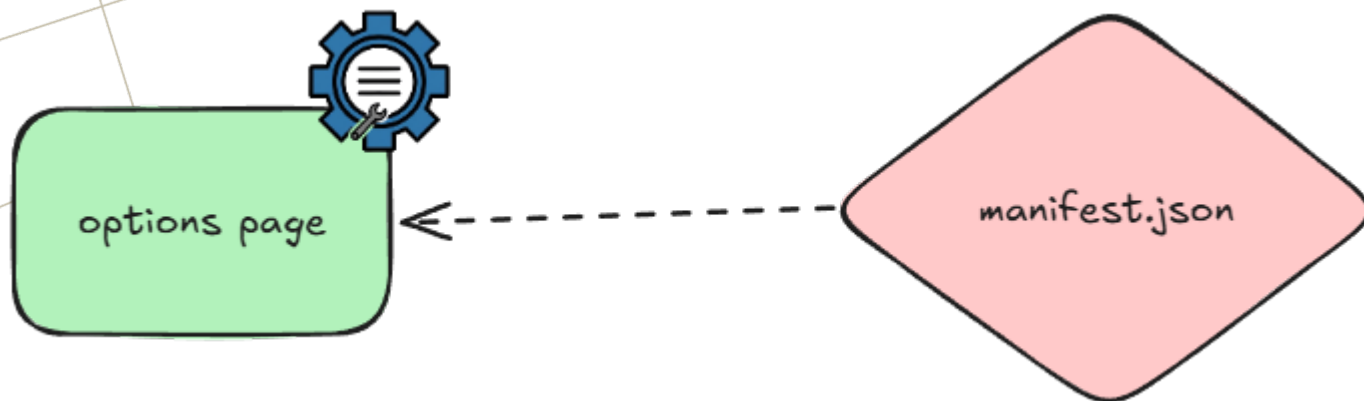
КАК РАБОТАЮТ РАСШИРЕНИЯ?



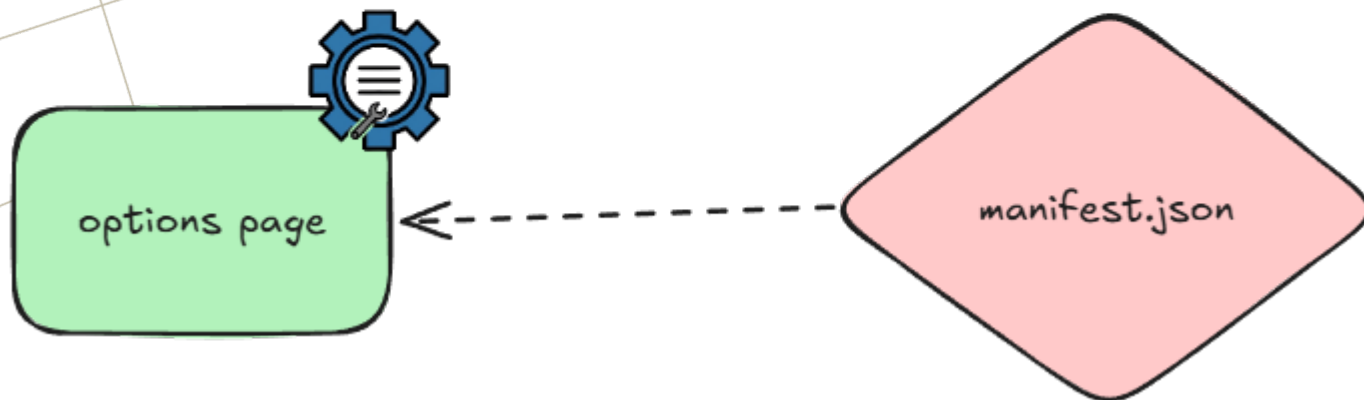
КАК РАБОТАЮТ РАСШИРЕНИЯ?



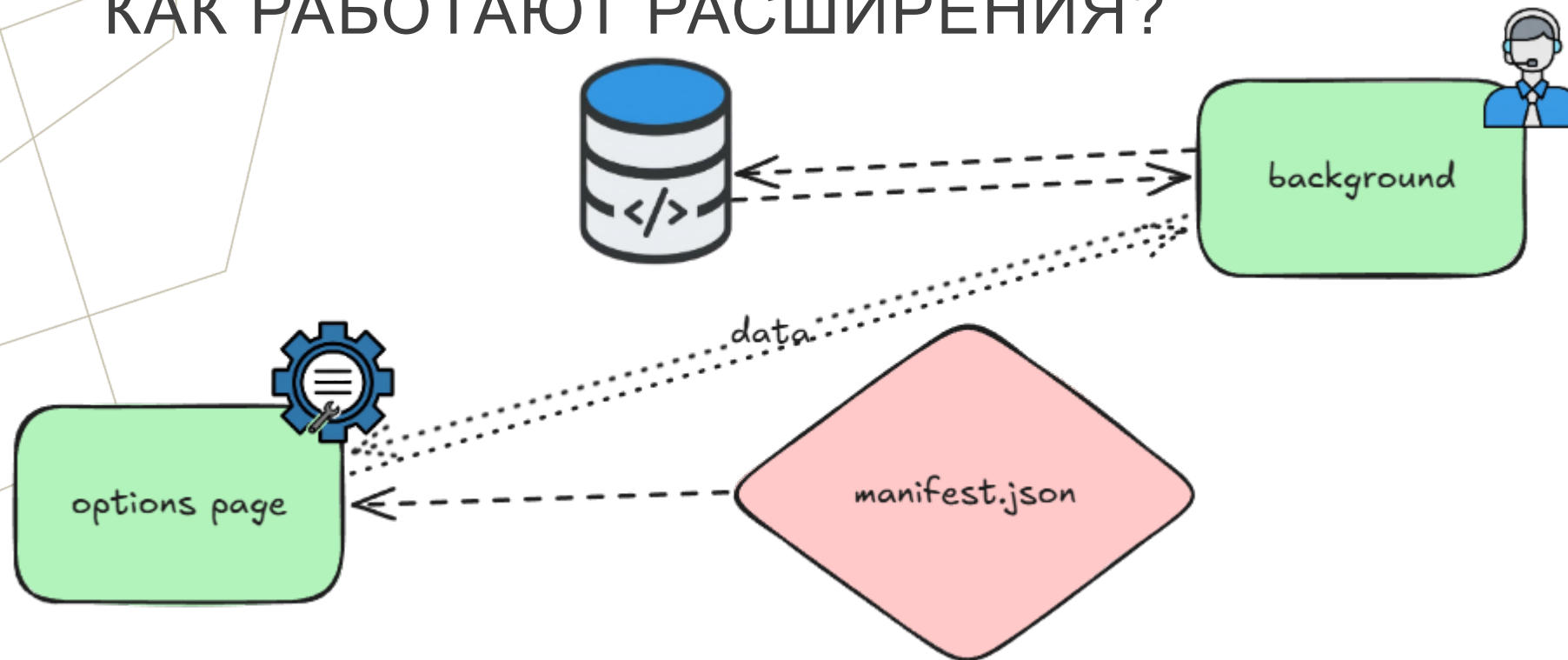
КАК РАБОТАЮТ РАСШИРЕНИЯ?



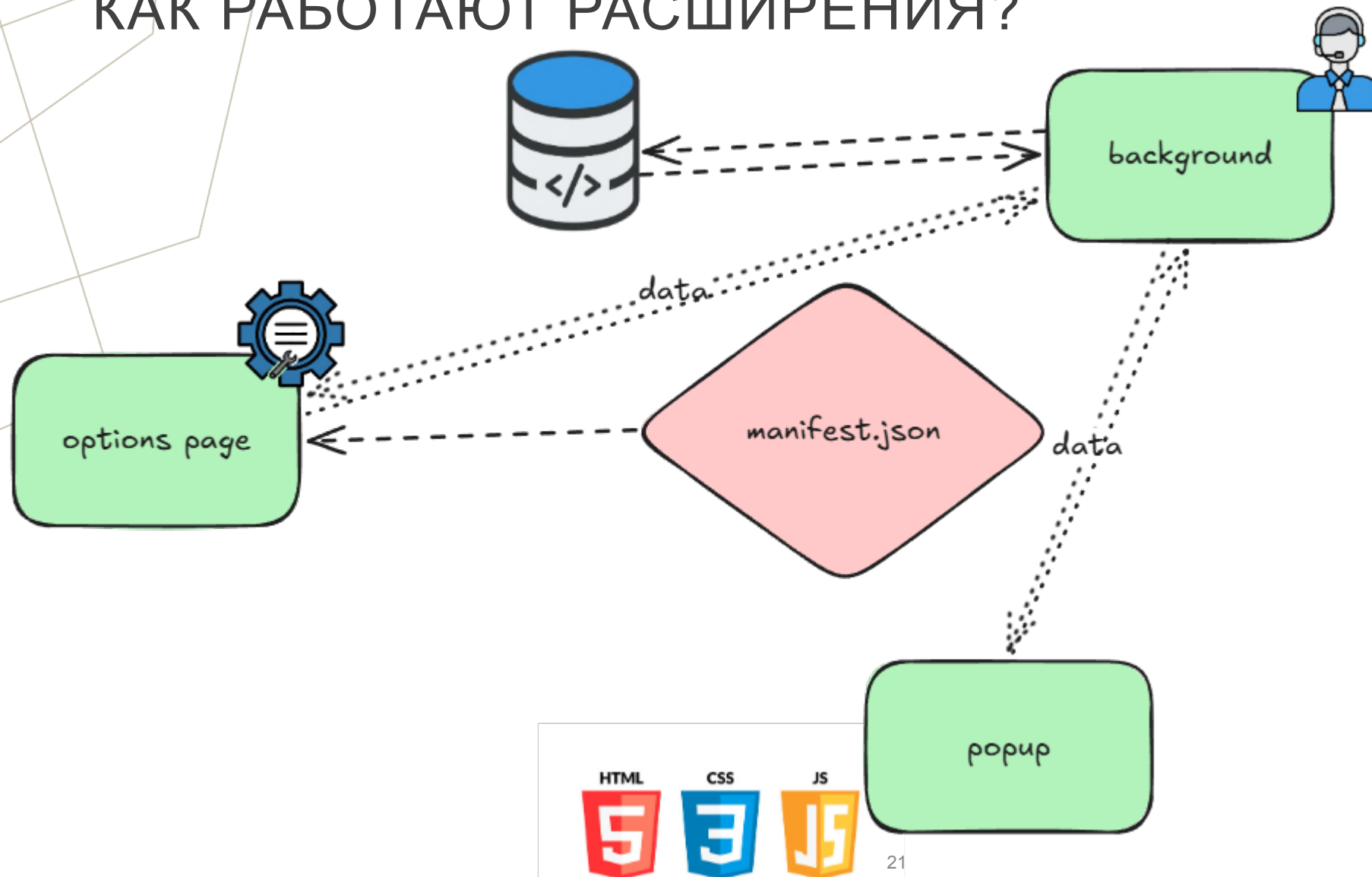
КАК РАБОТАЮТ РАСШИРЕНИЯ?



