

---

# **pybpodgui-api Documentation**

***Release 1.2.2***

**Scientific Software Platform (CF)**

**Nov 07, 2019**



<b>1</b>	<b>Examples</b>	<b>3</b>
1.1	Create a new project . . . . .	3
1.2	Access to data from a session . . . . .	4
<b>2</b>	<b>API Models</b>	<b>5</b>
2.1	<code>models.board</code> . . . . .	5
2.1.1	Overview . . . . .	5
2.1.2	Implementation . . . . .	5
2.2	<code>models.experiment</code> . . . . .	7
2.2.1	Overview . . . . .	7
2.2.2	Implementation . . . . .	7
2.3	<code>models.project</code> . . . . .	8
2.3.1	Overview . . . . .	8
2.3.2	Implementation . . . . .	8
2.4	<code>models.session</code> . . . . .	10
2.4.1	Overview . . . . .	10
2.4.2	Implementation . . . . .	10
2.5	<code>models.setup</code> . . . . .	12
2.5.1	<code>models.board.board_task</code> . . . . .	12
2.5.1.1	Overview . . . . .	12
2.5.1.2	Implementation . . . . .	12
2.5.2	<code>models.board.taskvariable</code> . . . . .	13
2.5.2.1	Overview . . . . .	13
2.5.2.2	Implementation . . . . .	13
2.5.3	Overview . . . . .	13
2.5.4	Implementation . . . . .	13
2.6	<code>models.subject</code> . . . . .	15
2.6.1	Overview . . . . .	15
2.6.2	Implementation . . . . .	15
2.7	<code>models.task</code> . . . . .	16
2.7.1	Overview . . . . .	16
2.7.2	Implementation . . . . .	16
<b>3</b>	<b>Project Info</b>	<b>19</b>
3.1	The SWP Team . . . . .	19
3.2	PyBpod project . . . . .	19
3.3	License . . . . .	19

3.4 Maintenance team . . . . . 20

**Index** **21**

This **API** makes available a set of functions and properties to allow the manipulation of a [PyBpod GUI](#) project using code.

Check the [Examples](#) section to learn how to use the API.



### 1.1 Create a new project

Create a new project programmatically. The resulting project can be opened with the [PyBpod GUI](#).

```
from pybpodgui_api.models.project import Project

# create the project
proj = Project()
proj.name = 'First project'

# add an experiment to the project
exp = proj.create_experiment()
exp.name = 'First experiment'

# add a setup to the project
stp = exp.create_setup()
stp.name = 'First setup'

# add a board to the project
board = proj.create_board()
board.name = 'First board'
board.serial_port = 'COM3'

# add a subject to the project
subj = proj.create_subject()
subj.name = 'First animal'

# add a new task\protocol to the project
task = proj.create_task()
task.name = 'First task'

exp.task = task # set the task\protocol to the experiment
stp.board = board # set the board to the setup
```

(continues on next page)

(continued from previous page)

```
stp += subj # add a subject to the setup

proj.save('my-project-folder')

task.code = 'print("My first protocol")'

proj.save(proj.path)
```

## 1.2 Access to data from a session

The next examples shows how to access the messages in a pybpod session.

```
from pybpodgui_api.models.project import Project

# create the project
proj = Project()
proj.load('my-project-folder')

exp = proj.experiments[0]
stp = exp.setups[0]

for session in stp.sessions:

    for msg in session.messages_history:

        if msg.check_type('INFO'):
            pass
            #Do something

        elif msg.check_type('stderr'):
            pass
            #Do something
```



## 2.1 models.board

### 2.1.1 Overview

Enumerates Bpod board supported operations.

### 2.1.2 Implementation

```
class pybpodgui_api.models.board.Board(project)

    end_run_task_handler()
    freogui()
        Function used to release the processing to update the GUI events
    load(path)
        Load board data from filesystem

    Variables
        • board_path(str) – Path of the board
        • data(dict) – data object that contains all board info

    log2board(data)
        Function used to update the board log

    pause_trial()
    remove()
    resume_trial()
    run_task_handler(flag=True)
```

**save()**

Save experiment data on filesystem.

