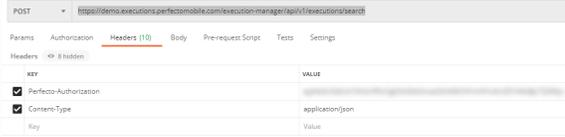# Execution Manager API

To see full Java code samples, go to the executions-api-sample in the Perfecto Git repository.

> **Notes:**
>
> - To be able to use this API, you must have Postman or a similar tool installed and add the following keys:
>   **Perfecto-Authorization:** Your Perfecto security token
>   **Content-Type:** `application/json`
>
>   
>
> - To use the API to view executions other than those running from the current user, the user must be assigned the Admin Execution role. For role assignments, contact Perfecto Support.

## Obtain a personal security token

The API requires the use of the Perfecto security token. Follow the procedure detailed in the Security Token documentation before proceeding.

> **Note:** The security token is associated with the specific Perfecto Lab and the personal registered user.

## Use the API

The API currently supports the following operations:

- Get Current Executions - Retrieve information regarding all currently running test executions
- Stop Executions - Stop all running executions for a user

### Get current executions

The following HTTP POST command is used to retrieve a list of the test executions in your lab:

| Method | API |
|---|---|
| POST | https://*<cloudname>*.executions.perfectomobile.com/execution-manager/api/v1/executions/search |

Retrieve running executions of the logged-in user according to received parameters. For execution manager admin user executions of all users will be returned.

#### Parameters

Parameters are added to the request in JSON format. This table lists the major parameters with sub-parameters that indicate different aspects.

| Major | Sub-flag | Description |
|---|---|---|
| layout | | Execution information fields to retrieve. In case no layout is provided, the full execution information will be returned. |
| | id | The execution ID |
| | name | The name of the test execution |
| | owner | The user name of the account running the execution |
| | platforms | The list of open devices associated with the execution |

| filter | | Data field values that limit the list of executions returned to those that match the filters. The execution fields that can be queried in fields and excludedFields fields are **id**, **name**, and **owner**. |
|--------|--------|--------|
| | fields | Any of the fields listed above with an appropriate value to match.  Only executions whose information matches the given filters will be listed. For example, to retrieve only those executions associated with the account of **user1**, supply the following:<br><br>`"filter" : {fields { "owner" : ["user1"] }}` |
| | excluded Fiel ds | Any of the fields listed above with an appropriate value to match. Only executions whose information does not match the given filters will be listed. For example, to retrieve executions whose script is not named **runExample**, supply the following:<br><br>`"filter" : {excludedFields { "name" : ["runExample"] }}` |
| | freeTe xtFilter | The value provided in this field is searched in the following fields:<br><br>• **Execution fields:** id, name, owner<br>• **Device fields:** id, device type, os name, os version |
| sort | | Defines the sort order of the result. Contains the fields by which the result should be sorted and their order. |
| | sortBy | The supported fields for sorting are: **id**, **name**, **owner**, **startTime** |
| | sortOr der | Indicates whether to sort in DESCEND or ASCEND order |

### Example

This example requests a list that includes the **execution id**, **script name**, and **devices** for all executions whose script name matches either **scriptName1** or  **scriptName2**, is run by **user1** but not run by **user2**, sorted by the **startTime** and then by the **script name**.

```
{
  "layout": ["id", "name", "devices"],
  "filter": {
    "fields": {
      "name": ["scriptName1", "scriptName2"],
      "owner": ["user1"]
    },
    "excludedFields": {
      "owner": ["user2"]
    },
"freeTextFilter": "someText"
  },
  "sort": [
    {
      "sortBy": "startTime",
      "sortOrder": "DESCEND"
    },
    {
      "sortBy": "name",
      "sortOrder": "ASCEND"
    }
  ],
}
```

The following is a successful response.

**Response code: 200**

```
[
    {
        "id": "tNWpYIDfNq",
        "name": "executionName1",
        "startTime": 1520789281226,
        "owner": "user1",
        "platforms": [
            {
                "id": "XQkEKGYKbU",
                "deviceType": "DESKTOP",
                "os": "Windows10",
                "osVersion": "2221",
                "screenResolution": "1600x1200",
                "location": "location1",
                "browserInfo": {
                    "browserType": "INTERNET_EXPLORER",
                    "browserVersion": "1.3"
                }
            }
        ]
    },
    {
        "id": "tNWwsfDfNq",
        "name": "executionName2",
        "startTime": 1520789245226,
        "owner": "user2",
        "platforms": [
            {
                "id": "ijzRvwSXed",
                "deviceType": "MOBILE",
                "os": "Windows10",
                "osVersion": "2221",
                "mobileInfo": {
                    "manufacturer": "manufacture",
                    "model": "model",
                    "description": "Description",
  "imei": "23456767",
 "imsi": "1234567",
 "phoneNumber": "1234567",
"distributor": "Distributor",
"firmware": "Firmware",
"operatorInfo": {
     "name": "operator name",
     "country": "operator country",
     "code": "operator code"
}
                },
                "screenResolution": "1440x2560",
                "location": "location"
            }
        ]
    }
]
```

The following shows an error response.

**Response code: 401 (Unauthorized) or 400 (Bad Request)**

```
[
    {
        "userMessage": "Error message 1"
    },
    {
        "userMessage": "Error message 2"
    }
]
```

## Stop executions

The following HTTP POST command is used to retrieve a list of the *test executions* in your lab.

| Method | API |
|--------|-----|
| POST | https://<*cloudname*>.executions.perfectomobile.com/execution-manager/api/v1/executions/stop |

You can stop running executions initiated by the logged-in user according to the received filter. For the execution manager admin, executions of all users will be stopped.

### Parameters

Parameters are added to the request in JSON format. The following table lists the parameter types that indicate the executions to stop.

| Parameter | Description |
|-----------|-------------|
| fields | Any of the fields listed above with an appropriate value to match. Only executions whose information matches the given filters will be listed. For example, to stop executions associated with the account of **user1**, supply: <br><br> ```fields { "owner" : ["user1"] }``` |
| excludedFields | Any of the fields listed above with an appropriate value to match. Only executions whose information does not match the given filters will be listed. For example, to retrieve executions whose script is not named **runExample**, supply: <br><br> ```excludedFields { "name" : ["runExample"] }``` |
| freeTextFilter | The value provided in this field is searched in the following fields: <br><br> • **Execution fields:** id, name, owner <br> • **Device fields:** id, device type, os name, os version |

### Example

This example requests to stop all executions whose script name matches either **scriptName1** or **scriptName2** and are run by **user1** but not run by **user2**.

```
{
    "fields": {
      "name": ["scriptName1", "scriptName2"],
      "owner": ["user1"]
    },
    "excludedFields": {
      "owner": ["user2"]
    },
    "freeTextFilter": "someText"
    }
```

The following is a successful response.

**Response code: 200**

```
{
    "stoppedExecutions": ["executionId1", "executionId2"],
    "unStoppedExecutions": ["executionId3]
    }
```

The following shows an error response.

**Response code: 401 (Unauthorized) or 400 (Bad Request)**

```
[
    {
        "userMessage": "Error message 1"
    },
    {
        "userMessage": "Error message 2"
    }
]
```