

Курс по JavaScript. 2021

19 июн 2022, 18:07:55

старт: 27 сен 2021, 14:53:11

начало: 1 сен 2020, 00:00:00

Мы очень хотим, чтобы код вы написали сами, а не пользовались внешними библиотеками.

Основное задание

После криминальной интриги с участием друзей Аркадий возвращается в мрачные стены университета — время слушать лекцию и впитывать новые знания.

Вас ждут студенты, готовые внимать каждому слову преподавателя:

```
const students = {
  Sam: {
    focus: 100,
    wisdom: 50
  },
  Daria: {
    focus: 100,
    wisdom: 60
  }
};
```

И преподаватель Сергей, который вот-вот начнёт лекцию и покажет первый слайд с бесценной информацией.

```
lecturer.on('begin', students.Sam, function() {
  // Внимательно слушаем преподавателя
  this.focus += 10;
});

lecturer.on('slide', students.Daria, function() {
  // И впитываем мудрость с каждым слайдом
  this.wisdom += 10;
});
```

Студента можно подписать на событие, производимое преподавателем (например, начало лекции или показ нового слайда) — то есть указать, какая функция должна быть вызвана при наступлении этого события.

Ваша задача — реализовать несколько методов:

- подписка на событие — `on`;
- отписка от события — `off`;
- вызов события — `emit`.

Чтобы задание не казалось совсем простым, вам также нужно реализовать поддержку пространства имён для событий.

```
// ...
```

```
lecturer.on('slide', students.Sam, function() {
  // И впитываем мудрость с каждым слайдом
  this.wisdom += 10;
});

lecturer.on('slide.funny', students.Sam, function() {
  this.wisdom -= 5;
});
```

Дополнительные условия и ограничения:

- События должны возникать в том порядке, в котором на них подписывались;
- На одно событие с одинаковыми объектами и обработчиками можно подписаться неограниченное количество раз.
- Обработчики вызываются в порядке подписки;
- Пространства имён разделены только точкой:
 - на событие `slide.funny` произойдут события `slide.funny` и `slide` (именно в таком порядке); на событие `slidee` произойдет `slidee`, но не `slide`;

- отписка от `slide.funny` отписывает только от него;
- отписка от `slide` отписывает и от `slide`, и от `slide.funny`.

Мы выложили пример того, как можно работать с вашим кодом, под условием задачи. А заготовку для того, чтобы реализовать свой код, вы можете найти [здесь](#).

Дополнительное задание

Необходимо реализовать два дополнительных метода эмиттера. Оба метода работают аналогично `on`, но обладают некоторыми особенностями:

- `several` — подписывает на первые `n` событий;
- `through` — подписывает на каждое `n`-ое событие, начиная с первого.

При отрицательном или нулевом значении `through` и `several` начинают работать, как `on`.

Пример работы этих методов вы можете отыскать всё в том же примере внизу.



Полезные ссылки

- [Продвинутая работа с функциями](#)
- [Метод `call` для функций](#)
- [Метод `apply` для функций](#)
- [Метод `bind` для функций](#)