**Variables** `parent_path` (*dict*) – Project path.

**Returns** Dictionary containing the board info to save. If None is returned, it means that there was a failure.

**Return type** *dict*

**start\_run\_task\_handler()**

**stop\_task()**

**stop\_thread()**

**stop\_trial()**

**INFO\_BOARD\_NAME** = 'BOARD-NAME'

**INFO\_BPODGUI\_VERSION** = 'BPOD-GUI-VERSION'

**INFO\_CREATOR\_NAME** = 'CREATOR-NAME'

**INFO\_EXPERIMENT\_NAME** = 'EXPERIMENT-NAME'

**INFO\_PROJECT\_NAME** = 'PROJECT-NAME'

**INFO\_SETUP\_NAME** = 'SETUP-NAME'

**INFO\_SUBJECT\_NAME** = 'SUBJECT-NAME'

**enabled\_behaviorports**

Get and set the experiment name

**Return type** *list*(Boolean)

**enabled\_bncports**

Get and set enabled bncports

**Return type** *list*(Boolean)

**enabled\_wiredports**

Get and set the enabled wired ports

**Return type** *list*(Boolean)

**name**

Get and set the board name

**Return type** *str*

**net\_port**

Get and set the experiment name

**Return type** *list*(Boolean)

**path**

Get and set the board path

**Return type** *str*

**project**

Get and set the board project

**Return type** *str*

**serial\_port**

Get and set the board serial port

**Return type** `str`

**status**

Get and set the board status

**Return type** `str`

## 2.2 models.experiment

### 2.2.1 Overview

Enumerates Bpod experiment supported operations.

### 2.2.2 Implementation

**class** `pybpodgui_api.models.experiment.Experiment` (*project*)

**create\_setup** ()

Create new instance of setup

**Return type** `Setup`

**load** (*path*)

Load experiment data from filesystem

**Variables**

- **experiment\_path** (*str*) – Path of the experiment
- **data** (*dict*) – data object that contains all experiment info

**Returns** Dictionary with loaded experiment info.

**remove** ()

Remove experiment

**save** ()

Save experiment data on filesystem.

**Variables** **parent\_path** (*dict*) – Project path.

**Returns** Dictionary containing the experiment info to save.

**Return type** `dict`

**name**

Get and set the experiment name

**Return type** `str`

**path**

Get and set the experiment files path

**Return type** `str`

**project**

Get and set the experiment project

**Return type** *Project*

**setups**

Get the experiment setups

**Return type** *list(Setup)*

**task**

Get and set the experiment task

**Return type** *Task*

## 2.3 models.project

### 2.3.1 Overview

Enumerates Bpod project supported operations.

### 2.3.2 Implementation

```
class pybpodgui_api.models.project.Project
```

**create\_board()**

Add an board to the project, and return it.

**Return type** *Board*

**create\_experiment()**

Add an experiment to the project, and return it.

**Return type** *Experiment*

**create\_subject()**

Add an subject to the project, and return it.

**Return type** *Subject*

**create\_task()**

Add an task to the project, and return it.

**Return type** *Task*

**create\_user()**

Add a user bject to the project, and return it.

**Return type** *User*

**find\_board(name)**

Find a board by the name

**Variables** *name* (*str*) – Name of the board to find.

**Return type** *Board*

**find\_session(uuid4)**

**find\_setup\_by\_id(uuid4)**

Find a setup by the id

**Variables** *uuid4* (*str*) – UUID4 of the setup to find.

**Return type** *Setup*

**find\_subject** (*name*)

Find a subject by the name

**Variables** *name* (*str*) – Name of the subject to find.

**Return type** *Subject*

**find\_subject\_by\_id** (*uuid4*)

Find a subject by the name

**Variables** *name* (*str*) – Name of the subject to find.

**Return type** *Subject*

**find\_task** (*name*)

Find a task by the name

**Variables** *name* (*str*) – Name of the task to find.

**Return type** *Task*

**find\_user** (*username*)

**import\_task** (*filepath*, *importdir=False*)

**is\_saved** ()

Verifies if project has changes by doing a recursive checksum on all entities

**Return type** *bool*

**load** (*project\_path*)

Load project from a folder.

