Main

Resource to verify the web service is running

- **URL**

  /api/main

- **Method:**

  `GET`

- **URL Params**

  No params needed

- **Data Params**

  No params needed

- **Success Response:**

  - **Code:** 200
    Content: "VES web service"

- **Error Response:**

  - **Code:** 404 Resource not found, verifz the wep application initial root

  - **Code:** 500 Web service not running

- **Sample Call:**

```
$.ajax({
      url: "http://<address>/api/main",
      success: function (data){
            console.log(data);
      }
});
```

Get service status

Verify the current status of the web service environment

- **URL**

  /api/cfg

- **Method:**

  `GET`

- **URL Params**

  No params needed

- **Data Params**

  No params needed

- **Success Response:**

  - **Code:** 200
    Content:
    ```
    {
    	available:<true if service can process sessions>,
    	configType:<direct/etcd>,
    	storageStatus:<status of storage>,
    	databaseStatus:<status of database or memory repository>
    }
    ```

- **Error Response:**

  - **Code:** 503 Service unavailable
    Content: same response of 200 – Success

  - **Code:** 500 Web Service not running

- **Sample Call:**

  ```
  $.ajax({
  	url: "http://<address>/api/cfg",
  	success: function (data){
  		console.log("%o", data);
  	}
  });
  ```

Configure service direct

Set the storage and database settings. After configured the service, the application should request the status.

- **URL**

  /api/cfg/direct

- **Method:**

  `PUT`

- **URL Params**

  No params needed

- **Data Params**

  Storage: [Mandatory] physical path to store and get files
  Database: [Optional] connection string to MongoDB database

- **Success Response:**

  - **Code:** 200

- **Error Response:**
    - **Code:** 500 Web Service not running
- **Sample Call:**
```
$.ajax({
      type: "POST",
      data: {
            storage: "/home/ves.storage",
            database: ""
      }
      url: "http://<address>/api/cfg/direct",
      success: function (data){
            console.log("configured");
      }
});
```
- **Note**
  If database parameter is blank, the sistem will store session information in memory

Configure service with etcd

Set the etcd address and key names to retrieve storage and database settings. After configured the service, the application should request the status.

- **URL**

  /api/cfg/etcd
- **Method:**

  `PUT`
- **URL Params**

  No params needed
- **Data Params**

  etcdEndPoint: [Mandatory] etcd server address
  StorageKey: [Mandatory] key containing the physical path to store and get files
  DatabaseKey: [Mandatory] key containing connection string to MongoDB database
- **Success Response:**
    - **Code:** 200
- **Error Response:**
    - **Code:** 500 Web Service not running
- **Sample Call:**
```
$.ajax({
      type: "POST",
      data: {
```

```
                storageKey: "storagekey",
                databaseKey: "databasekey",
                etcdEndPoint: <etcd address>
        }
        url: "http://<address>/api/cfg/etcd",
        success: function (data){
                console.log("configured");
        }
});
```

Get sessions

Retrieve the list of the current user sessions.

- **URL**

  /api/sessions

- **Method:**

  `GET`

- **URL Params**

  No params needed

- **Data Params**

  No params needed

- **Success Response:**

  - **Code:** 200
    Content:
    ```
    [
            {
            "id":"5a0fa962-e692-407d-89f7-c72f03167959",
            "status":2,
            "progress":0,
            "progressDescription":"",
            "progressLastModify":1457280875708,
            "resources": [
                            {
                                    "type":1,
                                    "file":"output5.avi"
                            },
                            {
                                    "type":3,
                                    "file":"sub.ass"
                            }
                    ]
            }
    ]
    ```

- **Error Response:**

        - **Code:** 500 Web Service not running

- **Sample Call:**
  ```
  $.ajax({
        type: "GET",
        url: "http://<address>/api/sessions",
        success: function (data){
              console.log("%o", data);
        }
  });
  ```

Create session

Create a new session.

