

API Reference

# Progetto CHeAria

API Version: 1.0

Documentazione del backend del progetto CHeAria

CONTACT

URL: <https://progettochearia.it>

# INDEX

<b>1. BOARD</b>	<b>3</b>
1.1 GET /board/time	3
1.2 GET /board/timems/{tz}	3
1.3 GET /board/timems	4
1.4 GET /board/time/hour	4
1.5 GET /board/time/min	4
1.6 GET /board/date/{tz}	5
1.7 GET /board/date	5
1.8 PUT /board/putdata/{dataid}	6
<b>2. RESOURCES</b>	<b>8</b>
2.1 GET /resources/graph/all	8
2.2 POST /resources/graph/query	8
2.3 GET /resources/datas	9
2.4 GET /resources/datas/last	10
2.5 GET /resources/datas_stream	11

# API

## 1. BOARD

REST API dedicate all'hardware del progetto, per esempio quella per caricare i dati su DB

### 1.1 GET /board/time

Gettime

#### REQUEST

No request parameters

#### RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - text/plain

string

---

### 1.2 GET /board/timems/{tz}

Gettimems

#### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*tz	string	

---

#### RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - text/plain

string

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*
      ANY OF
      prop0
        string
      prop1
        integer
    msg* string
    type* string
  }]
}
```

---

## 1.3 GET /board/timems

### Gettimems

#### REQUEST

##### QUERY PARAMETERS

---

NAME	TYPE	DESCRIPTION
tz	string	

---

#### RESPONSE

**STATUS CODE - 200:** Successful Response

**RESPONSE MODEL - text/plain**

string

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
      loc*
      ANY OF
      prop0
      string
      prop1
      integer
      msg* string
      type* string
    }]
}
```

---

## 1.4 GET /board/time/hour

### Gettime H

#### REQUEST

No request parameters

#### RESPONSE

**STATUS CODE - 200:** Successful Response

**RESPONSE MODEL - text/plain**

string

---

## 1.5 GET /board/time/min

### Gettime Min

#### REQUEST

No request parameters

## RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - text/plain

string

---

### 1.6 GET /board/date/{tz}

Getdate

## REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*tz	string	

---

## RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - text/plain

string

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*
      ANY OF
      prop0
        string
      prop1
        integer
    msg* string
    type* string
  }]
}
```

---

### 1.7 GET /board/date

Getdate

## REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
tz	string	

---

## RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - text/plain

string

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc*
        ANY OF
        prop0
          string
        prop1
          integer
      msg* string
      type* string
    }]
}

```

## 1.8 PUT /board/putdata/{dataid}

Putdata

### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*dataid	string	PATTERN: (itwork example CO altitude humidity ozone pressure temperature)

REQUEST BODY - application/json

```

{
  datavalue* number The value of data from sensor
  timestamptz string 2 to 3 chars
  The timzone to use to calculate the timestamp
  key* string 32 to 32 chars
  The key for authenticate request
}

```

### RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - text/plain

string

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc*

```

```
ANY OF
prop0
string
prop1
integer
msg* string
type* string
}]
}
```

---

## 2. RESOURCES

REST API dedicate alla gestione delle risorse, permettono di recuperare i dati dal DB e i grafici elaborati dal server

### 2.1 GET /resources/graph/all

Lista di tutti i grafici

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
type	string	

#### RESPONSE

STATUS CODE - 200: Return all graphs

RESPONSE MODEL - application/json

```
[string]
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*
      ANY OF
      prop0
        string
      prop1
        integer
    msg* string
    type* string
  }]
}
```

### 2.2 POST /resources/graph/query

Query Graph

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
dataid	array of string	
gte	string 10 to 19 chars	
lte	string 10 to 19 chars	



NAME	TYPE	DESCRIPTION
unique	string 10 to 10 chars PATTERN: (((((19 20)([2468][048] [13579][26] 0[48]) 2000)-02-29 ((19 20)[0-9]{2}-(0[4678] 1[02])-(0[1-9] [12][0-9] 30) (19 20)[0-9]{2}-(0[1359] 11)-(0[1-9] [12][0-9] 3[01]) (19 20)[0-9]{2}-02-(0[1-9] 1[0-9] 2[0-8])))	

## RESPONSE

**STATUS CODE - 200:** Successful Response

**RESPONSE MODEL - application/json**

undefined

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*
      ANY OF
      prop0
        string
      prop1
        integer
    msg* string
    type* string
  }]
}
```