**Variables** *project\_path* (*str*) – Full path of the project to load.

**remove\_non\_existing\_repositories** (*path*, *names*)

**save** (*project\_path*)

Save project data on file :param str project\_path: path to project :return: project data saved on settings file

**boards**

Get the list of boards in the project

**Return type** *list(Board)*

**experiments**

Get the list of experiments in the project

**Return type** *list(Experiment)*

**name**

Get and set the project name

**Return type** *str*

**path**

Get and set the project path

**Return type** *str*

**subjects**

Get the list of subjects in the project

**Return type** *list(Subject)*

**tasks**

Get the list of tasks in the project

**Return type** `list(Task)`

**users**

Get the list of users in the project

**Return type** `list(User)`

## 2.4 models.session

### 2.4.1 Overview

Enumerates Bpod session supported operations.

### 2.4.2 Implementation

```
class pybpodgui_api.models.session.Session(setup)
```

```
load(path)
```

**Parameters**

- **session\_path** –
- **data** –

**Returns**

```
load_contents(init_func=None, update_func=None, end_func=None)
```

Parses session history file, line by line and populates the history message on memory.

```
load_info()
```

```
remove()
```

Remove the session from the project

```
save()
```

**Parameters** **parent\_path** –

**Returns**

```
STATUS_READY = 0
```

```
STATUS_SESSION_RUNNING = 1
```

```
board_name
```

Get and set board name

**Return type** `str`

```
board_serial_port
```

Get and set board serial port

**Return type** `str`

```
ended
```

Get and set the end datetime of the session

**Return type** `datetime.datetime`

**filepath**

**is\_running**

**messages\_history**

Get and set the history of messages

**Return type** `list(BaseMessage)`

**name**

Get and set session name

**Return type** `str`

**path**

Get and set path name

**Return type** `str`

**project**

Get the session Project

**Return type** *Project*

**setup**

Get and set the setup

**Return type** *Setup*

**setup\_name**

Get and set setup name

**Return type** `str`

**started**

Get and set the start datetime of the session

**Return type** `datetime.datetime`

**status**

Get and set the session status

**Return type** `str`

**subjects**

Get and set session name

**Return type**

**task**

Get the session Task

**Return type** *Task*

**task\_name**

Get and set the task name

**Return type** `str`

**variables**

## 2.5 models.setup

### 2.5.1 models.board.board\_task

#### 2.5.1.1 Overview

Enumerates Bpod BoardTask supported operations.

#### 2.5.1.2 Implementation

**class** pybpodgui\_api.models.setup.board\_task.**BoardTask**(*setup*)

Represents the association between one board and one task

**Variables** *setup* (*Setup*) – Setup to which the BoardTask belongs to

**create\_variable** (*name=None, value=None, datatype='string'*)

Create a variable

**Return type** *TaskVariable*

**load** (*data*)

Load setup data from filesystem

**Variables**

- **setup\_path** (*str*) – Path of the setup
- **data** (*dict*) – data object that contains all setup info

**save** ()

Save board task data on filesystem.

**Variables** **setup\_path** (*str*) – Setup path.

**Returns** Dictionary containing the board task info to save.

**Return type** *dict*

**board**

Get and set the board

**Return type** *Board*

**task**

Get and set the Task

**Return type** *Task*

**update\_variables**

Get and set a flag that indicates if the variables should be updated at the end of the session

**Return type** *bool*

**variables**

Get and set list of variables

**Return type** *list(TaskVariable)*



## 2.5.2 models.board.taskvariable

### 2.5.2.1 Overview

Enumerates Bpod task variable supported operations.

### 2.5.2.2 Implementation

```
class pybpodgui_api.models.setup.task_variable.TaskVariable(board_task,
                                                         name=None,
                                                         value=None,
                                                         datatype='string')
```

**load**(*data*)  
Load variable data from filesystem

**Variables**

- **setup\_path**(*str*) – Path of the setup
- **data**(*dict*) – data object that contains all setup info

**save**()  
Save variable data on filesystem.

**Variables** **setup\_path**(*str*) – Setup path.

**Returns** Dictionary containing the variable info to save.

**Return type** *dict*

**datatype**  
Get and set variable data type. It can be from type ‘number’ or ‘string’

**Return type** *str*

**name**  
Get and set variable name

**Return type** *str*

**value**  
Get and set variable value

**Return type** *str*

## 2.5.3 Overview

Enumerates Bpod setup supported operations.

