

stats.js

- Исходный код модуля
- Выявление скачков одного параметра
 - 1. Инициализация обрабатываемого параметра
 - 2. Получение массива параметров WebRTC статистики и проверка параметра
- Работа со списком проверяемых параметров
 - 1. Добавление параметра в список
 - 2. Удаление параметра из списка
 - 3. Получение массива параметров WebRTC статистики и проверка списка параметров

В модуль `stats.js` вынесены функции проверки параметров WebRTC статистики и выявления скачков параметров выше заданного предела

Исходный код модуля

Для анализа исходного кода возьмем версию модуля `stats.js`, которая находится [здесь](#)

Выявление скачков одного параметра

1. Инициализация обрабатываемого параметра

`Threshold()` [code](#)

Здесь:

- запоминается имя параметра и максимально допустимый предел его изменения
- создается фильтр Калмана для этого параметра

```
const threshold = {
  parameter: parameter,
  maxLeap: maxLeap,
  filter: SFU.createFilter(),
  previousValue: -1,
  ...
}
```

2. Получение массива параметров WebRTC статистики и проверка параметра

`Threshold.isReached()` [code](#)

Здесь:

- текущее значение параметра проходит через фильтр Калмана
- если изменение значения превышает допустимую величину, фиксируется скачок параметра

```
isReached: function(stats) {
  let hasLeap = false;
  if (stats && parameter in stats) {
    let value = threshold.filter.filter(stats[parameter]);
    if (threshold.previousValue > -1) {
      if (Math.round(Math.abs(value - threshold.previousValue)) > maxLeap) {
        hasLeap = true;
      }
    }
    threshold.previousValue = value;
  }
  return hasLeap;
}
```

Работа со списком проверяемых параметров

1. Добавление параметра в список

Thresholds.add() [code](#)

```
add: function(parameter, maxLeap) {
  if (!thresholds.thresholds[parameter]) {
    thresholds.thresholds[parameter] = new Threshold(parameter, maxLeap);
  }
},
```

2. Удаление параметра из списка

Thresholds.remove() [code](#)

```
remove: function(parameter) {
  if (thresholds.thresholds[parameter]) {
    delete thresholds.thresholds[parameter];
  }
},
```

3. Получение массива параметров WebRTC статистики и проверка списка параметров

Thresholds.isReached() [code](#)

```
isReached: function(stats) {
  let result = false;
  Object.keys(thresholds.thresholds).forEach((key) => {
    result = result || thresholds.thresholds[key].isReached(stats);
  });
  return result;
}
```