Front end API

This page gathers all the API calls that can be used by the backend.

Backend -> Front end

> **⬛ Warning**
>
> The functions can only be called if they are available on the `web/build` directory, therefore if you make a change using `npm run serve` won't show it, you will need to rebuild the front end with `npm run buildWeb` or by using `npm run start`.

> **⬛ Note**
>
> These functions can only be called after eel is initiated with `eel.init()`.

Core API

Collection of all functions/API calls available to the backend. You can find them in the `bolinho_api/core.py` file.

The JavaScript file can be found in the api folder.

`def ping()`:

> Tries to ping the bolinho front-end, returns 1 if it worked
>
> **Python usage example**
> ```python
> from bolinho_api.core import core_api
>
> while True:
>     try:
>         if core_api.ping():
>             print("got a ping!")
>             break
>         pass
>     except:
>         eel.sleep(1)
> ```

def get_config_params() :

Tries to ping the bolinho front-end, returns 1 if it worked

Python usage example

```python
from bolinho_api.core import core_api

config = core_api.get_config_params()
current_save_version = config["configVersion"]
print(current_save_version)
```

This function is located at `src/web/src/App.js`

def go_to_experiment_page() :

Asks the front end to go to the experiment page.

Returns 1 if succeeded.

Python usage example

```python
from bolinho_api.core import core_api

change_pages = True
if change_pages:
    core_api.go_to_experiment_page()
```

def show_connect_prompt() :

Asks the front end to show the connection prompt.

The connection prompt is used to select the serial port.

Returns 1 if succeeded.

Python usage example

```python
from bolinho_api.core import core_api

config = core_api.get_config_params()
device_port = config["port"]

while not device_port:
    core_api.show_connect_prompt()
    device_port = config["port"]
```

UI API

Collection of all functions/API calls available to the backend for UI in general. You can find them in the `bolinho_api/ui.py` file.

The JavaScript file can be found in the api folder.

def success_alert(text) :

Uses React-Toastify to create an success alert.

Python usage example

from bolinho_api.ui import ui_api

ui_api.success_alert("Success!")

def error_alert(text) :

Uses React-Toastify to create an error alert.

Python usage example

from bolinho_api.ui import ui_api

UIapi.error_alert("Error!")

```python
def prompt_user(description, options, callback_func) :
```

Prompts the user with a 'description', and shows the 'options' to the user.

The result is passed to the callback_function

Python usage example

```python
from bolinho_api.ui import ui_api

def get_result(result):
    if result == "yes":
        print("The user chose yes")
    print("The user chose no")

UIapi.prompt_user(
    description="Do you want to pay 1000?",
    options=["yes", "no"],
    callback_func= get_result,
    )
```