## 2.5.4 Implementation

```
class pybpodgui_api.models.setup.Setup(experiment)
```

**can\_run\_task**()

**clear\_subjects**()

**create\_board\_task()**

Create a new BoardTask object

**Return type** *BoardTask*

**create\_session()**

Create a new Session object

**Return type** *Session*

**load(path)**

Load setup data from filesystem

**Variables**

- **setup\_path** (*str*) – Path of the setup
- **data** (*dict*) – data object that contains all setup info

**pause\_trial()**

**remove()**

Remove the setup from the project

**resume\_trial()**

**save()**

Save setup data on filesystem.

**Variables** **parent\_path** (*str*) – Experiment path.

**Returns** Dictionary containing the setup info to save.

**Return type** *dict*

**stop\_task()**

**stop\_trial()**

**board**

Get and set setup board

**Return type** *Board*

**detached**

**experiment**

Get and set the experiment

**Return type** *Experiment*

**last\_session**

Get last created session

**Return type** *Session*

**name**

Get and set setup name

**Return type** *str*

**net\_port**

**path**

Get and set setup path

**Return type** *str*

**project**  
Get project  
**Return type** *Project*

**sessions**  
Get the list of sessions  
**Return type** *list(Session)*

**status**

**subjects**  
Get list of subjects  
**Return type** *list(Subject)*

**task**  
Get and set task  
**Return type** *Task*

**update\_variables**

## 2.6 models.subject

### 2.6.1 Overview

Enumerates Bpod Subject supported operations.

### 2.6.2 Implementation

```
class pybpodgui_api.models.subject.Subject (project)
```

**can\_run\_task()**

**get\_sessions()**  
Get all subject sessions  
**Return type** *list(Session)*

**load(path)**  
Load subject data from filesystem

**Variables**

- **subject\_path** (*str*) – Path of the subject
- **data** (*dict*) – data object that contains all subject info

**post\_load()**

**remove()**  
Remove the subject from the project

**run\_task()**

**save()**  
Save subject data on filesystem.

**Variables** **project\_path** (*str*) – Project path.

**Returns** Dictionary containing the setup info to save.

**Return type** `dict`

**toJSON()**

**name**

Get and set setup name

**Return type** `str`

**path**

Get and set the path

**Return type** `str`

**project**

Get and set project

**Return type** `str`

**setup**

Get and set project

**Return type** `str`

## 2.7 models.task

### 2.7.1 Overview

Enumerates Bpod Task supported operations.

### 2.7.2 Implementation

```
class pybpodgui_api.models.task.Task(project=None)
```

```
    create_execcmd()
```

```
    create_scriptcmd()
```

```
    load(path)
```

```
        Load setup data from filesystem
```

**Variables**

- **task\_path** (`str`) – Path of the task
- **data** (`dict`) – data object that contains all task info

```
    make_emptyfile()
```

```
        Creates the task folder if does not exists and an empty code file
```

```
    make_path()
```

```
        Creates the task folder
```

```
    remove()
```

```
        Remove the task from the project.
```

```
    save()
```

```
        Save setup data on filesystem.
```

**Variables**

- **project\_path** (*str*) – Project path.
- **data** (*dict*) – Dictionary where to save the data to.

**Returns** Dictionary containing the task info to save.

**Return type** *dict*

**commands**

Get commands

**Return type** *list*(TaskCommand)

**filepath**

Get and set task file path

**Return type** *str*

**name**

Get and set task name

**Return type** *str*

**path**

Get and set task path

**Return type** *str*

**project**

Get and set project

**Return type** *Project*

**trigger\_softcodes**

Get net port

**Return type** *int*



### 3.1 The SWP Team



Scientific Software Platform (Champalimaud Foundation)

The Scientific Software Platform (SWP) from the Champalimaud Foundation provides technical know-how in software engineering and high quality software support for the Neuroscience and Cancer research community at the Champalimaud Foundation.

We typically work on computer vision / tracking, behavioral experiments, image registration and database management.

### 3.2 PyBpod project

**PyBpod** is a python port of the [Bpod Matlab project](#).

All examples and Bpod's state machine and communication logic were based on the original version made available by [Josh Sanders \(Sanworks\)](#).

### 3.3 License

This is Open Source software with a MIT license.

## 3.4 Maintenance team

The current and past members of the **pybpodgui-api** team.

