
DuelJS Documentation

Release 1.2.7

October 16, 2016

1	User Documentation	3
1.1	Getting Started	3
1.2	API:duel	4
1.3	API:channel object interface	7

Note: Documentation still in progress.

Welcome to the DuelJS JavaScript library documentation.

User Documentation

1.1 Getting Started

Getting started with DuelJS.

1.1.1 Requirements

With DuelJS you no need to any requirements - only vanilla js and modern web browser. Don't try to use it with node.js without browser emulators.

1.1.2 Installing DuelJS

You can install it using bower or simple copy duel.js main file into your site or even clone git repository.

Different ways:

- Bower package: `bower install duel --save`
- Git repo: `git clone https://github.com/studentIvan/dueljs.git`
- Main file: `duel.min.js`

Put it into your webpage: `<script type="text/javascript" src="path/to/duel.min.js"></script>`

So we've got all the set up out of the way. Let's write some simple code.

```
<script type="text/javascript">
var channel = duel.channel('my_first_channel');
// now you have opened some channel, this tab is connected to it

setInterval(function () {
  if (channel.currentWindowIsMaster()) {
    // here you checking is this tab active (in focus) or not
    // you can use alternative syntax: if (window.isMaster()) { ...

    document.title = 'Master ' + duel.getWindowID();
    // duel.getWindowID returns a unique browser tab id

  } else {
    document.title = 'Slave ' + duel.getWindowID();
  }
}
```

```
}, 100);  
</script>
```

1.1.3 Broadcasting

When your tab had some channel (e.g. **my_first_channel** from previous section) you can do cross-tab broadcasting.

Use simple commands:

- **channel.broadcast('event_name', a,r,g,u,m,e,n,t,s...);**
 - send event command to all another tabs in channel. Alias: **channel.emit**.
- **channel.on('event_name', function (a,r,g,u,m,e,n,t,s...) { do here what you want });**
 - define watcher for event_name.
- **channel.off('event_name');**
 - remove event_name watcher.
- **channel.once('event_name', function (a,r,g,u,m,e,n,t,s...) { do here what you want });**
 - define watcher for event_name, do it only one time and remove.

1.1.4 Articles and examples

- (russian)
- [Youtube-player integration demonstration \(russian\)](#)

If you still need more docs go to API section

1.2 API:duel

1.2.1 duel

Main global object.

- global
- type: object

1.2.2 duel.activeChannels

Collect inside all active channels.

- public
- type: array
- default: []

1.2.3 `duel.useStorageEvent`

Optional configuration. Storage event improves performance in modern browsers.

- public
- type: boolean
- default: true (IE - false)

1.2.4 `duel.noWarnings`

Optional configuration. You can turn it to false for debug.

- public
- type: boolean
- default: true

1.2.5 `duel.isLocalStorageAvailable()`

Common function for localStorage detection.

- public
- type: function
- returns: boolean

1.2.6 `duel.channel(name:string)`

Creates a new channel or join to existed. Hint: `a = duel.channel('x')` and `b = duel.channel('x')` will be linking on ONE object.

- public
- type: function
- returns: object (duel.DuelAbstractChannel inheritor)

1.2.7 `duel.makeCurrentWindowMaster()`

Take the all channels in current window and set current window as master for all of them.

- public
- type: function

1.2.8 `duel.clone(obj:object)`

Common function for copy objects.

- public
- type: function
- returns: object

- throws error on unsupported type

1.2.9 `duel._generateWindowID()`

Generates, sets up and returns new window/tab ID.

- private
- type: function
- returns: number

1.2.10 `duel.getWindowID()`

Get unique window/tab ID.

- public
- type: function
- returns: number

1.2.11 `duel.addEvent(el:object, type:string, fn:function)`

Cross-browser addEvent method.

- public
- type: function

1.2.12 `duel.storageEvent(event:object)`

Finds the specific channel and execute event on it. **event** object contains **key:string** and **newValue:string**. **newValue** is a JSON string, which contains **channelName:string** and **triggerDetails:object**. **triggerDetails** contains **name:string** and **args:array**.

- public
- type: function

1.2.13 `duel.DuelAbstractChannel`

Abstract class for possible duel channels. DuelJS probably will support another channels besides `duel.DuelLocalStorageChannel` in future.

- abstract
- type: function
- returns: object

1.2.14 `duel.DuelLocalStorageChannel`

Channel class for work with localStorage.

- abstract
- type: function
- returns: object

1.2.15 `duel.DuelFakeChannel`

Channel class for work without localStorage.

- abstract
- type: function
- returns: object

1.2.16 `window.isMaster()`

Take first channel in current window and check is it master or not

Standard window object spreading method. Looks like syntax sugar for `channelObject.currentWindowIsMaster()`

- public
- type: function
- returns: boolean

1.3 API:channel object interface

1.3.1 `channel.getItem(varName:string)`

Get jsonify var content from storage.

- public
- type: function
- returns: mixed

1.3.2 `channel.setItem(varName:string, value:mixed)`

Set storage variable content.

- public
- type: function

1.3.3 `channel.removeItem(varName:string)`

Remove var from storage.

- public
- type: function

1.3.4 `channel.getName()`

Returns the name of this channel.

- public
- type: function
- returns: string

1.3.5 `channel.setCurrentWindowAsMaster()`

Makes current window/tab as master in this channel.

- public
- type: function
- returns: boolean

1.3.6 `channel.currentWindowIsMaster()`

Checks the master state of this channel.

- public
- type: function
- returns: boolean

1.3.7 `channel.broadcast(trigger:string[, arguments:arguments])`

Emits broadcasting. Hint: only master can sends broadcast.

- public
- type: function

1.3.8 `channel.emit(trigger:string[, arguments:arguments])`

Alias of `channel.broadcast`

1.3.9 `channel.executeTrigger(triggerDetails:object[, force:boolean])`

Executes pointed trigger. **triggerDetails** contains **name:string** and **args:array**

- public
- type: function
- throws error if triggerDetails is not an instance of Object

1.3.10 `channel.on(trigger:string, callback:function)`

Attaches callback to trigger event.

- public
- type: function

1.3.11 `channel.once(trigger:string, callback:function)`

Attaches callback to trigger event only for one time.

- public
- type: function

1.3.12 `channel.off(trigger:string)`

Detaches callback from trigger event (destroys trigger).

- public
- type: function

1.3.13 `channel._triggers`

Contains triggers of this channel.

- private
- type: object