Freeseer Configuration API Spec

Configuration Profiles

List

Endpoint: GET /profiles/
Request Body: None
Response:

```
{
        "profiles":[
                "profile_name",
                ...
                ],
}
```

Status Codes:
200 OK

View

Endpoint: GET /profiles/:profile
Request Body: None
Response:

```
{
        "configuration": {
                        // profile details
                },
        “schema”: {
                // Configuration Schema
        }
}
```

Status Codes:
200 OK
404 NOT FOUND - if profile doesn’t exist.

Create

Endpoint: POST /profiles/
Request Body:
{
"name": "name_of_profile"
}
Response:

If invalid:
{
"error_message": "...",
“error_code”: 400
}
Otherwise:

None

Status Codes:
201 Created
400 Bad Request

Delete

Endpoint: DELETE /profiles/:profile
Request Body:
None

Response:
None

Status Codes:
204 No Content - if successfully deleted.
404 Not Found - if profile doesn’t exist.

Modify

Endpoint: PATCH /profiles/:profile
Request Body:
{
	“property”: “value”,
	...
}
Response:
If invalid:
{
	"error_message": "...",
	“error_code”: 400
}
Otherwise:

None
Status Codes:
204 OK
400 Bad Request - for invalid args in request body.
404 NOT FOUND - if profile doesn’t exist.

General Configuration

View

Endpoint: GET /profiles/:profile/general
Response Body:

```
{
        “configuration”: {
                "default_language":  {
                                        "languages": [...],
                                        "current_language": "english"
                                    },
                "auto_hide": "true",
        },
        “schema”: {
                ...
        }
}
```

Status Codes:
200 OK
404 Not Found - if profile doesn"t exist

Modify

Endpoint: PATCH /profiles/:profile/general
Request Body:

```
{
        "default_language": "french",
        "auto_hide": false
}
```

Response Body:
None if OK

Otherwise:

```
{
        "error_message": "...",
        “error_code”: 400
}
```

Status Codes:
204 OK - If body passes validation.
400 Bad Request - If args in request body don"t pass validation.
404 Not Found - if profile doesn"t exist.

Input Options

default_language	String	The .qm language filename.
auto_hide	Boolean	Auto-hide to system tray on record.

Recording Configuration

View

Endpoint: GET /profiles/:profile/recording/
Response Body:

```
{
        “configuration”: {
                "record_to_file": "true",
                "videodir": "path",
                "record_to_file_plugin": "format",
                "record_to_stream": "false",
                "record_to_stream_plugin": "format",
                "enable_video_recording": "true",
                "videomixer": "mixer",
                "enable_audio_recording":" false",
                "audiomixer": "mixer"
                },
        “schema”: {
                // option-specific schemas
        }
}
```

Status Codes:
200 OK
404 Not Found - if profile doesn"t exist.

Modify

Endpoint: PATCH /profiles/:profile/recording/
Request Body:

```
{
        "record_to_file": true,
        "videodir": "path",
        "record_to_file_plugin": "unique plugin id", // output plugin
        "record_to_stream": false,
        "record_to_stream_plugin": "unique plugin id", // streaming plugin
        "enable_video_recording": true,
        "videomixer": "unique plugin id", // videomixer plugin
        "enable_audio_recording": false,
        "audiomixer": "unique plugin id" // audiomixer plugin
}
```

Reponse Body:
None if OK

Otherwise:

```
{
        "error_message": "...",
        “error_code”: 400
}
```

Status Codes:
204 OK
400 Bad Request - If args in request body don’t pass validation.
404 Not Found - if profile doesn’t exist.

Input Options

record_to_file	Boolean	Recordings are saved to file if true.
videodir	String	Folder to save recordings to.
record_to_file_plugin	String	Plugin used for recording file format.
record_to_stream	Boolean	Stream recordings if true.
record_to_stream_plugin	String	Plugin used for streaming format.
enable_video_recording	Boolean	Enable video if true.

videomixer	String	Plugin to use for mixing video inputs.
enable_audio_recording	Boolean	Enable audio if true.
audiomixer	String	Plugin to use for mixing audio inputs.