- **URL**

  /api/sessions

- **Method:**

  `POST`

- **URL Params**

  No params needed

- **Data Params**

  No params needed

- **Success Response:**

    - **Code:** 200
      Content:
      ```
       {
      "id":"5a0fa962-e692-407d-89f7-c72f03167959",
      "status":2,
      "progress":0,
      "progressDescription":"",
      "progressLastModify":1457280875708,
      "resources": [
                  {
                        "type":1,
                        "file":"output5.avi"
                  },
                  {
                        "type":3,
                        "file":"sub.ass"
                  }
            ]
      }
      ```

- **Error Response:**

    - **Code:** 500 Web Service not running

- **Sample Call:**

  ```
  $.ajax({
        type: "POST",
        url: "http://<address>/api/sessions",
        success: function (data){
              console.log("%o", data);
        }
  });
  ```

Get specific session

Retrieve a specific user sessions. See "Session object" for further information.

- **URL**

  /api/sessions/{id}

- **Method:**

  `GET`

- **URL Params**

  id: the unique Id of the session to request

- **Data Params**

  No params needed

- **Success Response:**

  - **Code:** 200
    Content:

    ```
     {
    "id":"5a0fa962-e692-407d-89f7-c72f03167959",
    "status":2,
    "progress":0,
    "progressDescription":"",
    "progressLastModify":1457280875708,
    "resources": [
                {
                      "type":1,
                      "file":"output5.avi"
                },
                {
                      "type":3,
                      "file":"sub.ass"
                }
          ]
    }
    ```

- **Error Response:**

  - **Code:** 500 Web Service not running

  - **Code:** 404 Unknown session

- `Sample Call:`

  ```
  $.ajax({
        type: "GET",
        url: "http://<address>/api/sessions/5a0fa962-e692-407d-89f7-c72f03167959",
        success: function (data){
              console.log("%o", data);
        }
  });
  ```

Delete a session

Delete a specific user sessions and all associated file resources. It will stop ongoing burning process.

- **URL**

  /api/sessions/{id}

- **Method:**

  `DELETE`

- **URL Params**

  id: the unique Id of the session to request

- **Data Params**

  No params needed

- **Success Response:**

  - **Code:** 200

- **Error Response:**

  - **Code:** 500 Web Service not running

  - **Code:** 404 Unknown session

- `Sample Call:`

  ```
  $.ajax({
        type: "DELETE",
        url: "http://<address>/api/sessions/5a0fa962-e692-407d-89f7-c72f03167959",
        success: function (data){
              console.log("deleted");
        }
  });
  ```

Set video resize values

Specify the percentage of width and height of the processed video respect the original size.

- **URL**

  /api/sessions/{id}/resize

- **Method:**

  `POST`

- **URL Params**

  id: the unique Id of the session to request

- **Data Params**

  ```
  {
          widthPercentage: <0 – 100>,
          heightPercentage: <0 – 100>
  }
  ```

- **Success Response:**

  - **Code:** 200

- **Error Response:**

  - **Code:** 500 Web Service not running

  - **Code:** 404 Unknown session

- **Sample Call:**

  ```
  $.ajax({
          type: "POST",
          url: "http://<address>/api/sessions/12345678/resize",
          data: {
                  widthPercentage: 50,
                  heightPercentage: 50
          }
          success: function (data){
                  console.log("%o", data);
          }
  });
  ```

Upload the source video

Upload the original video to process.

- **URL**

  /api/sessions/{id}/video

- **Method:**

  `POST`

- **URL Params**

  id: the unique Id of the session to request

- **Data Params**

Multipart/form-data

- **Success Response:**
  - **Code:** 200

- **Error Response:**
  - **Code:** 500 Web Service not running
  - **Code:** 404 Unknown session

- `Sample Call:`

```
$('#form')
  .submit( function (e) {
      $.ajax({
            type: "POST",
            data = new FormData(this),
            processData: false,
            contentType: false,
            url: "http://<address>/api/sessions/12345678/video",
            success: function (data){
                  console.log("uploaded");
            }
      });
      e.preventDefault();
  });
```

Upload a subtitle file

Upload a subtitle file to apply on the final video.