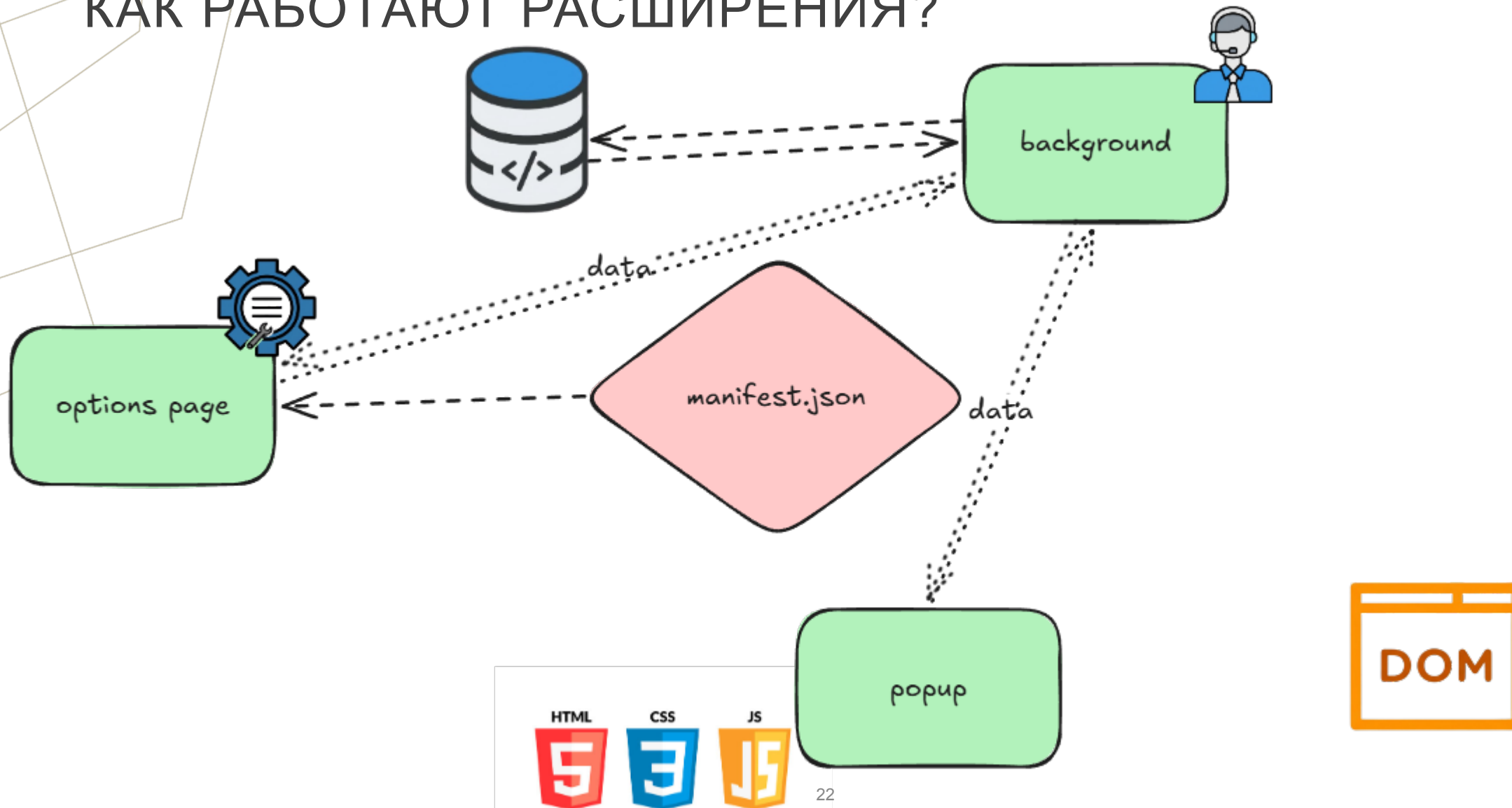
КАК РАБОТАЮТ РАСШИРЕНИЯ?



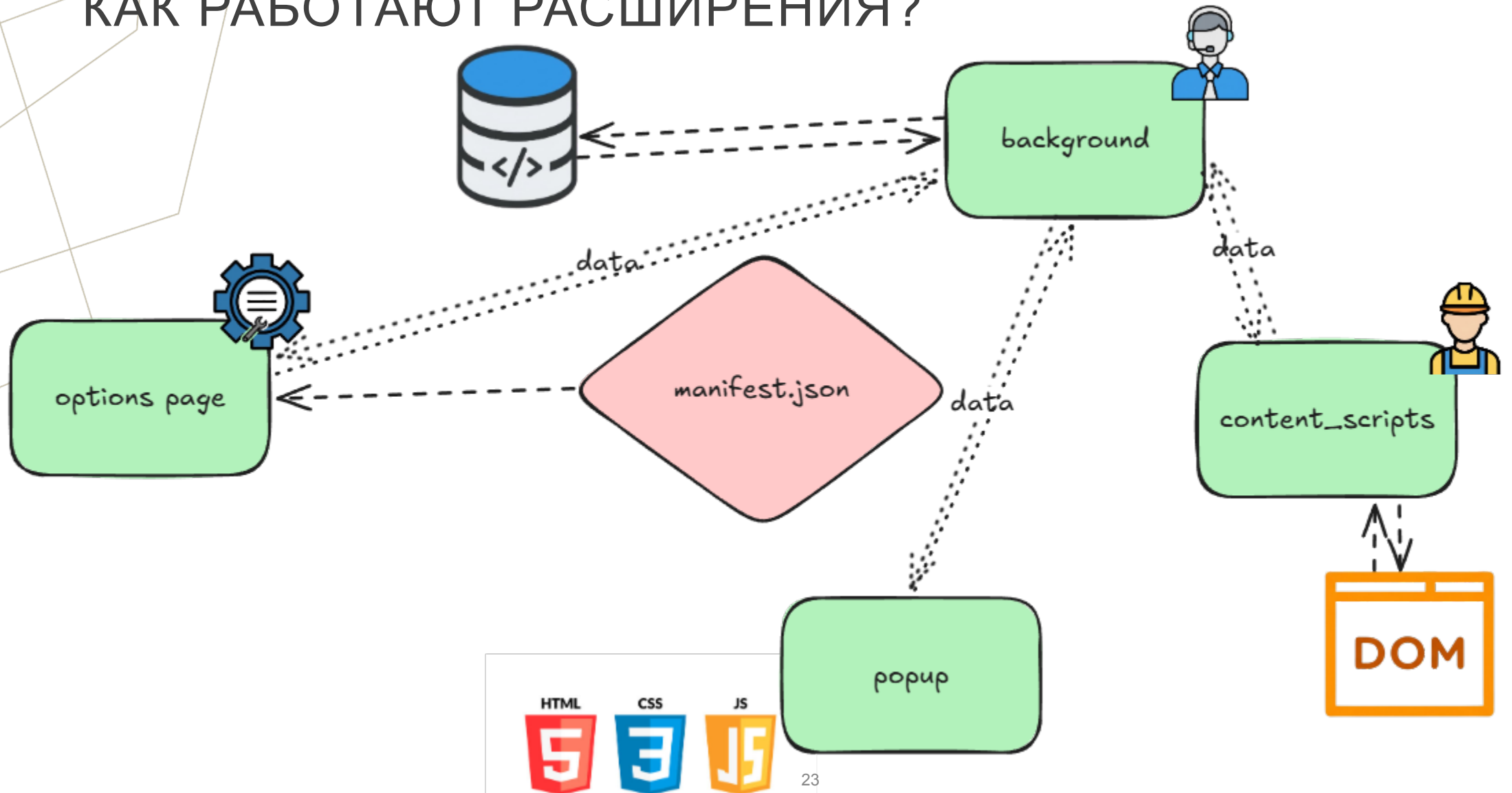
КАК РАБОТАЮТ РАСШИРЕНИЯ?



КАК РАБОТАЮТ РАСШИРЕНИЯ?



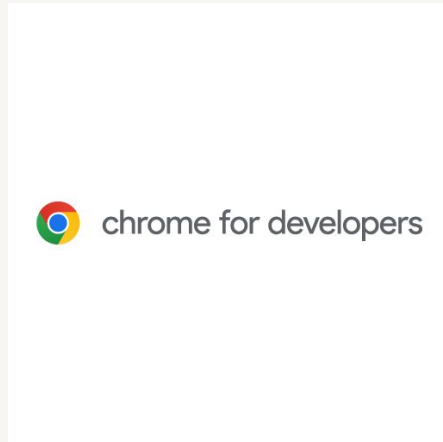
КАК РАБОТАЮТ РАСШИРЕНИЯ?





ПРОТОТИП

ПРОТОТИП



Документация для разработчиков



Vite



TypeScript



Cursor



ChatGPT

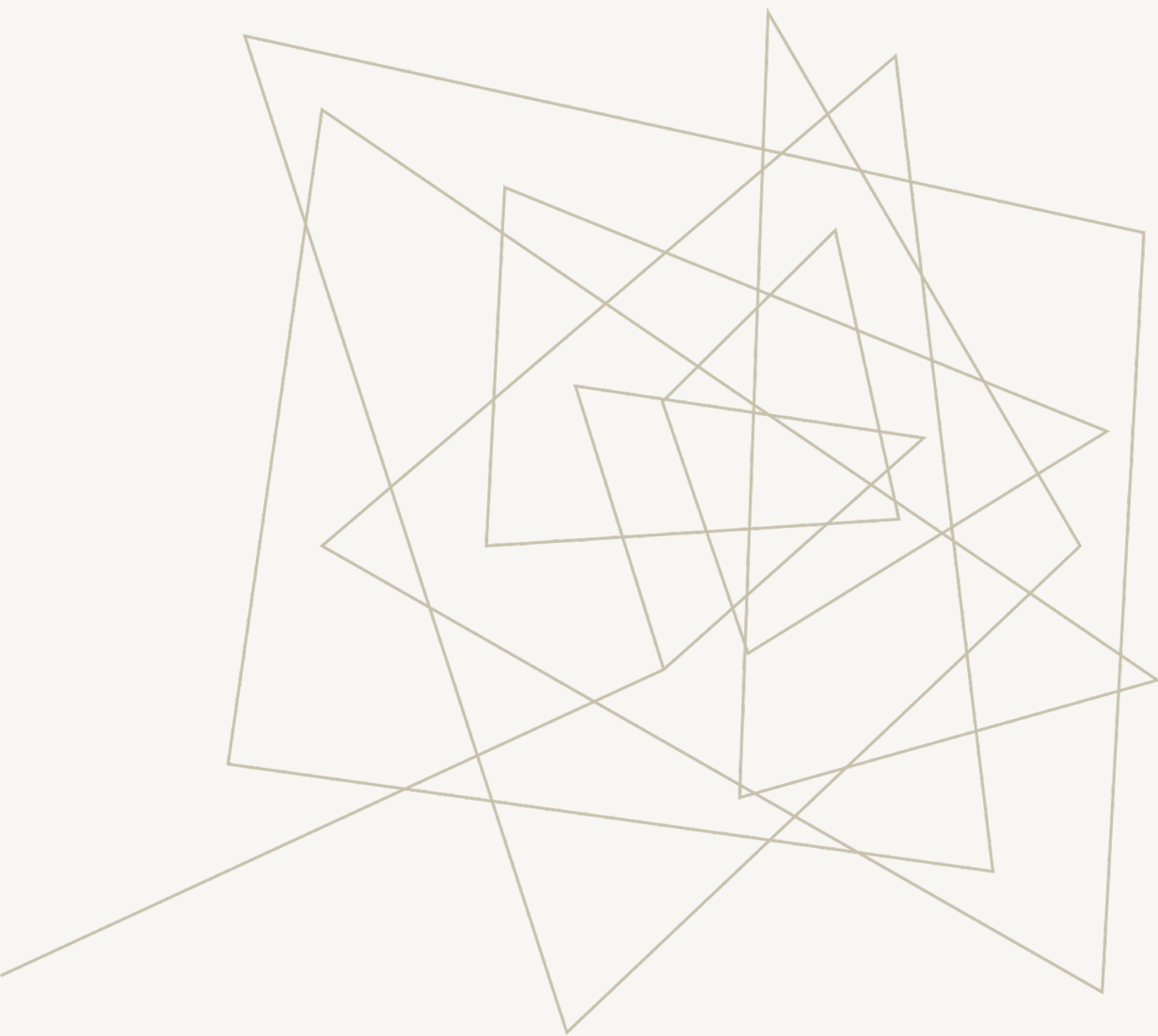


ПРОТОТИП

Браузерное расширение **QA Trace**

- Отслеживает действия пользователя
- Мониторит ошибки (консоль, network, UI)
- Собирает промпт для LLM