Plugin Configuration

The argument **:category** is used for the different plugin categories as shown in the Freeseer configuration tool. It will have 6 valid values (for now): output, importer, audioinput, and videoinput, audiomixer, videomixer.

The **:plugin** argument specifies the type plugin. Valid values are dependent on the category of plugin. For example: audioinput plugins include alsasrc, audiotestsrc, jackaudiosrc, etc.

The **:id** argument will specify the instance of the specific plugin, since a plugin.conf file can have multiple configurations for any given plugin and they are differentiated by a numeric id.

So a request to view the configuration for jackaudio-0 plugin might look like this:

GET /profiles/myprofile/recording/audioinput/jackaudiosrc/0

Additional notes: Will need to return unique identifiers to the plugin.conf entry for each instance of a plugin created, so that these identifiers can be referenced from within plugins that use embedded plugins. For example, if the pip-0 videomixer uses desktopsrc-0 and usbsrc-1 as its input plugins, then using view on this particular pip plugin to view its config might return something like:

GET /profiles/myprofile/recording/videomixer/pip/0

returns:

```
{
        “configuration”: {
                "main": "desktopsrc-0",
                "pip": "usbsrc-1",
        },
        “schema”: {
                ...
        }
}
```

Then, we could change one of the two inputs on the pip-0 plugin with a request like this:

PATCH /profiles/myprofile/recording/videomixer/pip/0

Request Body:

```
{
        "main": "firewire-0"
```

}

Plugin Categories

List

Lists available plugin types for the given :category. Ie, lists all videoinput plugins, etc.

Endpoint: GET /profiles/:profile/recording/:category
Response Body:

If valid:

```
{
        "plugins": [
                "plugin_name1", // note: this returns plugin classes, not instances.
                …
                ],
}
```

Otherwise:

```
{
        "error_message": "...",
        “error_code”: 400
}
```

Status Codes:
200 OK - If :category has available plugins
400 Bad Request - If :category is invalid
404 Not Found - if profile doesn’t exist.

Plugins

List

Lists available plugin instances for the given :category, :plugin.

Endpoint: GET /profiles/:profile/recording/:category/:plugin
Response Body:

If valid:

```
{
        "instances": [
                "plugin_instance1",
                …
                ],
}
```

Otherwise:

```
{
        "error_message": "...",
        “error_code”: 400
}
```

Status Codes:
200 OK - If :category has available plugins
400 Bad Request - If :category is invalid
404 Not Found - if profile doesn’t exist.

View

View the configuration for the specific :plugin instance with the given :id.

Endpoint: GET /profiles/:profile/recording/:category/:plugin/:id
Response Body:

```
{
        "configuration": {
                ...
        },
        “schema:” {
                ...
        }
}
```

Status Codes:

200 OK
400 Bad Request - If args are invalid.
404 Not Found - if profile doesn’t exist.

Create

Create a new instance of the given :plugin type. Returns this instance"s unique identifier, by which it"s configuration gets written to the plugin.conf file.

Endpoint: POST /profiles/:profile/recording/:category/:plugin
Request Body:
None

Response Body:
If valid:

```
{
        "id": "instance id"
}
```

Otherwise:

```
{
        "error_message": "...",
        “error_code”: 400
}
```

Status Codes:
201 Created - If args are valid.
400 Bad Request - If any args are invalid.
404 Not Found - if profile doesn’t exist.

Modify

Modifies option values for the configuration of the plugin given by :id.

Endpoint: PATCH /profiles/:profile/recording/:category/:plugin/:id
Request Body:

```
{
        // config options to modify
}
```

Response Body:
None if OK

Otherwise:

```
{
        "error_message": "...",
        “error_code”: 400
}
```

Status Codes:
204 OK - If body passes validation.
400 Bad Request - If args in request body don"t pass validation.
404 Not Found - if profile doesn’t exist.