- @cajomferro Carlos Mão de Ferro
- @JBauto João Baúto
- @UmSenhorQualquer Ricardo Ribeiro
- @MicBoucinha Luís Teixeira



## B

Board (class in `pybpodgui_api.models.board`), 5  
board (`pybpodgui_api.models.setup.board_task.BoardTask` attribute), 12  
board (`pybpodgui_api.models.setup.Setup` attribute), 14  
board\_name (`pybpodgui_api.models.session.Session` attribute), 10  
board\_serial\_port (`pybpodgui_api.models.session.Session` attribute), 10  
boards (`pybpodgui_api.models.project.Project` attribute), 9  
BoardTask (class in `pybpodgui_api.models.setup.board_task`), 12  
create\_setup() (`pybpodgui_api.models.experiment.Experiment` method), 7  
create\_subject() (`pybpodgui_api.models.project.Project` method), 8  
create\_task() (`pybpodgui_api.models.project.Project` method), 8  
create\_user() (`pybpodgui_api.models.project.Project` method), 8  
create\_variable() (`pybpodgui_api.models.setup.board_task.BoardTask` method), 12

## C

can\_run\_task() (`pybpodgui_api.models.setup.Setup` method), 13  
can\_run\_task() (`pybpodgui_api.models.subject.Subject` method), 15  
clear\_subjects() (`pybpodgui_api.models.setup.Setup` method), 13  
commands (`pybpodgui_api.models.task.Task` attribute), 17  
create\_board() (`pybpodgui_api.models.project.Project` method), 8  
create\_board\_task() (`pybpodgui_api.models.setup.Setup` method), 13  
create\_execcmd() (`pybpodgui_api.models.task.Task` method), 16  
create\_experiment() (`pybpodgui_api.models.project.Project` method), 8  
create\_scriptcmd() (`pybpodgui_api.models.task.Task` method), 16  
create\_session() (`pybpodgui_api.models.setup.Setup`

## D

datatype (`pybpodgui_api.models.setup.task_variable.TaskVariable` attribute), 13  
detached (`pybpodgui_api.models.setup.Setup` attribute), 14

## E

enabled\_behaviorports (`pybpodgui_api.models.board.Board` attribute), 6  
enabled\_bncports (`pybpodgui_api.models.board.Board` attribute), 6  
enabled\_wiredports (`pybpodgui_api.models.board.Board` attribute), 6  
end\_run\_task\_handler() (`pybpodgui_api.models.board.Board` method), 5  
ended (`pybpodgui_api.models.session.Session` attribute), 10  
Experiment (class in `pybpodgui_api.models.experiment`), 7  
experiment (`pybpodgui_api.models.setup.Setup` attribute), 14  
experiments (`pybpodgui_api.models.project.Project` attribute), 9

