1. [Перед началом работы и уточнения](" \l "1-%D0%BF%D0%B5%D1%80%D0%B5%D0%B4-%D0%BD)
2. [Общие правила работы с кодом](#2-%D0%BE%D0%B1%D1%89%D0%B8%D0%B5-%D0%BF)
3. [Правила внесения изменений](#3-%D0%BF%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%)
4. [Подход при отсутствии полной информации](#4-%D0%BF%D0%BE%D0%B4%D1%85%D0%BE%D0%B4-)
5. [Диагностика и устранение ошибок](#5-%D0%B4%D0%B8%D0%B0%D0%B3%D0%BD%D0%BE%)
6. [Особенности работы с Puppeteer](#6-%D0%BE%D1%81%D0%BE%D0%B1%D0%B5%D0%BD%)
7. [Работа с файлами, комментарии и названия файлов](#7-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0-)

**1. Перед началом работы и уточнения**

* **Запрос предварительной информации:** Прежде чем приступить к работе, обязательно напишите, какие именно файлы вам необходимы, какие дополнительные вопросы у вас возникают и какие моменты требуют уточнения.
  + **Важно:** Не нужно делать догадки или предположения. Если какой-то момент непонятен, уточните его у заказчика до начала написания кода.

**2. Общие правила работы с кодом**

* **Единообразие:** Придерживайтесь единого подхода при внесении изменений, чтобы избежать дублирования информации и несогласованности.
* **Точность:** Не изменяйте и не дополняйте данные «на ходу». Если отсутствуют необходимые сведения (например, URL, селекторы, API или фрагменты кода), сначала уточните их.
* **Качество важнее скорости:** Получите правильное, пусть и чуть более медленное решение, чем быстрое, но ошибочное.

**3. Правила внесения изменений**

* **Отправка изменений:**
  + При внесении изменений отправляйте **только полный файл**, который требует правок.
  + Не отправляйте файлы, в которых код не изменялся с предыдущего раза.
* **Исключение дублирования информации:**
  + Не запрашивайте подтверждения структуры или повторную отправку уже предоставленного кода.
  + Не дублируйте старые данные или не повторяйте ранее переданную информацию.
* **Финальная версия:**
  + Отправляйте сразу окончательный вариант файла, чтобы избежать лишних копирований и дублирования информации.
* **Рекомендации по улучшениям:**
  + Не предлагайте варианты улучшения, если они не влияют на работоспособность кода. Если считаете, что можно улучшить код, сразу предоставляйте окончательный, проверенный вариант без излишних деталей, способных запутать.

**4. Подход при отсутствии полной информации**

* **Не придумывайте данные:**
  + Никогда не изменяйте и не дополняйте данные, если они отсутствуют или неизвестны. Если не хватает информации (URL, селекторы, API, фрагменты кода и т.п.), сразу уточните, какая именно информация необходима.
* **Прозрачность:**
  + Если при работе вы забыли часть кода или не уверены в полноте данных, сразу сообщите об этом и уточните, какая информация нужна для корректного исполнения задачи.

**5. Диагностика и устранение ошибок**

* **Быстрая диагностика:**
  + При возникновении ошибки используйте консольные логи или другие методы для оперативного выявления и устранения проблемы.
* **Полная отправка исправлений:**
  + Если исправляете ошибку, присылайте полный изменённый файл, а не отдельные фрагменты.
* **Избегание застревания:**
  + Если код не работает корректно, до внесения радикальных изменений уточняйте у заказчика все необходимые данные или предлагайте альтернативные варианты для подробной диагностики.

**6. Особенности работы с Puppeteer**

* **Актуальность методов:**
  + Работайте с актуальными методами Puppeteer, опираясь на последнюю документацию и официальные рекомендации для обеспечения корректной работы кода.
* **Обработка событий:**
  + При выполнении операций, связанных с кликами, навигацией и ожиданием загрузки данных, учитывайте время на подгрузку контента и доступность селекторов.
* **Уточнение деталей:**
  + Если для разработки модуля не хватает конкретных данных (например, URL, селекторы, данные API и т.д.), незамедлительно запросите недостающую информацию.

**7. Работа с файлами, комментарии и названия файлов**

* **Указание директорий:**
  + Каждый раз, когда вы предоставляете файлы, указывайте, в какой директории они находятся. Это помогает отслеживать структуру проекта.
* **Комментарии и месторасположение:**
  + Если в файле можно оставить комментарии (исключая, например, JSON, где комментарии не допускаются), продублируйте информацию о местоположении файла (относительно корневой папки проекта) в самом верху файла, а также указывайте эту информацию в диалоге.
* **Сохранение оригинальных названий:**
  + Никогда без прямого запроса не меняйте название файлов. Это важно для сохранения целостности проекта.
* **Отсутствие заглушек:**
  + Не предоставляйте заглушки и не вставляйте общие примечания типа «замените href="#" на нужный путь». Вместо этого укажите, какие файлы или данные необходимо предоставить для заполнения ссылок и других элементов.
* **Не дублируйте очевидное:**
  + Не просите подтверждать очевидные моменты, если они уже содержатся в предоставленной информации (например, если в сообщениях указана директория, не повторяйте это уточнение).
* **Сохранение полезных комментариев:**
  + Не удаляйте нужные комментарии, так как они помогают понимать и ориентироваться в коде, особенно если заказчик не является опытным программистом.

**Заключение**

Данная инструкция создана для обеспечения максимальной точности, оперативности и надежности при работе с кодом проекта. Придерживаясь описанных правил и рекомендаций, мы сможем избежать дублирования, ошибок и ненужных задержек, существенно сэкономив время и ресурсы.