- **URL**

  /api/sessions/{id}/subtitle

- **Method:**

  `POST`

- **URL Params**

  id: the unique Id of the session to request

- **Data Params**

  Multipart/form-data

- **Success Response:**
  - **Code:** 200

- **Error Response:**
  - **Code:** 500 Web Service not running

    - **Code:** 404 Unknown session

- **`Sample Call:`**

```
$('#form')
  .submit( function (e) {
      $.ajax({
            type: "POST",
            data = new FormData(this),
            processData: false,
            contentType: false,
            url: "http://<address>/api/sessions/12345678/subtitle",
            success: function (data){
                  console.log("uploaded");
            }
      });
      e.preventDefault();
  });
```

Process the video

Start processing the video. In order to execute succesfuly this request, the original video file and the subtitle file or the resize data must be provided in the current session

- **URL**

  /api/sessions/{id}/burn

- **Method:**

  `POST`

- **URL Params**

  id: the unique Id of the session to request

- **Data Params**

  No params needed

- **Success Response:**

    - **Code:** 200

- **Error Response:**

    - **Code:** 500 Web Service not running

    - **Code:** 404 Unknown session

- **`Sample Call:`**

```
$.ajax({
      type: "POST",
      url: "http://<address>/api/sessions/12345678/burn",
      success: function (data){
            console.log("processing");
      }
});
```

Interrupt the video processing

Interrupt the current video processing.

- **URL**

  /api/sessions/{id}/cancel

- **Method:**

  `POST`

- **URL Params**

  id: the unique Id of the session to request

- **Data Params**

  No params needed

- **Success Response:**

  - **Code:** 200

- **Error Response:**

  - **Code:** 500 Web Service not running

  - **Code:** 404 Unknown session

- **Sample Call:**

```
$.ajax({
      type: "POST",
      url: "http://<address>/api/sessions/12345678/cancel",
      success: function (data){
            console.log("stopped");
      }
});
```

Get the processed video

Retrieves the processed video.

- **URL**

  /api/sessions/{id}/result

- **Method:**

  `GET`

- **URL Params**

  id: the unique Id of the session to request

- **Data Params**

  No params needed

- **Success Response:**

  - **Code:** 200

- **Error Response:**

  - **Code:** 500 Web Service not running

  - **Code:** 404 Unknown session

- **Sample Call:**

```
$.ajax({
      type: "GET",
      url: "http://<address>/api/sessions/12345678/result",
      success: function (data){
            window.location = "http://<address>/api/sessions/12345678/result";
      }
});
```

Session object

A session is defined by the following parameters:

- id: unique identifier of the session
- status: the current status of the session. It can be:
  - 1: waiting for input (video, subtitles, resize values)
  - 2: ready to start process
  - 3: video in process
  - 4: interrupted by the user
  - 5: video processed and ready to be downloaded
  - 6: process failed, check processDescription for details
- progress: percentage of process completition
- progressDescription: current detailed process or error
- progressLastModify: timestamp of last session change
- resources: list of loaded resources of the current session
  - type: the resource type
    - 1: origin video file
    - 2: resize values
    - 3: subtitle file
  - file: the associated resource file
  - w: percentage of width respect the original video

- h: percentage of height respect the original video

Example:

```
{
	"id":"5a0fa962-e692-407d-89f7-c72f03167959",
	"status":2,
	"progress":0,
	"progressDescription":"",
	"progressLastModify":1457280875708,
	"resources":[
		{
			"type":1,
			"file":"output5.avi"
		},
		{
			"type":3,
			"file":"sub.ass"
		},
		{
			"type":2,
			"w":100,
			"h":100
		}
	]
}
```