**F**

`filepath` (*pybpodgui\_api.models.session.Session* attribute), 11

`filepath` (*pybpodgui\_api.models.task.Task* attribute), 17

`find_board()` (*pybpodgui\_api.models.project.Project* method), 8

`find_session()` (*pybpodgui\_api.models.project.Project* method), 8

`find_setup_by_id()` (*pybpodgui\_api.models.project.Project* method), 8

`find_subject()` (*pybpodgui\_api.models.project.Project* method), 9

`find_subject_by_id()` (*pybpodgui\_api.models.project.Project* method), 9

`find_task()` (*pybpodgui\_api.models.project.Project* method), 9

`find_user()` (*pybpodgui\_api.models.project.Project* method), 9

`freegui()` (*pybpodgui\_api.models.board.Board* method), 5

**G**

`get_sessions()` (*pybpodgui\_api.models.subject.Subject* method), 15

**I**

`import_task()` (*pybpodgui\_api.models.project.Project* method), 9

`INFO_BOARD_NAME` (*pybpodgui\_api.models.board.Board* attribute), 6

`INFO_BPODGUI_VERSION` (*pybpodgui\_api.models.board.Board* attribute), 6

`INFO_CREATOR_NAME` (*pybpodgui\_api.models.board.Board* attribute), 6

`INFO_EXPERIMENT_NAME` (*pybpodgui\_api.models.board.Board* attribute), 6

`INFO_PROJECT_NAME` (*pybpodgui\_api.models.board.Board* attribute), 6

`INFO_SETUP_NAME` (*pybpodgui\_api.models.board.Board* attribute), 6

`INFO_SUBJECT_NAME` (*pybpodgui\_api.models.board.Board* attribute), 6

`is_running` (*pybpodgui\_api.models.session.Session* attribute), 11

`is_saved()` (*pybpodgui\_api.models.project.Project* method), 9

**L**

`last_session` (*pybpodgui\_api.models.setup.Setup* attribute), 14

`load()` (*pybpodgui\_api.models.board.Board* method), 5

`load()` (*pybpodgui\_api.models.experiment.Experiment* method), 7

`load()` (*pybpodgui\_api.models.project.Project* method), 9

`load()` (*pybpodgui\_api.models.session.Session* method), 10

`load()` (*pybpodgui\_api.models.setup.board\_task.BoardTask* method), 12

`load()` (*pybpodgui\_api.models.setup.Setup* method), 14

`load()` (*pybpodgui\_api.models.setup.task\_variable.TaskVariable* method), 13

`load()` (*pybpodgui\_api.models.subject.Subject* method), 15

`load()` (*pybpodgui\_api.models.task.Task* method), 16

`load_contents()` (*pybpodgui\_api.models.session.Session* method), 10

`load_info()` (*pybpodgui\_api.models.session.Session* method), 10

`log2board()` (*pybpodgui\_api.models.board.Board* method), 5

**M**

`make_emptyfile()` (*pybpodgui\_api.models.task.Task* method), 16

`make_path()` (*pybpodgui\_api.models.task.Task* method), 16

`messages_history` (*pybpodgui\_api.models.session.Session* attribute), 11

**N**

`name` (*pybpodgui\_api.models.board.Board* attribute), 6

`name` (*pybpodgui\_api.models.experiment.Experiment* attribute), 7

`name` (*pybpodgui\_api.models.project.Project* attribute), 9

`name` (*pybpodgui\_api.models.session.Session* attribute), 11

`name` (*pybpodgui\_api.models.setup.Setup* attribute), 14

`name` (*pybpodgui\_api.models.setup.task\_variable.TaskVariable* attribute), 13

`name` (*pybpodgui\_api.models.subject.Subject* attribute), 16

`name` (*pybpodgui\_api.models.task.Task* attribute), 17

`net_port` (*pybpodgui\_api.models.board.Board* attribute), 6

`net_port` (*pybpodgui\_api.models.setup.Setup* attribute), 14

**P**

`path` (*pybpodgui\_api.models.board.Board* attribute), 6

path (*pybpodgui\_api.models.experiment.Experiment attribute*), 7

path (*pybpodgui\_api.models.project.Project attribute*), 9

path (*pybpodgui\_api.models.session.Session attribute*), 11

path (*pybpodgui\_api.models.setup.Setup attribute*), 14

path (*pybpodgui\_api.models.subject.Subject attribute*), 16

path (*pybpodgui\_api.models.task.Task attribute*), 17

pause\_trial() (*pybpodgui\_api.models.board.Board method*), 5

pause\_trial() (*pybpodgui\_api.models.setup.Setup method*), 14

post\_load() (*pybpodgui\_api.models.subject.Subject method*), 15

Project (*class in pybpodgui\_api.models.project*), 8

project (*pybpodgui\_api.models.board.Board attribute*), 6

project (*pybpodgui\_api.models.experiment.Experiment attribute*), 7

project (*pybpodgui\_api.models.session.Session attribute*), 11

project (*pybpodgui\_api.models.setup.Setup attribute*), 14

project (*pybpodgui\_api.models.subject.Subject attribute*), 16

project (*pybpodgui\_api.models.task.Task attribute*), 17

## R

remove() (*pybpodgui\_api.models.board.Board method*), 5

remove() (*pybpodgui\_api.models.experiment.Experiment method*), 7

remove() (*pybpodgui\_api.models.session.Session method*), 10

remove() (*pybpodgui\_api.models.setup.Setup method*), 14

remove() (*pybpodgui\_api.models.subject.Subject method*), 15

remove() (*pybpodgui\_api.models.task.Task method*), 16

remove\_non\_existing\_repositories() (*pybpodgui\_api.models.project.Project method*), 9

