# Условия домашки

|  |
| --- |
| Чтобы выполнить основную часть домашнего задания, используйте материалы урока – в них вы найдете все необходимое для решения. |

|  |
| --- |
| 💡 В этой домашке мы продолжим разрабатывать приложение «Лента комментариев». |

## В каком виде сдавать работу

Сделайте pull request в свой репозиторий с прошлой домашкой с результатами выполнения этой домашки. На момент сдачи работы PR должен быть открытым.

После того как наставник примет работу, смержите свой пул-реквест в main-ветку.

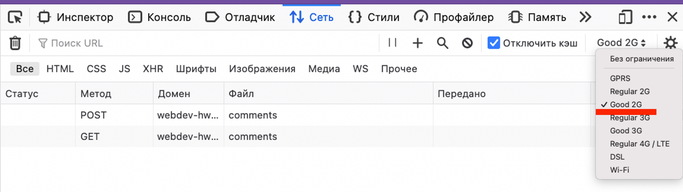
## Задание

В этой домашке мы покажем пользователю приложения состояние, когда данные загружаются:

* Я как пользователь при запуске приложения понимаю, что данные загружаются и мне нужно немного подождать.
* Я как пользователь при добавлении комментария понимаю, что мне нужно подождать и не могу случайно отправить коммент повторно во время загрузки предыдущего.

Начните с того, что переделайте все вызовы then на цепочки промисов, вынесите код получения списка комментариев в отдельную функцию — так вам будет проще выполнить задачу.

Используйте «замедление интернета» в консоли разработчика на вкладке сеть, чтобы отладить работу лоадеров:



## Дополнительное задание

|  |
| --- |
| Приступайте к дополнительным заданиям только после того, как закончите с основной частью.  Наставники примут и проверят задание и без дополнительной части, но ее выполнение поможет вам лучше разобраться с темой и упростит понимание следующих и предыдущих тем курса. |

Реализуем сценарий

* Я как пользователь понимаю, что лайк ставится/снимается и мне нужно подождать, пока он поставится.

*У нас пока нет API лайков, поэтому сымитируем запрос в API с помощью функции delay.*

|  |
| --- |
| 💡 Функция delay возвращает промис, который выполнится через указанное количество миллисекунд. |

Скопируйте код функции и вызовите ее вместо fetch-функции во время обработки клика по лайку.

|  |
| --- |
| **Код функции delay и пример использования**  // Функция для имитации запросов в API  // Не смотрите особо на внутренности, мы разберемся с этим позже  function delay(interval = 300) {  return new Promise((resolve) => {  setTimeout(() => {  resolve();  }, interval);  });  }  // Пример использования:  delay(2000).then(() => {  comment.likes = comment.isLiked  ? comment.likes - 1  : comment.likes + 1;  comment.isLiked = !comment.isLiked;  comment.isLikeLoading = false;  renderComments();  }); |

Для анимации иконки лайка можете использовать готовый CSS, а также можете придумать свой способ анимировать сердечко:

|  |
| --- |
| **CSS-класс для анимации загрузки лайка**  @keyframes rotating {  from {  transform: rotate(0deg);  }  25% {  transform: rotate(30deg);;  }  75% {  transform: rotate(-30deg);;  }  to {  transform: rotate(0deg);  }  }  .-loading-like {  animation: rotating 2s linear infinite;  } |

Как должно работать приложение в результате:

### Условия проверки и выполнения задания

* В коде приложения нет вложенных then, везде используются цепочки промисов.
* Целевые сценарии выполнены:
  + - при запуске приложения отображается сообщение о загрузке списка комментариев, при добавлении нового комментария это сообщение не отображается;
    - при добавлении нового комментария, на время выполнения запроса, скрывается форма добавления, вместо нее отображается надпись «Комментарий добавляется».