**Site Failures**

* HTTP response status codes
* JavaScript errors
* Style issues
* Performance concerns
* Mixed content warning

PS: Compare a success and a failure

Ошибка на стороне клиента – это HTML, CSS, JS, валидация на полях

**HTML**

* + Не указано на кнопке ahref, и по этому не осуществляется перенаправление
  + …

**CSS**

* Когда кнопка не корректного размера, поле не корректного размера
* …

**JS**

* + Когда к кнопке не привязан скрипт, и она не срабатывает
  + Когда скрипт не может найти элемент (например при пагинации на следующую страницу, где нет элементов)

**Валидация**

* Когда не срабатывает валидация на поле при вводе не корректных данных
* …

1. **HTTP response status codes**

Все ошибки с кодами - это ошибки на стороне сервера. Можно посмотреть логи сервера для локализации

**DevTools > Network** или **Fiddler**, смотрим что все запросы выполнились корректно (нет красных запросов)



*Если упала 400 ошибка в Response*

**400** - значит мы сами не корректно составили запрос, но сервер умеет ее обрабатывать

**401** - не авторизованы,

**403** - у нас нет прав на изменение,

**404** - не корректно указан ендпоинт

*Если упала 500 ошибка в Response*

**500** - Серверу не удалось обработать запрос к сайту. Возможных причин для этого может быть много, но сузить их круг можно, восстановив последовательность ваших действий перед сообщением об ошибке. Также изучите само сообщение: комментарий «Internal Server Error» говорит о проблемах с файлом .htaccess, текст «HTTP ERROR 500» — о проблемах со скриптами, а текст «PHP Parse error: syntax error, unexpected» или «Internal Server Error nginx» — о неполадках в CMS.

* Проверьте сайт, созданный с помощью CMS, на наличие проблем с плагинами или ошибок в коде. В этом вам могут лог-файлы. При обнаружении проблемного плагина обновите его или верните прежнюю версию. Если это не помогло, откажитесь от него. Если ошибка произошла после обновления CMS, проведите обновление повторно.
* Посмотрите файл .htaccess на предмет ошибок в командах. Закомментируйте директиву Options, поставив перед ней решётку: если после этого ошибка 500 перестанет появляться, значит, есть нарушения в синтаксисе и в описании команд.
* Убедитесь, что права доступа к файлам, папкам и скриптам выставлены верно. Для папок рекомендуется значение 755, для скриптов — 600, а для других файлов — 644. При других вариантах прав доступ к сайту может блокироваться в целях безопасности.
* Проверьте, всё ли в порядке со скриптами. Возможно, какой-то из скриптов слишком медленный или время ожидания ответа от сервера слишком мало. Если при просмотре лог-файлов выяснится, что какой-то из скриптов незапланированно требует слишком много памяти, оптимизируйте его или удалите. А если обнаружится, что какой-то из скриптов вовсе не запускается, убедитесь, что функция прописана верно, поддерживается сервером и соответствует используемой версии PHP.
* Отдельно обратите внимание на CGI-скрипты: вероятно, строки в них имеют не те окончания, что исправляется загрузкой скриптов через FTP в режиме ASCII. Также некорректная работа CGI-скриптов может быть причиной ошибок в HTTP-заголовках, что тоже приводит к ошибке 500. Либо же имеются ошибочные директивы, предназначенные для работы со скриптами.

**501** - Сервер не поддерживает функциональности для обработки запроса (возможно мы указали не тот метод - GET/POST/PUT/DELETE или отправили в body что сервер не ожидал получить)

**502** - Ошибочный шлюз. Разбираться с этой ошибкой нужно лишь тогда, когда она появляется регулярно. А говорит она о перегруженности сервера или о неполадках в его работе, в связи с чем он посылает недопустимые для продолжения работы ответы.

* Перезагрузите страницу. Зайдите на любой другой сайт, которой точно должен работать в данный момент. Это поможет узнать, есть ли у вас доступ к интернету в принципе. Если доступ есть, очистите файлы cookies в браузере, а затем посетите сайт снова.
* Убедитесь, что на ваш сайт не совершается DDoS-атака. В противном случае обратитесь к хостинг-провайдеру.
* Если на вашем ресурсе фиксируется значительный рост посещаемости, то подберите более продвинутые условия хостинга, чтобы ошибка не появлялась вновь.
* Проверьте нагрузку на сервер. Если лимит превышается, необходимо увеличить объём оперативной памяти.
* Посмотрите настройки сервера. Возможными поводами для появления ошибки 502 могут быть:
  + неполадки после установки обновлений;
  + превышение лимитов на число обращений к внешним ресурсам и на время ответа сервера;
  + некорректные лимиты в файлах конфигурации ini;
  + превышение лимита на число php-cgi-процессов;
  + недостаточная оптимизация скриптов;
  + недостаточная оптимизация запросов;
  + неправильная работа модулей (если ошибка возникает при обращении к скриптам конкретного расширения).
* Если ошибка продолжает появляться и если вы пользуетесь виртуальным хостингом, уточните у хостинг-провайдера, не создают ли другие сайты на сервере чрезмерную нагрузку