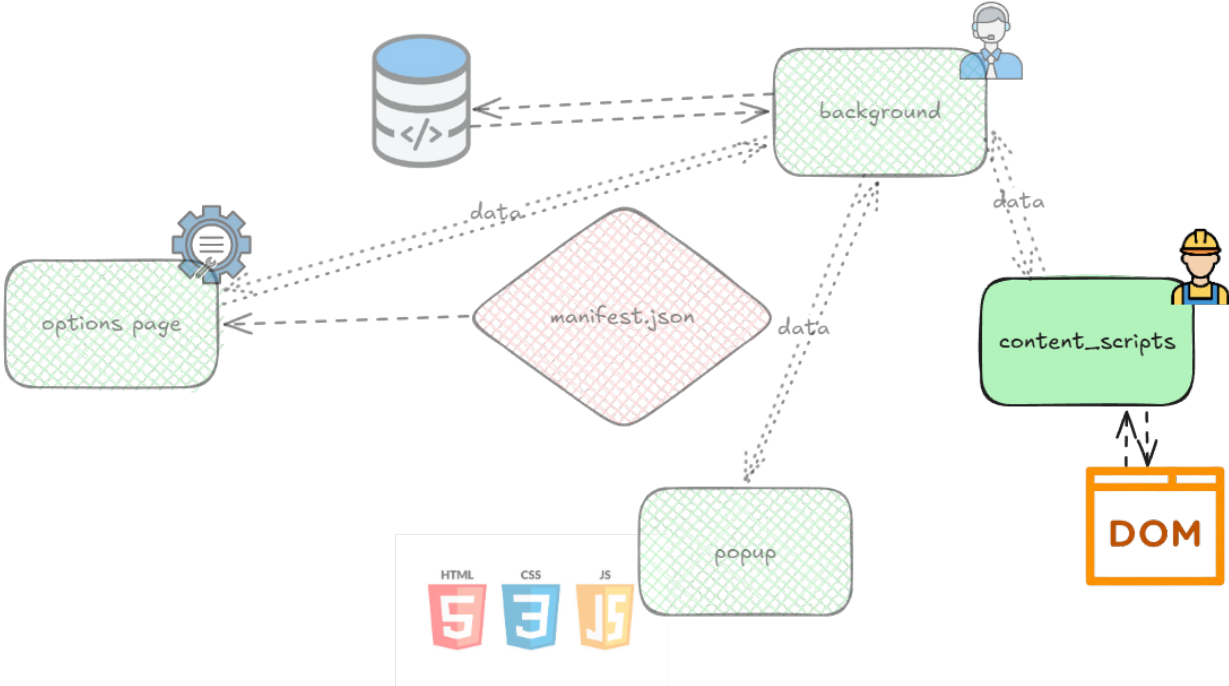




ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ

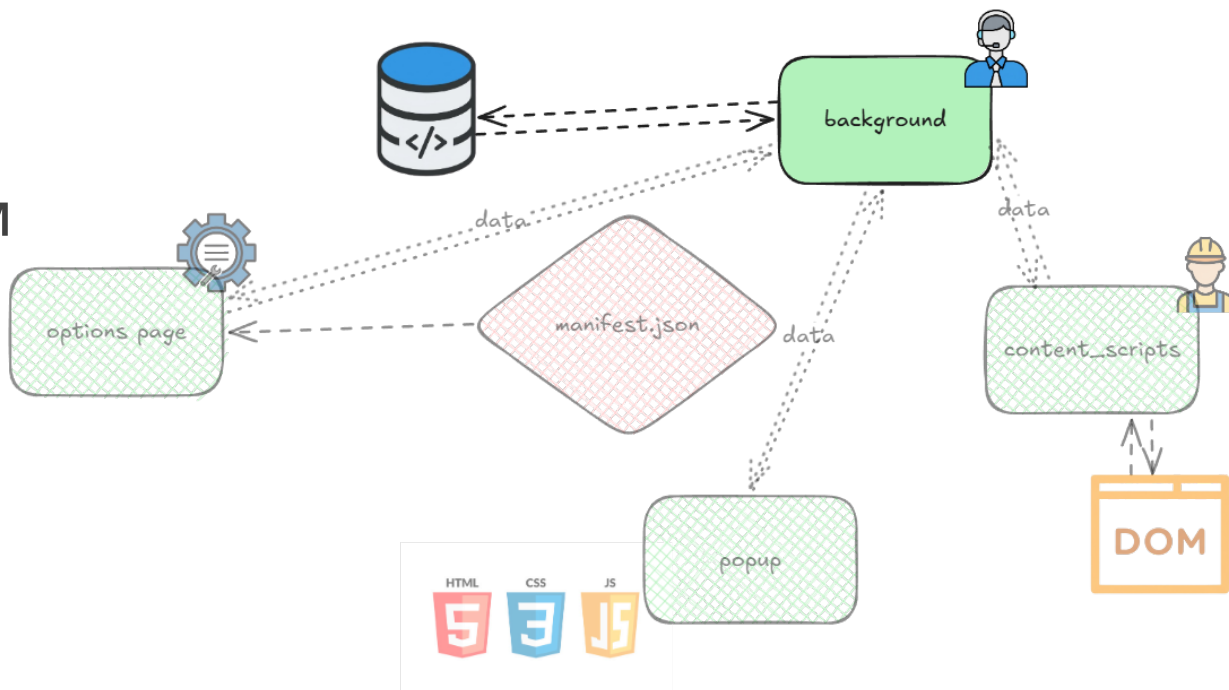
ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ

1. Event listeners (click, input, ...)



ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ

1. Event listeners (click, input, ...)
2. Каждое действие записываем в хранилище



USER-ACTION-TRACKER.TS

```
private setupEventListeners(): void {  
  document.addEventListener('click', this.handleClick.bind(this), options: true);  
  document.addEventListener('dblclick', this.handleDbClick.bind(this), options: true);  
  document.addEventListener('input', this.handleInput.bind(this), options: true);  
  document.addEventListener('change', this.handleChange.bind(this), options: true);  
}
```

USER-ACTION-TRACKER.TS

```
private async handleClick(event: Event) {  
  const target = event.target as HTMLElement  
  await this.recordAction({  
    type: 'click',  
    element: target.tagName,  
    value: target.textContent?.trim() || '',  
    selector: this.getElementSelector(target),  
  })  
}
```

USER-ACTION-TRACKER.TS

```
private async recordAction(action: Omit<UserAction, 'id' | 'timestamp' | "tabInfo">) {  
  const labelText : string = this.getClosestLabelText(action.selector)  
  const limit : number = await TextUtils.getConfiguredTextLimit()  
  const fullAction: Omit<UserAction, "tabInfo"> = {  
    timestamp: Date.now(),  
    ...action,  
    value: TextUtils.truncateText(action.value, limit) || undefined,  
    labelText: TextUtils.truncateText(labelText, limit) || undefined  
  }  
  
  await browser.runtime.sendMessage({  
    type: 'USER_ACTION',  
    data: fullAction,  
  })  
}
```

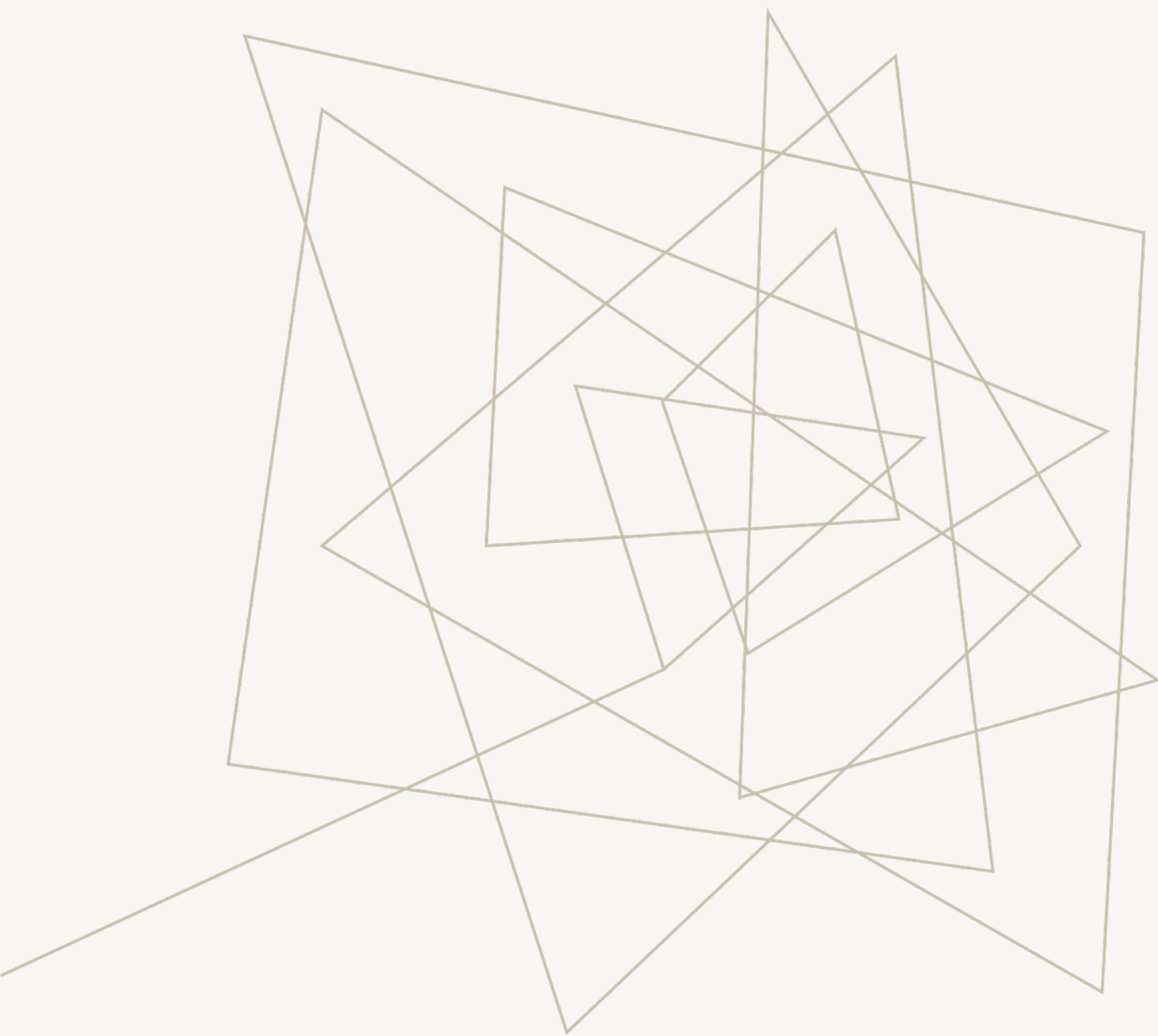
USER-ACTION-TRACKER.TS

```
private async recordAction(action: Omit<UserAction, 'id' | 'timestamp' | "tabInfo">) {  
  const labelText : string = this.getClosestLabelText(action.selector)  
  const limit : number = await TextUtils.getConfiguredTextLimit()  
  const fullAction: Omit<UserAction, "tabInfo"> = {  
    timestamp: Date.now(),  
    ...action,  
    value: TextUtils.truncateText(action.value, limit) || undefined,  
    labelText: TextUtils.truncateText(labelText, limit) || undefined  
  }  
}
```

```
await browser.runtime.sendMessage({  
  type: 'USER_ACTION',  
  data: fullAction,  
})  
}
```