## 2.3 GET /resources/datas

### Cerca i dati per tipo

Questa funzione permette di cercare per dato e per tipo i dati raccolti nel DB, [Trova i dati](https://web.progettochearia.it)

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*dataid	array of string PATTERN: (itwork example CO altitude humidity ozone pressure temperature)	
gte	string 10 to 19 chars PATTERN: ^((((19 20)([2468][048] [13579][26] 0[48]) 2000)-02-29 ((19 20)[0-9]{2}-(0[4678] 1[02])-(0[1-9] [12][0-9] 30) (19 20)[0-9]{2}-(0[1359] 11)-(0[1-9] [12][0-9] 3[01]) (19 20)[0-9]{2}-02-(0[1-9] 1[0-9] 2[0-8])))-([01][0-9] 2[0-3]):([012345][0-9]):([012345][0-9])) (((19 20)([2468][048] [13579][26] 0[48]) 2000)-02-29 ((19 20)[0-9]{2}-(0[4678] 1[02])-(0[1-9] [12][0-9] 30) (19 20)[0-9]{2}-(0[1359] 11)-(0[1-9] [12][0-9] 3[01]) (19 20)[0-9]{2}-02-(0[1-9] 1[0-9] 2[0-8]))))?)?\$	Data d'inizio, ex: 2022-05-15_10:24:00 or 2022-05-15

NAME	TYPE	DESCRIPTION
lte	string 10 to 19 chars PATTERN: ^((((19 20)([2468][048] [13579][26] 0[48]) 2000)-02-29 ((19 20)[0-9]{2}-(0[4678] 1[02])-(0[1-9] [12][0-9] 30) (19 20)[0-9]{2}-(0[1359] 11)-(0[1-9] [12][0-9] 3[01])) (19 20)[0-9]{2}-02-(0[1-9] 1[0-9] 2[0-8])))_([01][0-9] 2[0-3]):([012345][0-9]):([012345][0-9])) (((19 20)([2468][048] [13579][26] 0[48]) 2000)-02-29 ((19 20)[0-9]{2}-(0[4678] 1[02])-(0[1-9] [12][0-9] 30) (19 20)[0-9]{2}-(0[1359] 11)-(0[1-9] [12][0-9] 3[01])) (19 20)[0-9]{2}-02-(0[1-9] 1[0-9] 2[0-8])))?}\$	Data di fine, ex: 2022-10-15_16:12:00 or 2022-10-15
day	string 10 to 10 chars PATTERN: ^((((19 20)([2468][048] [13579][26] 0[48]) 2000)-02-29 ((19 20)[0-9]{2}-(0[4678] 1[02])-(0[1-9] [12][0-9] 30) (19 20)[0-9]{2}-(0[1359] 11)-(0[1-9] [12][0-9] 3[01])) (19 20)[0-9]{2}-02-(0[1-9] 1[0-9] 2[0-8])))?}\$	Giorno singolo, ex: 2022-10-15
type	string PATTERN: (html json)	
sort	string PATTERN: (asc desc)	

## RESPONSE

**STATUS CODE - 200:** success response

**RESPONSE MODEL - application/json**

```
{
  dataid* [{
    Array of object:
    time* string
    value* number
    metadata* {
      id* string
    }
  }]
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*
    ANY OF
    prop0
    string
    prop1
    integer
    msg* string
    type* string
  }]
}
```

## 2.4 GET /resources/datas/last

Ritorna l'ultimo dato per tipo

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*dataid	array of string PATTERN: (itwork example CO altitude humidity ozone pressure temperature)	
type	string PATTERN: (html json)	

## RESPONSE

STATUS CODE - 200: success response

RESPONSE MODEL - application/json

```
{
  dataid* {
    time* string
    value* number
    metadata* {
      id* string
    }
  }
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*
    ANY OF
    prop0
    string
    prop1
    integer
    msg* string
    type* string
  }]
}
```

## 2.5 GET /resources/datas\_stream

### Stream dei dati in tempo reale

Questa funzione restituisce uno streaming dei dati che arrivano dai sensori

## REQUEST

No request parameters

## RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

undefined