**503** - Сервер лежит. Сервер не работает из-за перегрузок. Либо же происходит плановая перезагрузка или отключение сервера: в этом случае вместе с сообщением об ошибке после слов «Retry-After» должно отображаться время, когда сервер вернётся в работу. Если же ошибка 503 появляется часто и не по причине плановых работ, то это говорит о неполадках, которые следует устранить.

Каждому аккаунту на сервере выделено определенное количество процессов-рабочих, обрабатывающих запросы пользователей. Запросы поступают на сервер и становятся в очередь. Легкие запросы обрабатываются быстро, а тяжелые, проблемные — медленно, тормозя продвижение очереди. Когда длина очереди достигает определенной величины, сервер перестает принимать новые запросы, возвращая ошибку 503 (Service Temporarily Unavailable — сервис временно недоступен).

* Сначала просто подождите. Возможно, причина в длинной очереди запросов к серверу, что не требует вмешательства.
* Как и в случае с ошибкой 502, удостоверьтесь, что на сайт не производится DDoS-атака.
* Если используется связь с удалённым сервером, убедитесь, что она стабильная, а тайм-аут ожидания ответа невысокий.
* Проверьте, не слишком ли активно посещают ваш сайт поисковые роботы. Если это имеет место быть, ограничьте их активность.
* Удалите тяжёлые или вовсе ненужные плагины и компоненты.
* Если возможно, оптимизируйте подгрузку файлов сайта, чтобы снизить число запросов.
* Организуйте передачу больших статичных файлов напрямую, а не через скрипты.
* Оптимизируйте почтовую рассылку: распределяйте отправку писем по времени, запускайте рассылку в часы наименьшей нагрузки.
* Оптимизируйте SQL-запросы, выявите самые медленные из них с помощью лог-файлов.

**504** - Сервер не дождался ответа от вышестоящего сервера

* Перезагрузите страницу, убедитесь в стабильности работы сетевых устройств.
* Как и в предыдущих случаях, проверьте работу скриптов. Важно, чтобы они выполнялись не слишком долго, а внешние соединения происходили успешно.
* При чрезмерной нагрузке на сервер увеличьте его ресурсы или оптимизируйте сайт.
* Если возможно, увеличьте время ожидания при использовании nginx как прокси-сервера для Apache. Для этого добавьте эти строки в блоке server в файле nginx.conf:
  + proxy\_connect\_timeout 600;
  + proxy\_send\_timeout 600;
  + proxy\_read\_timeout 600;
  + send\_timeout 600;
* Если у вас нет возможности менять настройки сервера, обратитесь к хостинг-провайдеру.

**507** - не хватает места в БД

**Как локализовать в чем причина ошибки**

* Проверить Headers и Body запроса и ответа. Вдруг в ответе придет не то значение header, которое мы ожидаем
* Проверить не пропущены ли обязательные поля

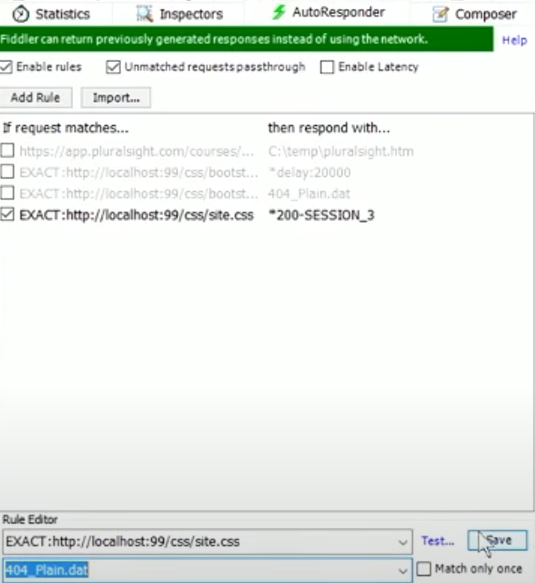
**DevTools > Console**

Проверить ошибки

**Fiddler**

*Что бы сымитировать на фронте проблему, когда не доступен Backend*

* Перехватить код ответа сервера > поменять его
* Перетаскиваем нужный запрос в Fiddler > Autoresponder. Enable Rules. После меняем для этого запроса код ответа