SERVICE-WORKER.TS

```
browser.runtime.onMessage.addListener(async (message: any, sender: MessageSender) => {  
  const tabInfo : TabInfo = getTabInfoFromSender (sender);  
  
  switch (message.type) {  
    case 'USER_ACTION':  
      const userAction = message.data as Omit<UserAction, "tabInfo">;  
      await StorageManager.addUserAction(userAction, tabInfo);  
      break  
    }  
  });
```



ОШИБКИ (UI)

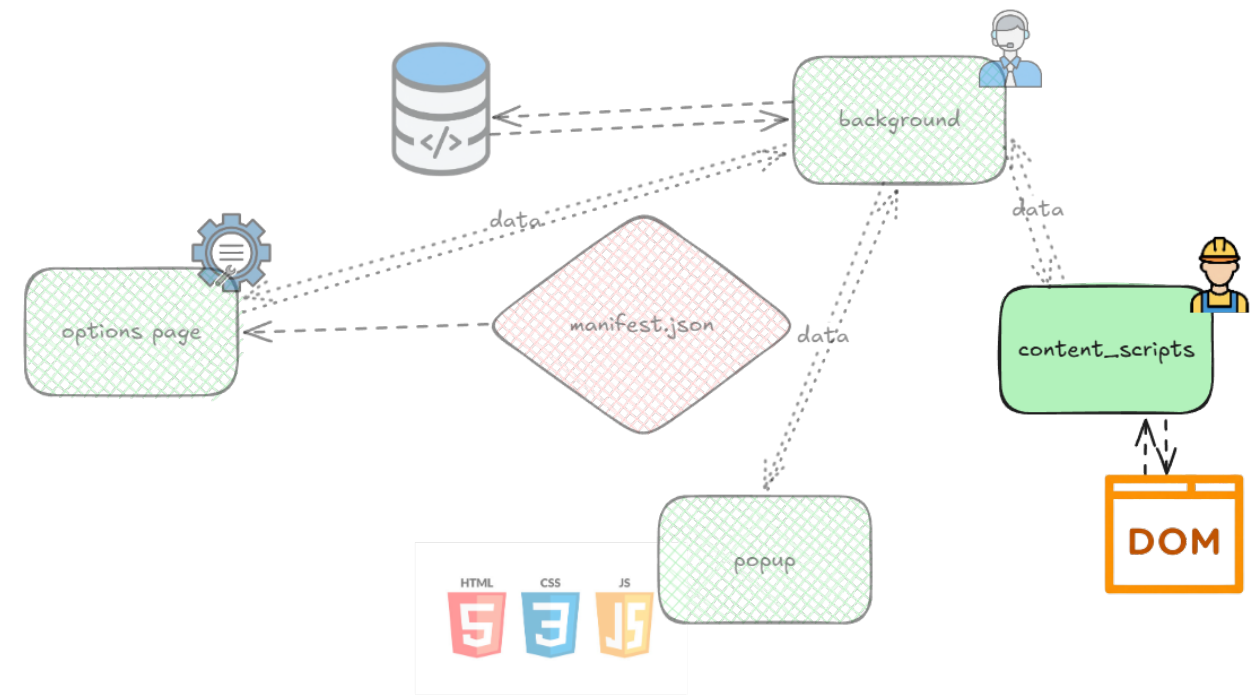
ОШИБКИ (UI)

1. Селекторы ошибок



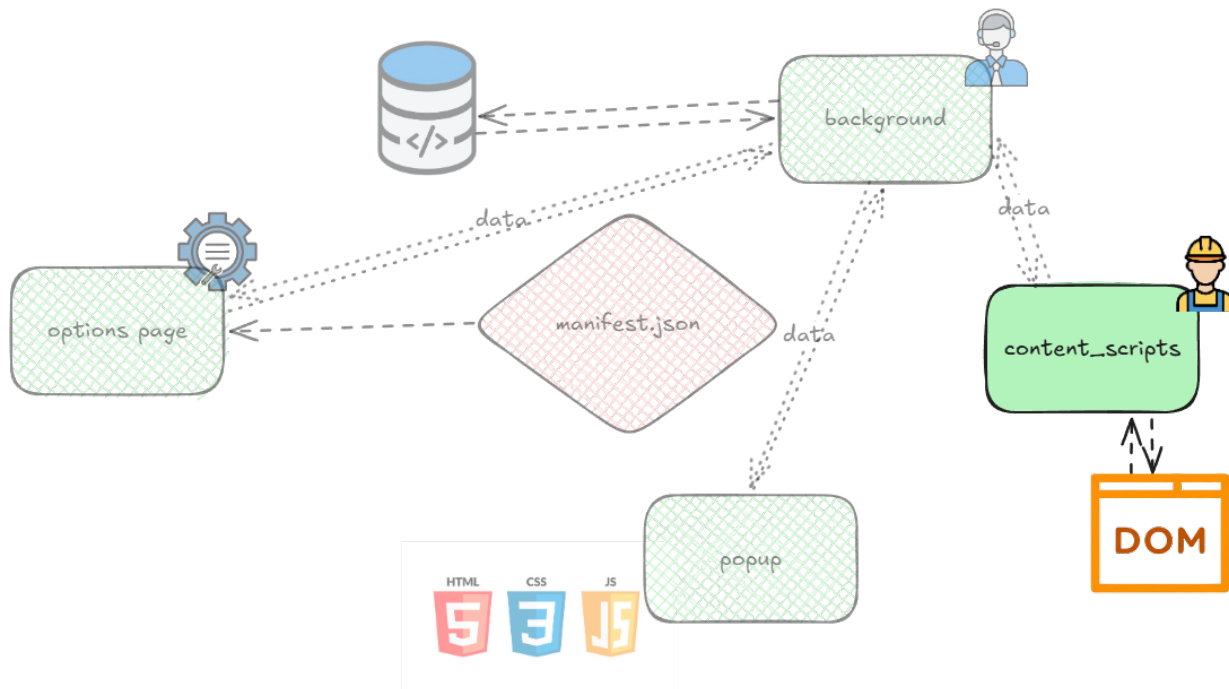
ОШИБКИ (UI)

1. Селекторы ошибок
2. Отслеживаем изменения в DOM



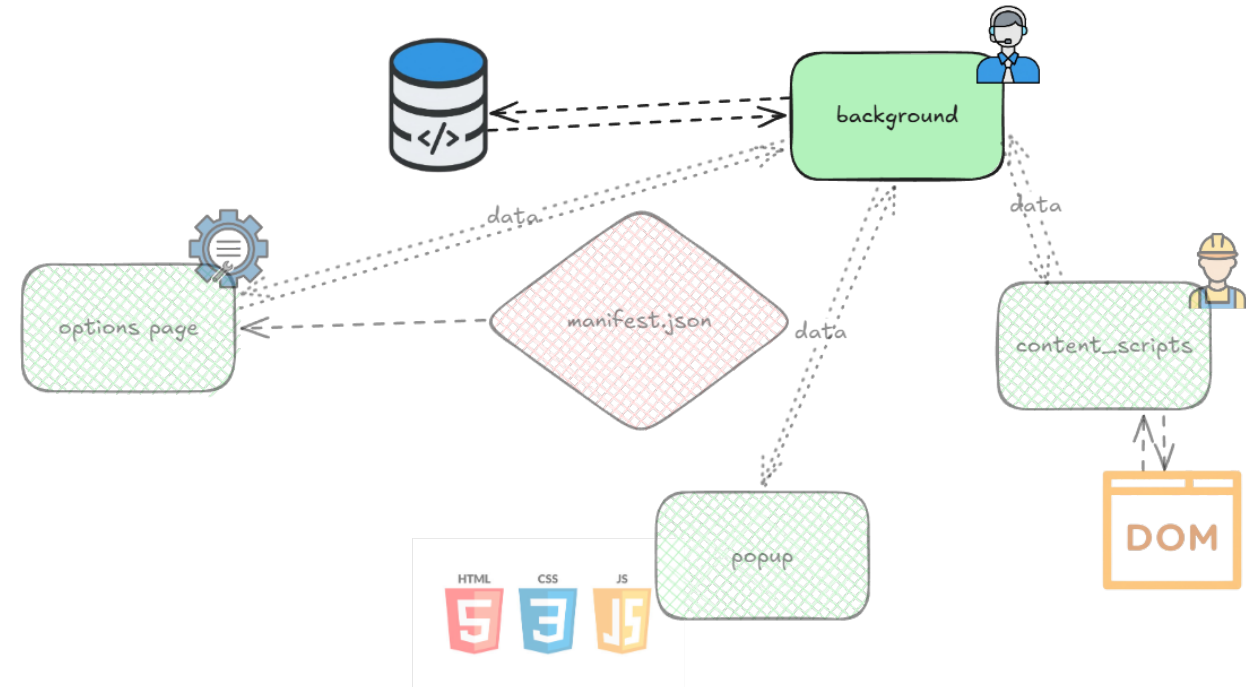
ОШИБКИ (UI)

1. Селекторы ошибок
2. Отслеживаем изменения в DOM
3. Проверяем каждый элемент (ошибка или нет)



ОШИБКИ (UI)

1. Селекторы ошибок
2. Отслеживаем изменения в DOM
3. Проверяем каждый элемент (ошибка или нет)
4. Добавляем ошибку в хранилище



