

# **Exstrata Control Api V1.0**

The Exstrata Control Api enables the development and use of custom control software. Anyone can download the Api and use it to send commands to the Exstrata.

## **Open Api**

The current version of the api use is fully open and does not require registration. Anyone can download it and immediately use it to send commands to the Exstrata. When it is detected that the api is being used in an abusive way the api may be blocked and replaced by a new version, after which registration will be required to obtain an access key. But until that point we're hoping for fair use and we will keep this api fully open.

## **jQuery**

This api requires jQuery to function properly. jQuery is not included in the api itself and needs to be downloaded and implemented separately. Even though the api should work well with older versions of jQuery, it is recommended to use the latest jQuery version.

## **Some definitions**

### **Live control**

To be able to send commands to the exstrata you need to have live control. Only one client can have live control at any given time. If you ask for live control while another client already has live control you will automatically be placed in the livecontrol queue. Of course it is possible that others are already in the queue before you. The queue policy is "first come, first serve". Live control duration is typically maxed out at 60 seconds.

### **Live queue**

The live queue is a queue of clients waiting for live control. Only one client can have live control at any given time.

## Pattern

A sequence of instructions for the Extrata. Where each instruction commands the Extrata to display a specific color for a range of zones at a given time.

The Extrata has 80 zones numbered from 1 to 80 where 1 is the bottom zone and 80 is the top zone.

The color is supplied in rgb format.

Example dataset for a pattern:

```
var pattern = {
    'frames': [
        {
            'ms': 0,
            'zones': {
                '2': '77,23,34',
                '22': '77,23,34'
            }
        },
        {
            'ms': 700,
            'zones': {
                '13': '12,64,124',
                '56': '46,23,88',
                '74': '212,212,99'
            }
        },
        {
            'ms': 1200,
            'zones': {
                '1': '0,0,0',
                '2': '0,0,0'
            }
        }
    ]
};
```

This pattern will command the Extrata to display colors in zone 2 and 22 from the start and add colors for zone 13, 56 and 74 after 700 milliseconds and end the pattern after 1200 milliseconds (500 milliseconds after the last color change). This dataset is not entirely valid. The Extrata has 80 zones. The ending instruction should contain an `rgb(0, 0, 0)` for every single zone, so 80 in total.

## **How to start**

*Anyone who wants to work with the Extrata api is expected to have a basic understanding of Javascript. If the instructions in this documentation are too complicated, please consult the extensive Javascript information and tutorials that can be found on the internet.*

Please follow these instructions to start using the api:

### **Step 1**

Download the newest version of the api from the website (<http://extrata.nl/pagina/9/how-can-i-collaborate.html>).

### **Step 2**

Put the extrata.js file anywhere where your application can reach it.

### **Step 3**

Include the api in your application by referencing it.

Example (html): <script type="text/javascript" src="extrata.js"></script>

*Do not forget to download and implement the latest jQuery version.*

### **Step 4**

Create an instance of the api in your code.

Example (Javascript): var extrata = new Extrata();

### **Step 5**

Test if the api works, make sure you can see what the Extrata does (webcam or at the scene) as you execute the api test command "runTestPattern" (check the api methods for more information).

Now you're ready to go and use the api. See below for further specific methods and instructions.

## Api methods

### **runTestPattern**

Call this method to test if your api works. You should only do this once to simply test if your configuration works well in combination with this api. Remove any calls to this method as soon as the test result is satisfactory. Do not call this method in combination with any other api calls , doing so can cause unexpected behavior. The test should report through window alerts.

Example use:

```
extrata.runTestPattern();
```

### **subscribeToLiveControl**

Call this method to request live control. Subscribe to the liveControlChange method to be notified of the state of your liveControl request. It is possible that you'll be placed in the live control queue. Subscribe to the liveControlQueuePositionChange event to be notified of any liveControlQueuePosition changes.

Example use:

```
extrata.subscribeToLiveControl();
```

### **unsubscribeFromLiveControl**

Call this method to leave live control (and the live control queue). By running this method, when live control is no longer desired, you ensure the server ends your live control session or frees your position in the queue so that subsequent client wait time is shortened.

Example use:

```
extrata.unsubscribeFromLiveControl();
```

## **playPattern**

Call this method, when in live control, to command the Extrata to play the supplied pattern immediatly. See the "pattern" description for more information about pattern data format.

Example use:

```
extrata.playPattern(pattern);
```

## **bind**

```
.bind(eventname, callback);
```

To subscribe to any of the events dispatched by the api, use the bind method.

Example use:

```
extrata.bind('liveControlChange', function(event){
    alert('Hello world!');
});
```

## States & Events

In a typical use case, the api is constantly interacting with the Extrata Control server. At any given time api states can change.

The api dispatches events on important state changes, subscribing (listening) to these events will enable you to react accordingly. Subscribing can be done by using the bind method.

### State: liveControl

Boolean(true, false)

This state is true when you have live control (are allowed to send commands).

Example use (read the current state):

```
extrata.liveControl();
```

### Event: liveControlChange

This event is dispatched when you gain or lose live control.

Callback example use:

```
extrata.bind('liveControlChange', function(event){  
    alert('liveControlChange: '+ event.liveControl);  
});
```

### State: liveControlQueuePosition

Integer

This state contains the number of clients to wait for before gaining live control.

Example use (read the current state):

```
extrata.liveControlQueuePosition();
```

### Event: liveControlQueuePositionChange

This event is dispatched each time your liveControlQueuePosition changes. This event will only be dispatched while you're in the live queue (see subscribeToLiveControl for more information).

Callback example use:

```
extrata.bind('liveControlQueuePositionChange', function(event){  
    alert('liveControlQueuePositionChange: '+ event.liveControlQueuePosition);  
});
```

**State: liveControlSecondsTillEnd**

`Integer`

This state contains the number of seconds before liveControl ends

Example use (read the current state):

```
extrata.liveControlSecondsTillEnd();
```

**Event: liveControlSecondsTillEndChange**

This event is dispatched each time your liveControlSecondsTillEnd changes. This event will only be dispatched while you're in live control.

Callback example use:

```
extrata.bind('liveControlSecondsTillEndChange', function(event){
    alert('liveControlSecondsTillEndChange: ' + event.liveControlSecondsTillEnd);
});
```

## Full functional code sample

```
var extrata = new Extrata();

extrata.bind('liveControlChange', function(event){
    if (event.liveControl) {
        var pattern = extrata.getRandomPattern();
        extrata.playPattern(pattern);
    }
});

extrata.bind('liveControlQueuePositionChange', function(event){
    alert('Your new position in the queue: ' + event.liveControlQueuePosition);
});

extrata.subscribeToLiveControl();
```

## Good luck!

Good luck playing with the Extrata and developing your own control application. If you have any suggestions or contributions to make to the api or this documentation, please let us know at [info@kenniscampus.nl](mailto:info@kenniscampus.nl)