*Что бы проверить валидацию на стороне сервера*

Заполнить поля корректными данными и отправить запрос > перехватить его на Fiddler и поменять данные на не корректные > посмотреть код ответа

1. **Style Issues**

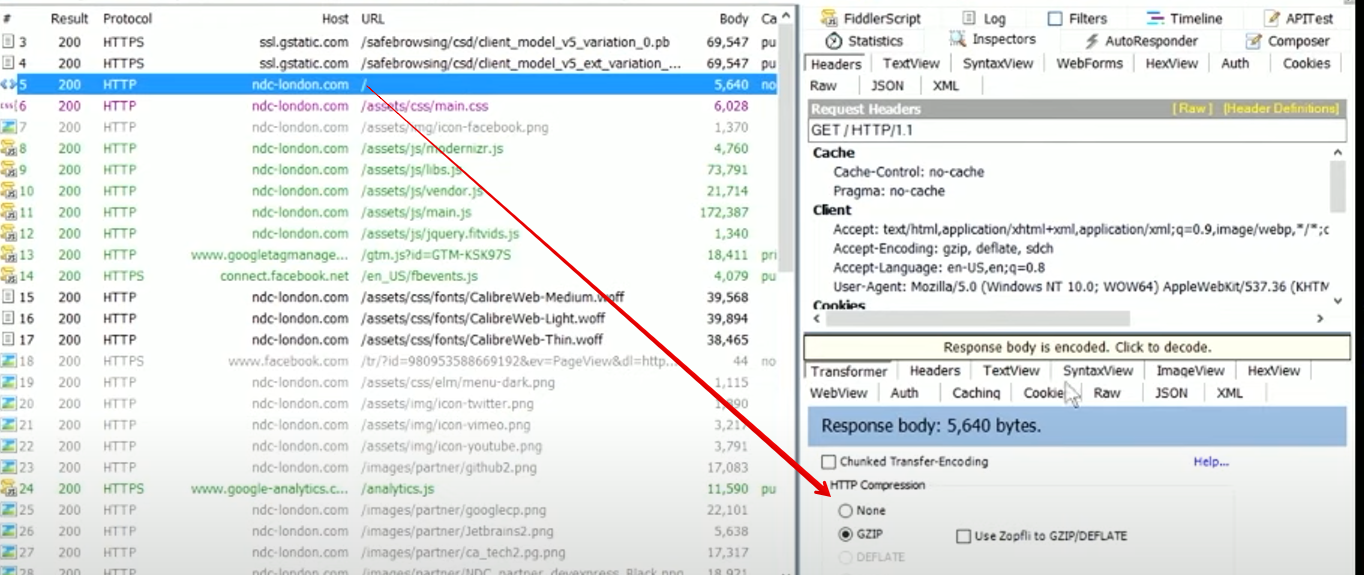
**DevTools > Elements**

Выбираем нужный элемент в инспекторе. После смотрим в Styles и в Computed. Какие стили применились, а какие нет. Какие стили перекрываются другими

1. **Performance concerns**

**Fiddler > Inspectors**

Смотрим используется ли HTTP Compression для статичных и динамичных файлов



**Fiddler > Inspectors**

Фильтруем по размеру Body и смотрим какие файлы самые большие

Смотрим закешиваются ли файлы

PS: Compare a success and a failure

Сравнивать запросы на тех устройствах, где он проходят успешно, с запросами которые проходят не успешно на других устройствах (или функциональности в разных модулях). Например в 1 модуле запрос проходит успешно, но эта же функциональность в другом модуле (устройстве (Android/iOS,..)) не выполняется. Вполне возможно, что ключевые вещи в запросах могут отличаться (дефект)

**Например:**

Приложение определяет локацию куда доставлять товар. Так работает на iPhone



Так работает на других устройствах



Из за этого на iPhone не будет отрабатывать эта функциональность, так как по каким то причинам в запросе передается меньше десятичных цифр

**Who modified that?**

Если на странице появляется html элемент спустя какое то время, то можно посмотреть что внесло изменения, дабы он появился

**DevTools > Inspect**

Выбираем наш элемент

**Right Click > Break on > Subtree modifications / Attributes modifications**

Теперь когда элемент будет изменен, то мы попадает в дебагер, где видно что делает эти изменения

**Who requested that?**

**DevTools > Network > Initiator** – здесь показаны какой ресурс запросил тот или иной запрос