ERROR-DETECTOR.TS

```
async setupUIErrorTracking(selectors: string[] = ['div[id^="__error"]']) {
  const observer = new MutationObserver(() => {
    const errors : NodeListOf<Element> = document.querySelectorAll(selectors.join( separator: ', '));

    (async () => {
      const recordPromises : Promise<void>[] = Array.from(errors).map(
        (error : Element ) => {
          return this.recordError({
            type: 'ui',
            message: error.textContent?.trim() || 'UI Error'
          });
        }
      );

      await Promise.all(recordPromises);
    })();
  });

  observer.observe(document.body, { childList: true, subtree: true });
  return observer;
}
```

ERROR-DETECTOR.TS

```
async setupUIErrorTracking(selectors: string[] = ['div[id^="__error"]']) {
  const observer = new MutationObserver(() => {
    const errors : NodeListOf<Element> = document.querySelectorAll(selectors.join( separator: ','));

    (async () => {
      const recordPromises : Promise<void>[] = Array.from(errors).map(
        (error : Element ) => {
          return this.recordError({
            type: 'ui',
            message: error.textContent?.trim() || 'UI Error'
          });
        }
      );

      await Promise.all(recordPromises);
    })();
  });

  observer.observe(document.body, { childList: true, subtree: true });
  return observer;
}
```

ERROR-DETECTOR.TS

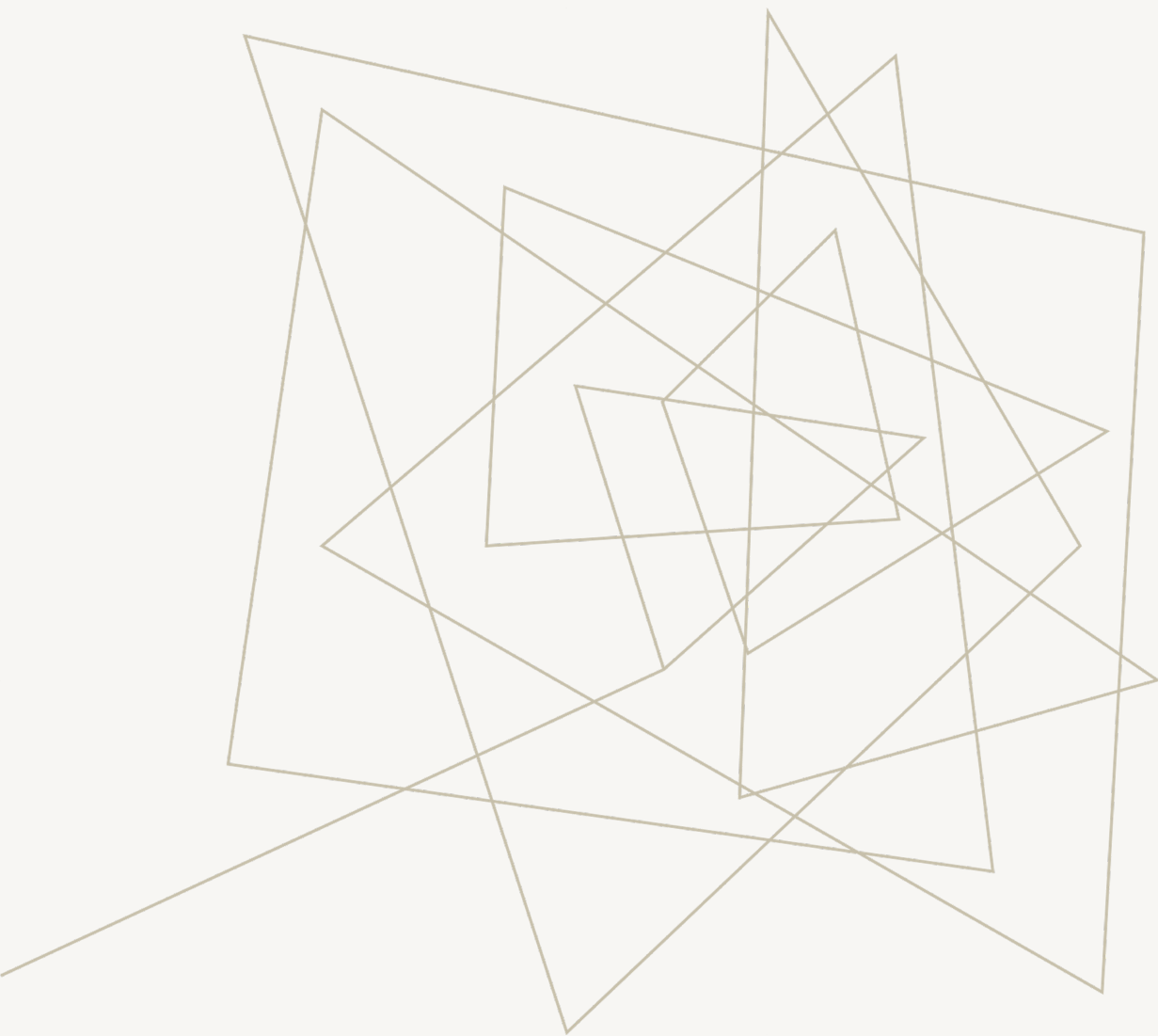
```
private async recordError(error: Omit<ErrorLog, 'id' | 'timestamp' | "tabInfo">) {  
  const limit: number = await this.getTextLimit();  
  const fullError: Omit<ErrorLog, "tabInfo"> = {  
    timestamp: Date.now(),  
    ...error,  
    message: truncateText(error.message, limit),  
    stack: error.stack  
  };  
  
  await browser.runtime.sendMessage({  
    type: 'ERROR_DETECTED',  
    data: fullError,  
  });  
}
```

ERROR-DETECTOR.TS

```
private async recordError(error: Omit<ErrorLog, 'id' | 'timestamp' | "tabInfo">) {  
  const limit : number = await this.getTextLimit();  
  const fullError: Omit<ErrorLog, "tabInfo"> = {  
    timestamp: Date.now(),  
    ...error,  
    message: truncateText(error.message, limit),  
    stack: error.stack  
  };  
  
  await browser.runtime.sendMessage({  
    type: 'ERROR_DETECTED',  
    data: fullError,  
  });  
}
```

SERVICE-WORKER.TS

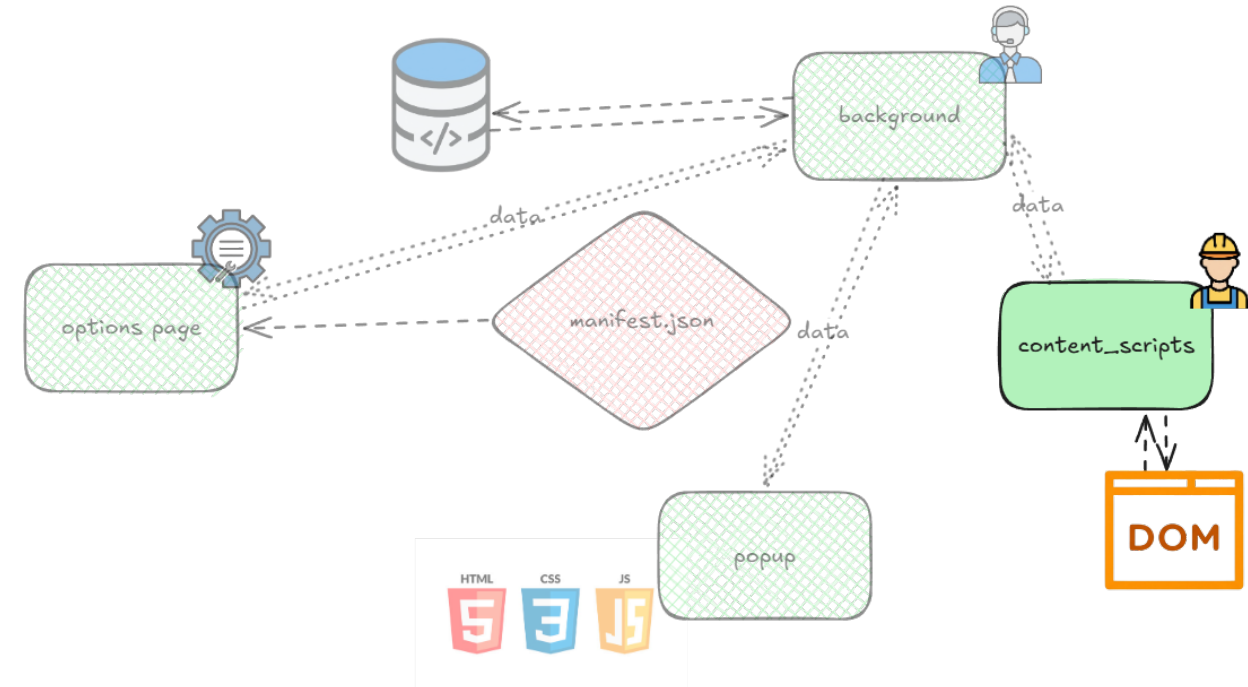
```
browser.runtime.onMessage.addListener(async (message: any, sender: MessageSender) => {  
  const tabInfo : TabInfo = getTabInfoFromSender (sender);  
  
  switch (message.type) {  
    case 'USER_ACTION':  
      const userAction = message.data as Omit<UserAction, "tabInfo">;  
      await StorageManager.addUserAction(userAction, tabInfo);  
      break  
  
    case 'ERROR_DETECTED':  
      const error = message.data as Omit<ErrorLog, "tabInfo">;  
      await StorageManager.addError(error, tabInfo);  
      await showErrorNotification(error);  
      break  
  
  }  
});
```



ОШИБКИ
(КОНСОЛЬ,
NETWORK)

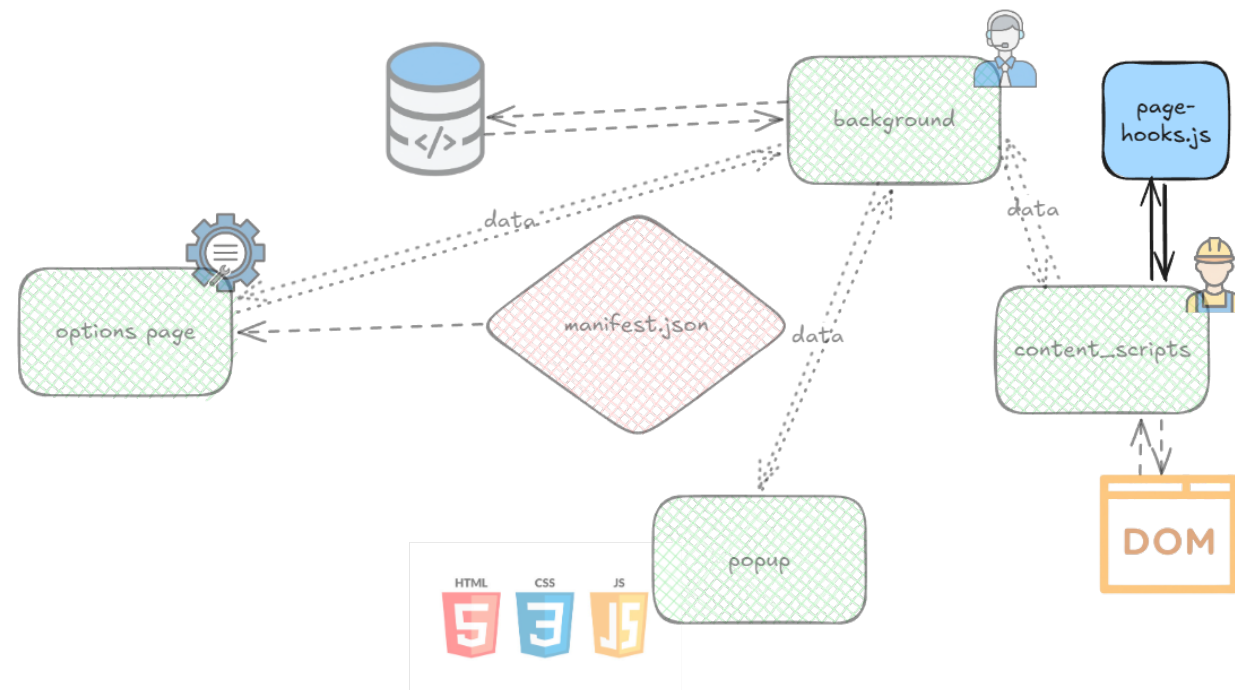
ОШИБКИ (КОНСОЛЬ, NETWORK)