resume\_trial() (*pybpodgui\_api.models.board.Board method*), 5

resume\_trial() (*pybpodgui\_api.models.setup.Setup method*), 14

run\_task() (*pybpodgui\_api.models.subject.Subject method*), 15

run\_task\_handler() (*pybpodgui\_api.models.board.Board method*), 6

5

## S

save() (*pybpodgui\_api.models.board.Board method*), 5

save() (*pybpodgui\_api.models.experiment.Experiment method*), 7

save() (*pybpodgui\_api.models.project.Project method*), 9

save() (*pybpodgui\_api.models.session.Session method*), 10

save() (*pybpodgui\_api.models.setup.board\_task.BoardTask method*), 12

save() (*pybpodgui\_api.models.setup.Setup method*), 14

save() (*pybpodgui\_api.models.setup.task\_variable.TaskVariable method*), 13

save() (*pybpodgui\_api.models.subject.Subject method*), 15

save() (*pybpodgui\_api.models.task.Task method*), 16

serial\_port (*pybpodgui\_api.models.board.Board attribute*), 6

Session (*class in pybpodgui\_api.models.session*), 10

sessions (*pybpodgui\_api.models.setup.Setup attribute*), 15

Setup (*class in pybpodgui\_api.models.setup*), 13

setup (*pybpodgui\_api.models.session.Session attribute*), 11

setup (*pybpodgui\_api.models.subject.Subject attribute*), 16

setup\_name (*pybpodgui\_api.models.session.Session attribute*), 11

setups (*pybpodgui\_api.models.experiment.Experiment attribute*), 8

start\_run\_task\_handler() (*pybpodgui\_api.models.board.Board method*), 6

started (*pybpodgui\_api.models.session.Session attribute*), 11

status (*pybpodgui\_api.models.board.Board attribute*), 7

status (*pybpodgui\_api.models.session.Session attribute*), 11

status (*pybpodgui\_api.models.setup.Setup attribute*), 15

STATUS\_READY (*pybpodgui\_api.models.session.Session attribute*), 10

STATUS\_SESSION\_RUNNING (*pybpodgui\_api.models.session.Session attribute*), 10

stop\_task() (*pybpodgui\_api.models.board.Board method*), 6

stop\_task() (*pybpodgui\_api.models.setup.Setup method*), 14

stop\_thread() (*pybpodgui\_api.models.board.Board method*), 6

`stop_trial()` (*pybpodgui\_api.models.board.Board method*), 6  
`stop_trial()` (*pybpodgui\_api.models.setup.Setup method*), 14  
`Subject` (*class in pybpodgui\_api.models.subject*), 15  
`subjects` (*pybpodgui\_api.models.project.Project attribute*), 9  
`subjects` (*pybpodgui\_api.models.session.Session attribute*), 11  
`subjects` (*pybpodgui\_api.models.setup.Setup attribute*), 15

## T

`Task` (*class in pybpodgui\_api.models.task*), 16  
`task` (*pybpodgui\_api.models.experiment.Experiment attribute*), 8  
`task` (*pybpodgui\_api.models.session.Session attribute*), 11  
`task` (*pybpodgui\_api.models.setup.board\_task.BoardTask attribute*), 12  
`task` (*pybpodgui\_api.models.setup.Setup attribute*), 15  
`task_name` (*pybpodgui\_api.models.session.Session attribute*), 11  
`tasks` (*pybpodgui\_api.models.project.Project attribute*), 9  
`TaskVariable` (*class in pybpodgui\_api.models.setup.task\_variable*), 13  
`toJSON()` (*pybpodgui\_api.models.subject.Subject method*), 16  
`trigger_softcodes` (*pybpodgui\_api.models.task.Task attribute*), 17

## U

`update_variables` (*pybpodgui\_api.models.setup.board\_task.BoardTask attribute*), 12  
`update_variables` (*pybpodgui\_api.models.setup.Setup attribute*), 15  
`users` (*pybpodgui\_api.models.project.Project attribute*), 10

## V

`value` (*pybpodgui\_api.models.setup.task\_variable.TaskVariable attribute*), 13  
`variables` (*pybpodgui\_api.models.session.Session attribute*), 11  
`variables` (*pybpodgui\_api.models.setup.board\_task.BoardTask attribute*), 12