Package api

```
import "github.com/emilyhorsman/4zp6/backend/controller/api"
```

Overview
Index

Overview ▾

Index ▾

Variables
func Start(s *state.State) error
func controllerHandler(s *state.State, w http.ResponseWriter, r *http.Request)
func dataHandler(s *state.State, w http.ResponseWriter, r *http.Request)
func indexHandler(w http.ResponseWriter, r *http.Request)
func notFoundHandler(s *state.State, w http.ResponseWriter, r *http.Request)
func provisioningHandler(s *state.State, w http.ResponseWriter, r *http.Request)
func registerRoutes(s *state.State, r *mux.Router)
func websocketEventLoop(s *state.State)
func websocketHandler(s *state.State, w http.ResponseWriter, r *http.Request)

Package files

api.go controller.go data.go index.go provisioning.go websocket.go

Variables

```go
var (
    // websocket client tracking
    clients = make(map[*websocket.Conn]bool)

    // upgrade HTTP reqeusts to wbesockets
    upgrader = websocket.Upgrader{
        CheckOrigin: func(r *http.Request) bool {
            return true
        },
    }
)
```

func **Start**

```
func Start(s *state.State) error
```

Start will start the HTTP API server.

func **controllerHandler**

```
func controllerHandler(s *state.State, w http.ResponseWriter, r
*http.Request)
```

controllerHandler is "/api/microcontroller". It returns to the browser a list of registered microcontrollers and their connected peripherals.

func **dataHandler**

```
func dataHandler(s *state.State, w http.ResponseWriter, r *http.Request)
```

dataHandler is "/api/data". It is responsible for querying data. It accepts a mandatory "uuid" and "busAddr" parameter. It will query the provided UUID and bus address for data. An optional "start" and "stop" time can be provided for time filtering. The time parameters must be specified in RFC3339. By default "start" is the earliest data entry and "stop" is the time the request was performed.

func **indexHandler**

```
func indexHandler(w http.ResponseWriter, r *http.Request)
```

func **notFoundHandler**

```
func notFoundHandler(s *state.State, w http.ResponseWriter, r
*http.Request)
```

notFoundHandler returns the HTTP 404 response.

func **provisioningHandler**

```
func provisioningHandler(s *state.State, w http.ResponseWriter, r
*http.Request)
```

provisioningHandler is "/api/provisioning". It returns a list of provisioning configurations
and read definitions.

func **registerRoutes**

```
func registerRoutes(s *state.State, r *mux.Router)
```

registerRoutes registers all of the HTTP routes.

func **websocketEventLoop**

```
func websocketEventLoop(s *state.State)
```

websocketEventLoop is responsible for emitting data events to all connected websock
subscribers. If a client is no longer reachable, it will remove the client from the list.

func **websocketHandler**

```
func websocketHandler(s *state.State, w http.ResponseWriter, r
*http.Request)
```

websocketHandler is "/api/ws". It is responsible for upgrading HTTP connections to
websockets. It registers websocket clients into the client global variable map.