1. Внедряемся в контекст страницы



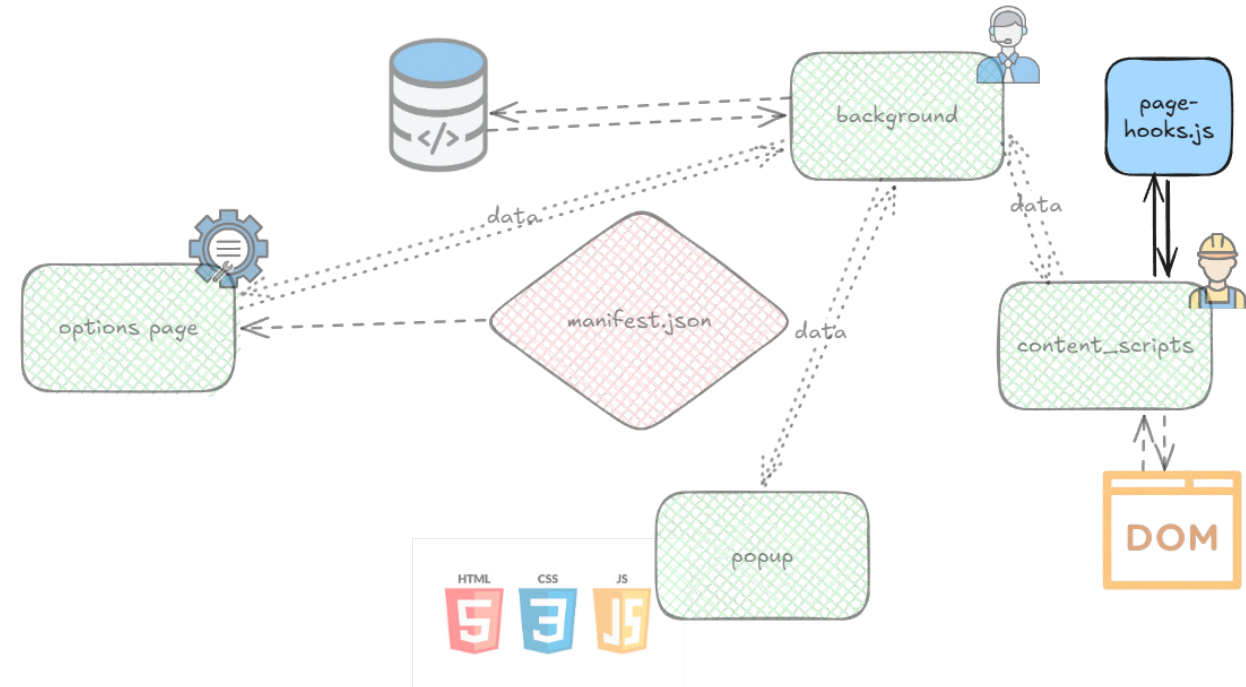
ОШИБКИ (КОНСОЛЬ, NETWORK)

1. Внедряемся в контекст страницы
2. Перехватываем ошибки (консоль, network)



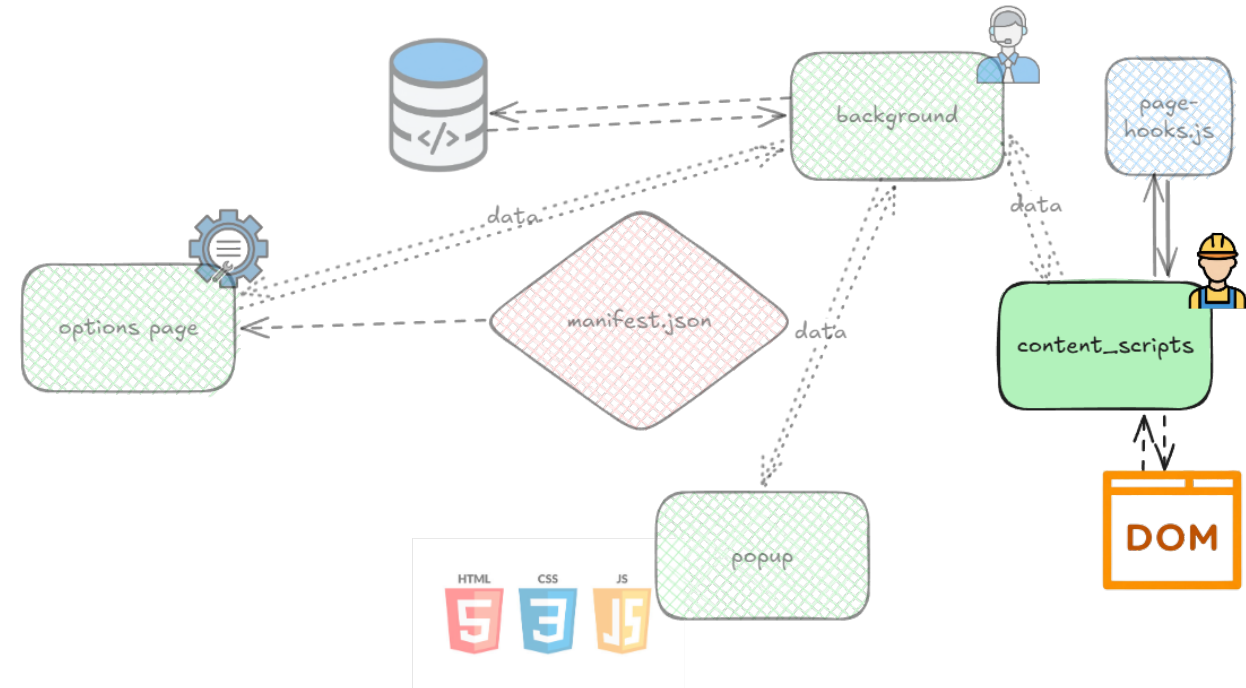
ОШИБКИ (КОНСОЛЬ, NETWORK)

1. Внедряемся в контекст страницы
2. Перехватываем ошибки (консоль, network)
3. Передаем ошибку расширению



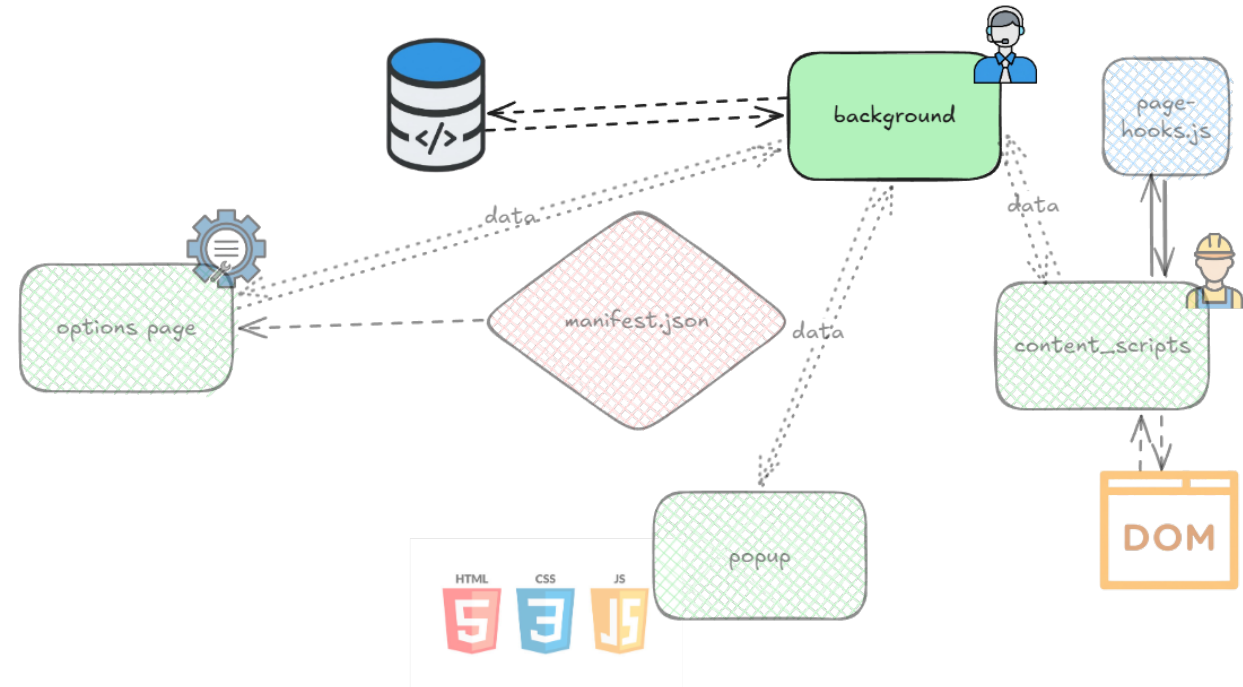
ОШИБКИ (КОНСОЛЬ, NETWORK)

1. Внедряемся в контекст страницы
2. Перехватываем ошибки (консоль, network)
3. Передаем ошибку расширению
4. Обрабатываем ошибку



ОШИБКИ (КОНСОЛЬ, NETWORK)

1. Внедряемся в контекст страницы
2. Перехватываем ошибки (консоль, network)
3. Передаем ошибку расширению
4. Обрабатываем ошибку
5. Добавляем ошибку в хранилище



PAGE-HOOKS.TS

```
const originalFetch : ((input: URL | RequestInfo, init?: Reques... = window.fetch.bind(window);
window.fetch = async (...args: Parameters<typeof fetch>): Promise<Response> => {
  const input : URL | RequestInfo = args[0];
  const init : RequestInit = args[1] || {};
  const requestUrlRaw : string = typeof input === 'string' ? input : ((input as Request)?.url || '');
  const requestUrl : string = stripRequestUrlForTelemetry(requestUrlRaw);
  const requestMethod : string = (init as RequestInit)?.method || (input as Request)?.method || 'GET';
  const requestHeaders : Record<string, string> = headersToObject((init as RequestInit)?.headers || (input as Request)?.headers);
  const requestBody : string = parseRequestBody((init as RequestInit)?.body);
  try {
    const response : Response = await originalFetch(...args);
    if (!response.ok) {
      const responseHeaders : Record<string, string> = headersToObject(response.headers);
      const responseBody : string = await readResponseBodySafe(response);
      post( kind: 'network', {status: response.status...});
    }
    return response;
  } catch (error) {...}
};
```