Пример:

```
'use strict';

const { getEmitter } = require('./emitter');

let students = {
  Sam: {
    focus: 100,
    wisdom: 50
  },
  Sally: {
    focus: 100,
    wisdom: 60
  },
  Bill: {
    focus: 90,
```

```
wisdom: 50
},
Sharon: {
  focus: 110,
  wisdom: 40
}
};

let lecturer = getEmitter();

// С началом лекции у всех резко повышаются показатели
lecturer
.on('begin', students.Sam, function () {
  this.focus += 10;
})
.on('begin', students.Sally, function () {
  this.focus += 10;
})
.on('begin', students.Bill, function () {
  this.focus += 10;
  this.wisdom += 5;
})
.on('begin', students.Sharon, function () {
  this.focus += 20;
});

// На каждый слайд внимательность падает, но растет мудрость
lecturer
.on('slide', students.Sam, function () {
  this.wisdom += Math.round(this.focus * 0.1);
  this.focus -= 10;
})
.on('slide', students.Sally, function () {
  this.wisdom += Math.round(this.focus * 0.15);
  this.focus -= 5;
})
.on('slide', students.Bill, function () {
  this.wisdom += Math.round(this.focus * 0.05);
  this.focus -= 10;
})
.on('slide', students.Sharon, function () {
  this.wisdom += Math.round(this.focus * 0.01);
  this.focus -= 5;
});

// На каждый веселый слайд всё наоборот
lecturer
.on('slide.funny', students.Sam, function () {
  this.focus += 5;
  this.wisdom -= 10;
})
.on('slide.funny', students.Sally, function () {
  this.focus += 5;
  this.wisdom -= 5;
})
.on('slide.funny', students.Bill, function () {
  this.focus += 5;
  this.wisdom -= 10;
})
.on('slide.funny', students.Sharon, function () {
  this.focus += 10;
  this.wisdom -= 10;
});

// Начинаем лекцию
lecturer.emit('begin');
// Sam(110,50); Sally(110,60); Bill(100,55); Sharon(130,40)

lecturer
.emit('slide.text')
.emit('slide.text')
.emit('slide.text')
```

```
.emit('slide.funny');
// Sam(75,79); Sally(95,118); Bill(65,63); Sharon(120,34)

lecturer
  .off('slide.funny', students.Sharon)
  .emit('slide.text')
  .emit('slide.text')
  .emit('slide.funny');
// Sam(50,90); Sally(85,155); Bill(40,62); Sharon(105,37)

lecturer
  .off('slide', students.Bill)
  .emit('slide.text')
  .emit('slide.text')
  .emit('slide.text');

lecturer.emit('end');
// Sam(20,102); Sally(70,191); Bill(40,62); Sharon(90,40)

// Пример работы дополнительного задания
students = {
  Sam: {
    focus: 100,
    wisdom: 50
  },
  Bill: {
    focus: 90,
    wisdom: 50
  }
};

lecturer = getEmitter()
  .several(
    'begin',
    students.Sam,
    function () {
      this.focus += 10;
    },
    1
  )
  .several(
    'begin',
    students.Bill,
    function () {
      this.focus += 10;
      this.wisdom += 5;
    },
    1
  )
// На Сэма действуют только нечетные слайды
.through(
  'slide',
  students.Sam,
  function () {
    this.wisdom += Math.round(this.focus * 0.1);
    this.focus -= 10;
  },
  2
)
// Концентрации Билла хватит ровно на 4 слайда
.several(
  'slide',
  students.Bill,
  function () {
    this.wisdom += Math.round(this.focus * 0.05);
    this.focus -= 10;
  },
  4
)
.on('slide.funny', students.Sam, function () {
  this.focus += 5;
  this.wisdom -= 10;
```

```
    })
    .on('slide.funny', students.Bill, function () {
        this.focus += 5;
        this.wisdom -= 10;
    });

    lecturer.emit('begin');
    // Sam(110,50); Bill(100,55)

    lecturer
        .emit('slide.text')
        .emit('slide.text')
        .emit('slide.text')
        .emit('slide.funny');
    // Sam(95,61); Bill(65,63)

    lecturer
        .emit('slide.text')
        .emit('slide.text')
        .emit('slide.funny');
    // Sam(80,70); Bill(70,53)
```

Набрать здесь

Отправить файл

```
1  /**
2   * Возвращает новый emitter
3   * @returns {Object}
4   */
5  function getEmitter() {
6      let map = new Map();
7
8      return {
9          /**
10           * Подписаться на событие
11           * @param {String} event
12           * @param {Object} context
13           * @param {Function} handler
14           * @param {Number} severalCount
15           * @param {Number} throughCount
16           */
17          on: function (event, context, handler, severalCount = null, throughCount = null) {
18              if (!map.has(event)) {
19                  map.set(event, [])
20              }
21
22              map.get(event).push({
23                  context: context,
24                  handler: handler,
25                  severalCount: severalCount,
26                  throughCount: throughCount,
27                  count: 0
28              })
29
30              return this;
31          },
32
33          /**
34           * Отписаться от события
35           * @param {String} event
36           * @param {Object} context
37           */
38          off: function (event, context) {
```

Отправить

Предыдущая

Следующая