PAGE-HOOKS.TS

```
const originalFetch : ((input: URL | RequestInfo, init?: Reques... = window.fetch.bind(window);
window.fetch = async (...args: Parameters<typeof fetch>): Promise<Response> => {
  const input : URL | RequestInfo = args[0];
  const init : RequestInit = args[1] || {};
  const requestUrlRaw : string = typeof input === 'string' ? input : ((input as Request)?.url || '');
  const requestUrl : string = stripRequestUrlForTelemetry(requestUrlRaw);
  const requestMethod : string = (init as RequestInit)?.method || (input as Request)?.method || 'GET';
  const requestHeaders : Record<string, string> = headersToObject((init as RequestInit)?.headers || (input as Request)?.headers);
  const requestBody : string = parseRequestBody((init as RequestInit)?.body);
  try {
    const response : Response = await originalFetch(...args);
    if (!response.ok) {
      const responseHeaders : Record<string, string> = headersToObject(response.headers);
      const responseBody : string = await readResponseBodySafe(response);
      post( kind: 'network', {status: response.status...});
    }
    return response;
  } catch (error) {...}
};
```

PAGE-HOOKS.TS

```
const originalFetch : ((input: URL | RequestInfo, init?: Reques... = window.fetch.bind(window);
window.fetch = async (...args: Parameters<typeof fetch>): Promise<Response> => {
  const input : URL | RequestInfo = args[0];
  const init : RequestInit = args[1] || {};
  const requestUrlRaw : string = typeof input === 'string' ? input : ((input as Request)?.url || '');
  const requestUrl : string = stripRequestUrlForTelemetry(requestUrlRaw);
  const requestMethod : string = (init as RequestInit)?.method || (input as Request)?.method || 'GET';
  const requestHeaders : Record<string, string> = headersToObject((init as RequestInit)?.headers || (input as Request)?.headers);
  const requestBody : string = parseRequestBody((init as RequestInit)?.body);
  try {
    const response : Response = await originalFetch(...args);
    if (!response.ok) {
      const responseHeaders : Record<string, string> = headersToObject(response.headers);
      const responseBody : string = await readResponseBodySafe(response);
      post( kind: 'network', {status: response.status...});
    }
    return response;
  } catch (error) {...}
};
```

PAGE-HOOKS.TS

```
const originalFetch : ((input: URL | RequestInfo, init?: Reques... = window.fetch.bind(window);
window.fetch = async (...args: Parameters<typeof fetch>): Promise<Response> => {
  const input : URL | RequestInfo = args[0];
  const init : RequestInit = args[1] || {};
  const requestUrlRaw : string = typeof input === 'string' ? input : ((input as Request)?.url || '');
  const requestUrl : string = stripRequestUrlForTelemetry(requestUrlRaw);
  const requestMethod : string = (init as RequestInit)?.method || (input as Request)?.method || 'GET';
  const requestHeaders : Record<string, string> = headersToObject((init as RequestInit)?.headers || (input as Request)?.headers);
  const requestBody : string = parseRequestBody((init as RequestInit)?.body);
  try {
    const response : Response = await originalFetch(...args);
    if (!response.ok) {
      const responseHeaders : Record<string, string> = headersToObject(response.headers);
      const responseBody : string = await readResponseBodySafe(response);
      post( kind: 'network', {status: response.status...});
    }
    return response;
  } catch (error) {...}
};
```

PAGE-HOOKS.TS

```
const post : (kind: PostKind, payload: ConsolePayload ... = (kind: PostKind, payload: ConsolePayload | NetworkPayload) => {  
  if (!qaTraceToken) return;  
  const targetOrigin : string = typeof window !== 'undefined' && window.location ? window.location.origin : '*';  
  window.postMessage({  
    source: 'qa-trace',  
    token: qaTraceToken,  
    kind,  
    payload  
  }, targetOrigin);  
};
```

ERROR-DETECTOR.TS

```
private async ensurePageMessageListener() {
  if (this.pageMessageListenerAdded) return;
  window.addEventListener( type: 'message', (event: MessageEvent) => {
    if (event.source !== window || !event.data || event.data.source !== 'qa-trace')
      return;
    if (event.data.token !== this.pageMessageToken)
      return;

    const { kind, payload } = event.data;
    if (!kind || !payload)
      return;

    if (kind === 'console') {
      void this.recordError({type: 'console'...});
    } else if (kind === 'network') {
      void this.recordError({type: 'network'...});
    }
  });
  this.pageMessageListenerAdded = true;
}
```

ERROR-DETECTOR.TS

```
private async ensurePageMessageListener() {
  if (this.pageMessageListenerAdded) return;
  window.addEventListener( type: 'message', (event: MessageEvent) => {
    if (event.source !== window || !event.data || event.data.source !== 'qa-trace')
      return;
    if (event.data.token !== this.pageMessageToken)
      return;

    const { kind, payload } = event.data;
    if (!kind || !payload)
      return;


    if (kind === 'console') {
      void this.recordError({type: 'console'...});
    } else if (kind === 'network') {
      void this.recordError({type: 'network'...});
    }
  });
  this.pageMessageListenerAdded = true;
}
```

- **Summary:** The user created and configured [redacted] by setting various parameters including title [redacted] [redacted] then saved the configuration.

- **Description:**

1. Click the "Create [redacted] button
2. Enter "123" in the Title field
3. Click the [redacted]
4. Select [redacted]
5. Click "[redacted]"
6. In the "[redacted]"
7. In the "[redacted]"
8. In the "[redacted]"
9. Enable [redacted]
10. Enable [redacted]
11. In the "[redacted]"
12. Click "Save" [redacted]

Сообщение для DeepSeek

 Глубокое мышление

 Поиск







ДЕМО

Настройка

Адреса для мониторинга

https://example.com

Удалить

Добавить адрес

Отслеживаемые ошибки

- Сетевые ошибки
- Ошибки консоли
- UI-ошибки

Селекторы отслеживаемых UI-ошибок

div[id^="__error"]

Удалить

Добавить селектор

Язык ответов LLM

Язык ответа

Использовать язык браузера





Heisenbug

2026 Spring

Конференция по тестированию не только для тестировщиков

27–28 апреля

МОСКВА + ONLINE ⓘ

Гостиница «Рэдиссон Славянская»: пл. Евразии, 2

Купить билет

Расписание

45

спикеров

40

докладов

Heisenbug – крупнейшая в России конференция по тестированию программного обеспечения. В программе – доклады и дискуссии для специалистов

ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ

11

ОШИБКИ

1

Недавние действия

**Click In DIV With Label: К Основному
КонтентуРасписаниеСпикерыПартнерыАтмосфераЕщеМеди
НасБудущие КонференцииАрхивС...**

К основному
контентуРасписаниеСпикерыПартнерыАтмосфераЕщеМедиаЭксп
ертыО насБудущие конференцииАрхивС...
20:51:27

Click In A With Label: Закрыть

Краткое описание: При переходе из страницы спикера к архивному докладу и работе с материалами фиксируется network error: HEAD-запрос к next-data страницы спикера возвращает HTTP 307

Описание:

Предусловия (опционально): Ошибка зафиксирована в сетевых логах браузера.

Шаги для воспроизведения:

1. На вкладке «Heisenbug 2026 Spring» перейти в раздел «Расписание».
2. На странице расписания переключиться на дату «28 апреля» и открыть доклад «Инструменты тестировщика 2026».
3. На странице доклада перейти в профиль спикера «Юлия Атлыгина».
4. На странице спикера в блоке активностей применить фильтр по сезону «2021».
5. Открыть доклад «Исследовать нельзя тестировать».
6. На странице архивного доклада последовательно нажать «Презентация», затем «Скрыть чат», затем «Закрыть».
7. После этого вернуться к странице спикера (в логах та же вкладка фиксирует страницу «Юлия Атлыгина на Heisenbug 2026 Spring»).

Фактический результат:

В сетевых логах фиксируется ошибка:

HTTP 307 для HEAD-запроса к служебному next-data ресурсу страницы спикера «Юлия Атлыгина на Heisenbug 2026 Spring» (тип ошибки: network).


Ожидаемый результат:

При выполнении указанных действий не должна фиксироваться network error для запроса данных страницы спикера. Загрузка/переход между страницами спикера и архивного доклада должна обрабатываться без HTTP 307 в логе ошибки.



ОПИСЫВАЕМ ШАГИ ДЛЯ ТИКЕТА

ErrorMonitor

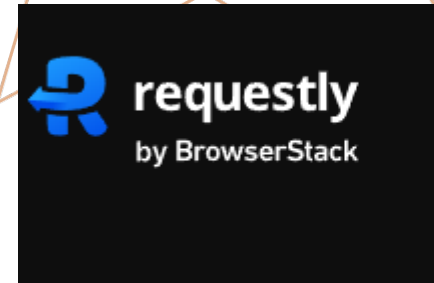
Анастасия Бортникова
Domeo
Санкт-Петербург, Россия
Язык доклада: 



ErrorMonitor - расширение
созданное тестировщиком для
тестировщиков

Блиц доклад (20 мин)

Requestly



OpenReplay



Open Replay



Loom



Brie



СИТУАЦИИ

+ Описываем шаги для тикета

Проверяем списки опций, ссылок, ...

Не открыты DevTools

Время, когда возникла проблема, чтобы быстро
найти в логах

Отчет о сессии исследовательского тестирования



Юлия Атлыгина

Фанат инструментов и чек-листов, ментор и преподаватель, консультант по тестированию и SAFe-методологиям, Jira-гуру.



Спикер

Keynote 28.04 / 11:00 – 12:00

Инструменты тестировщика 2026

Юлия Атлыгина

Зал 1 Tools/Frameworks



СИТУАЦИИ

- + Описываем шаги для тикета
- + Проверяем списки опций, ссылок, ...

Не открыты DevTools

Время, когда возникла проблема, чтобы быстро
найти в логах

Отчет о сессии исследовательского тестирования

Reload tab

<https://heisenbug.ru/archive/2025%20Autumn/schedule/table/>

11:58:24

Недавние ошибки

Console

Unhandled Promise rejection: Request failed with status code 500 ; Zone: <root> ; Task: Promise.then...

11:58:24

Копировать

Network

XHR GET

<https://embedded.jugru.org/api/v2/ratings/talk/>

Копировать

Сгенерировать промт

Настройки

Очистить данные

СИТУАЦИИ

- + Описываем шаги для тикета
- + Проверяем списки опций, ссылок, ...
- + Не открыты DevTools

Время, когда возникла проблема, чтобы быстро
найти в логах

Отчет о сессии исследовательского тестирования

Недавние ошибки

Network

HTTP 307

20:55:24

Копировать

Network

HTTP 307

20:55:18

Копировать

Сгенерировать промт

Настройки

Очистить данные

СИТУАЦИИ

- + Описываем шаги для тикета
- + Проверяем списки опций, ссылок, ...
- + Не открыты DevTools
- + Время, когда возникла проблема, чтобы быстро найти в логах

Отчет о сессии исследовательского тестирования

ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ

73

ОШИБКИ

12

Недавние действия

Click In BUTTON With Label: Support Chat

Support chat

21:04:52

Click In BUTTON With Label: Онлайн Чат

Онлайн чат

21:04:44

Thought for 31s >

Короткое описание: В ходе исследовательской сессии QA проверил навигацию по расписанию, карточкам докладов и спикеров, архиву конференций, FAQ, переключение языка, будущие конференции и сценарий покупки персонального билета.

Время: 25.03.2026 07:13:47 — 25.03.2026 07:30:22

Количество обнаруженных ошибок: 3


Подробная информация об ошибках:

Краткое описание: На странице покупки персонального билета возникает console error из-за отсутствующего модуля RegistrationTariffs

Описание:

Предусловия: Нет.

Шаги для воспроизведения:

1. Открыть расписание конференции и перейти в карточку любого доклада.
2. Из карточки доклада перейти в сценарий покупки персонального билета по кнопке «Купить билет».
3. На странице покупки билета выбрать вариант оплаты «Да, у меня карта российского банка».
4. Дождаться загрузки страницы тарифов. 

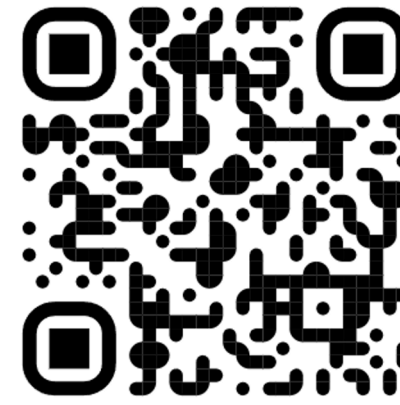
Фактический результат:

ОТЧЕТ О СЕССИИ ИССЛЕДОВАТЕЛЬСКОГО ТЕСТИРОВАНИЯ

Xray



Rapid Reporter



СИТУАЦИИ

- + Описываем шаги для тикета
- + Проверяем списки опций, ссылок, ...
- + Не открыты DevTools
- + Время, когда возникла проблема, чтобы быстро найти в логах
- + Отчет о сессии исследовательского тестирования

СИТУАЦИИ

- + Описываем шаги для тикета
- + Проверяем списки опций, ссылок, ...
- + Не открыты DevTools
- + Время, когда возникла проблема, чтобы быстро найти в логах
- + Отчет о сессии исследовательского тестирования

Документация

Настройка

Адреса для мониторинга

https://heisenbug.ru

Удалить

Добавить адрес

Отслеживаемые ошибки

- Сетевые ошибки
- Ошибки консоли
- UI-ошибки

Селекторы отслеживаемых UI-ошибок

div[id^="__error"]

Удалить

Добавить селектор

Язык ответов LLM

Язык ответа

Русский



ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ

73

ОШИБКИ

12

Недавние действия

Click In BUTTON With Label: Support Chat

Support chat

21:04:52

Click In BUTTON With Label: Онлайн Чат

Онлайн чат

21:04:44

Заголовок: Просмотр расписания конференции и переход к докладу из архива

Шаги:

1. Откройте новую вкладку и перейдите на главную страницу конференции.
2. Перейдите в раздел с расписанием конференции.
3. В расписании выберите нужный день мероприятия.
4. Найдите интересующий доклад и откройте его страницу.
5. Перейдите на страницу спикера из карточки доклада.
6. Отфильтруйте активности спикера по нужному году.
7. Выберите интересующий доклад из списка доступных выступлений.
8. На странице доклада откройте дополнительные материалы (например, презентацию).
9. При необходимости взаимодействуйте с элементами страницы (например, скрытие чата).
10. Закройте дополнительные элементы интерфейса для завершения просмотра.

СИТУАЦИИ

- + Описываем шаги для тикета
- + Проверяем списки опций, ссылок, ...
- + Не открыты DevTools
- + Время, когда возникла проблема, чтобы быстро найти в логах
- + Отчет о сессии исследовательского тестирования
- + Документация



ИНТЕГРАЦИИ

LLM

QA Trace v1.0.0

● Мониторинг

ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ

86

ОШИБКИ

13

Недавние действия

Click In A With Label: Закреть

Закреть

21:12:12

Click In BUTTON With Label: Скрыть Чат

Скрыть чат

21:12:11

WEBHOOK (N8N)

Смотреть запись



Следуй за белым кроликом

AI-агенты — это автономные системы, которые воспринимают окружение, принимают решения и выполняют действия для достижения целей. В докладе разберем конкретные технологии: как агенты используют RAG для сохранения памяти, работают с внешними инструментами через API и координируются в multi-agent системах.



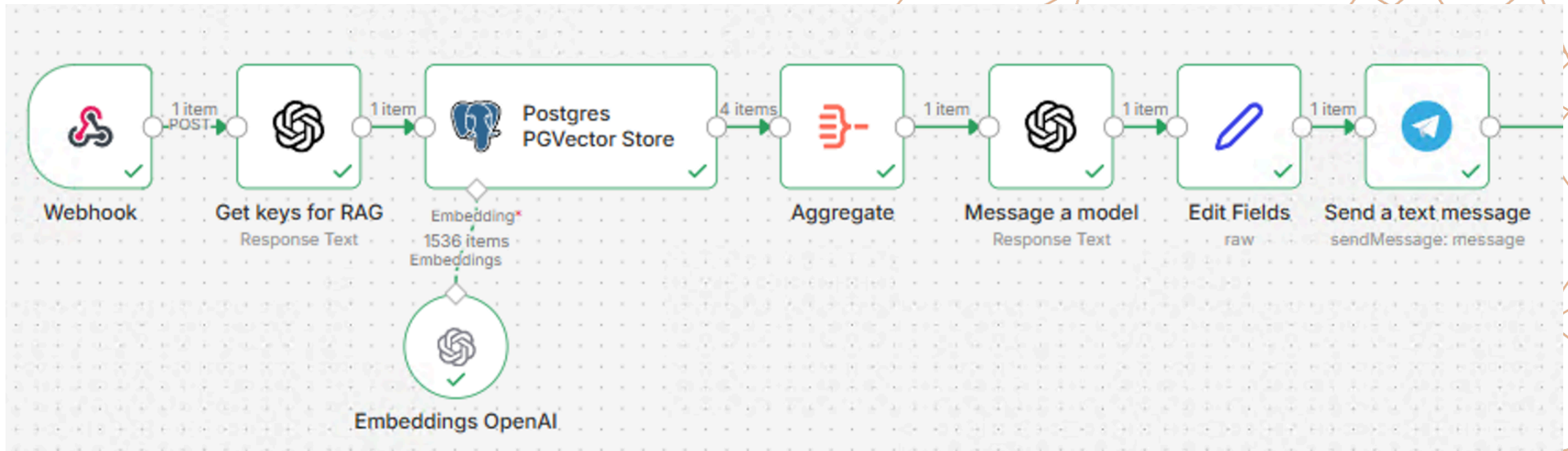
Артем Ерошенко
Allure Team



WEBHOOK (N8N)

```
{  
  "prompt": string,  
  "systemPrompt": string,  
  "userActions": Array<>,  
  "errors": Array<>,  
  "language": string,  
  "timestamp": number  
}
```

WEBHOOK (N8N)



Тикет:

Короткое описание:

HTTP 307 перенаправления и 500 ошибки API при загрузке данных профиля/доклада и рейтингов.

Описание:

Шаги для воспроизведения:

- 1) Откройте приложение и перейдите к профилю Юлии Атлыгиной на Heisenbug 2026 Spring.
- 2) Откройте доклад Инженерный vs Процессный подходы к тестированию и попробуйте загрузить данные доклада; наблюдайте HEAD-запросы к данным страницы, которые возвращают HTTP 307.
- 3) Попробуйте загрузить рейтинги доклада через API; наблюдается 500 (XHR GET рейтингов) и Unhandled Promise rejection: Request failed with status code 500; AxiosError: Request failed with status code 500.

- 4) Повторите аналогичные действия на другом докладе/архиве (например Autumn 2022); повторно вызываются запросы рейтингов и активности; также возникают 500 ошибки.
- 5) В процессе переключайтесь между вкладками (профиль, доклад, архив) и фиксируйте сетевые ошибки и консольные исключения; сохраняйте логи из разных вкладок.

Фактические исходники ошибок согласно журналу:

- HTTP 307 на HEAD-запросы к данным профиля/страницы архива, возвращающие перенаправление;
- 500 на XHR GET рейтингов и AxiosError: Request failed with status code 500;
- Unhandled Promise rejection: Request failed with status code 500 в консоли.

This message was sent automatically with n8n

13:27



ИСХОДНИКИ

QA TRACE

- Расширение в браузерах Firefox, Chrome и Яндекс
- Релизы в репозитории.
В режиме разработчика добавляем распакованное расширение



РАЗНЫЕ СИТУАЦИИ – ОДНО РЕШЕНИЕ

	Было	Стало
Баг-репорты	Полное описание занимало много времени Попадались баг-репорты только со скриншотом	Готовый баг-репорт (описание, скриншот, логи, ...) в несколько кликов (ну почти 😁)

РАЗНЫЕ СИТУАЦИИ – ОДНО РЕШЕНИЕ

	Было	Стало
Баг-репорты	Полное описание занимало много времени Попадались баг-репорты только со скриншотом	Готовый баг-репорт (описание, скриншот, логи, ...) в несколько кликов (ну почти 😅)
Возвращение в контекст	Почти невозможно (разве что сам вспомнишь)	Возвращаемся в контекст в один клик

РАЗНЫЕ СИТУАЦИИ – ОДНО РЕШЕНИЕ

	Было	Стало
Баг-репорты	Полное описание занимало много времени Попадались баг-репорты только со скриншотом	Готовый баг-репорт (описание, скриншот, логи, ...) в несколько кликов (ну почти 😄)
Возвращение в контекст	Почти невозможно (разве что сам вспомнишь)	Возвращаемся в контекст в один клик
Лог файл	Качаешь ? архивов, распаковываешь и повторяешь пока не найдёшь нужный лог файл	Качаешь 1 архив, и он точно тот самый

РАЗНЫЕ СИТУАЦИИ – ОДНО РЕШЕНИЕ

	Было	Стало
Баг-репорты	Полное описание занимало много времени Попадались баг-репорты только со скриншотом	Готовый баг-репорт (описание, скриншот, логи, ...) в несколько кликов (ну почти 😄)
Возвращение в контекст	Почти невозможно (разве что сам вспомнишь)	Возвращаемся в контекст в один клик
Лог файл	Качаешь ? архивов, распаковываешь и повторяешь пока не найдёшь нужный лог файл	Качаешь 1 архив, и он точно тот самый
Отчёт о сессии исследовательского тестирования	Почти не было отчётов	После каждой сессии есть отчёт

СПАСИБО ЗА ВНИМАНИЕ!

