

EcoShower API (2.4.0)

API for the EcoShower Smart Shower System using AWS Serverless (Lambda, DynamoDB, IoT Core).

List User Devices

AUTHORIZATIONS: > *CognitoAuth*

Responses

✓ 200

List of user's devices

RESPONSE SCHEMA: application/json

— devices > Array of objects (Device)

GET

/devices

▼

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{  
  - "devices": [  
    + { ... }  
  ]  
}
```

Add New Device

AUTHORIZATIONS: > *CognitoAuth*

REQUEST BODY SCHEMA: application/json
required

```
name          string  
device_code  string = 12 characters  
required
```

Responses

✓ 210

Device successfully created

RESPONSE SCHEMA: application/json

```
device > object (Device)
```

POST /devices

Request samples

Payload

Content type
application/json

```
{  
  "name": "string",  
  "device_code": "stringstring"  
}
```

Copy

Response samples

210

Content type

application/json

Copy Expand all Collapse all

```
{  
  - "device": {  
      "device_id": "3bafab7b-4400-4bcf-8e6e-09f954699940",  
      "user_id": "string",  
      "name": "Master Bathroom",  
      "device_code": "A1B2C3D4E5F6",  
      "status": "ready",  
      "target_temp": 38.5,  
      "current_temp": 22  
    }  
}
```

Get Device Details

AUTHORIZATIONS: > *CognitoAuth*PATH PARAMETERS

device_id string
required

Responses

✓ 200

Device details

RESPONSE SCHEMA: application/json

device > object (Device)

GET /devices/{device_id}

▼

Response samples

200**Content type**

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
  - "device": {  
      "device_id": "3bafab7b-4400-4bcf-8e6e-09f954699940",  
      "user_id": "string",  
      "name": "Master Bathroom",  
      "device_code": "A1B2C3D4E5F6",  
      "status": "ready",  
      "target_temp": 38.5,  
      "current_temp": 22  
    }  
}
```

Update Device

AUTHORIZATIONS: >*CognitoAuth*PATH PARAMETERS

device_id required string

REQUEST BODY SCHEMA: application/json

name string

target_temp number [30 .. 45]

Responses

— 200

Device updated

PUT /devices/{device_id}

▼

Request samples

Payload

Content type

application/json

```
{  
  "name": "string",  
  "target_temp": 30  
}
```

Copy

Start Shower Session

AUTHORIZATIONS: >

CognitoAuth

PATH PARAMETERS

device_id string
required

REQUEST BODY SCHEMA: application/json

target_temp number
duration integer
Planned duration in minutes

Responses

✓ 210

Session created

RESPONSE SCHEMA: application/json

session > object (Session)

POST /devices/{device_id}/start

Request samples

Payload

Content type

application/json

Copy

```
  "target_temp": 0,  
  "duration": 0  
}
```

Response samples

210

Content type

application/json

Copy Expand all Collapse all

```
{  
- "session": {  
    "session_id": "1ffd059c-17ea-40a8-8aef-70fd0307db82",  
    "status": "active",  
    "start_time": "2019-08-24T14:15:22Z",  
    "water_saved": 0,  
    "money_saved": 0  
}  
}
```

Stop Shower Session

AUTHORIZATIONS: > *CognitoAuth*

PATH PARAMETERS

→ device_id string
required

REQUEST BODY SCHEMA: application/json

duration number
Actual duration in seconds

Responses

✓ 200

Session stopped and stats calculated

RESPONSE SCHEMA: application/json

water_saved number
money_saved number
duration_seconds number

POST /devices/{device_id}/stop



Request samples

Payload

Content type
application/json

```
{  
  "duration": 0  
}
```

Copy

Response samples

200

Content type
application/json

```
{  
  "water_saved": 0,  
  "money_saved": 0  
}
```

Copy

```
        "money_saved": 0,  
        "duration_seconds": 0  
    }
```

Send IoT Command

AUTHORIZATIONS: > *CognitoAuth*

PATH PARAMETERS

device_id string
required

REQUEST BODY SCHEMA: application/json

command string
required
Enum: "START_HEATING" "STOP_HEATING" "OPEN_VALVE"
"CLOSE_VALVE"

Responses

200

Command sent via MQTT

POST /devices/{device_id}/command

▼

Request samples

Payload

Content type

application/json

Copy

```
{  
    "command": "START_HEATING"  
}
```

Get User Settings

AUTHORIZATIONS: > *CognitoAuth*

Responses

✓ 200

User profile and settings

RESPONSE SCHEMA: application/json

— settings > object (User)

GET /settings

▼

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{  
  - "settings": {  
      "user_id": "string",  
      "name": "string",  
      "email": "string",  
      "role": "user",  
      + "notifications": { ... },  
      + "system": { ... }  
    }  
}
```

Update User Settings

AUTHORIZATIONS: >

*CognitoAuth*REQUEST BODY SCHEMA: `application/json`

```

  name          string
  notifications > object
  system >      object

```

Responses

— 200

Settings updated

PUT /settings ▼

Request samples

Payload

Content type
`application/json`

Copy Expand all Collapse all

```
{
  "name": "string",
  - "notifications": {
      "water_ready_alert": true
    },
  - "system": {
      "temperature_unit": "string",
      "language": "string"
    }
}
```

Get User Usage Stats

AUTHORIZATIONS: >

CognitoAuth

Responses

✓ 200

Monthly aggregation

RESPONSE SCHEMA: application/json

```
total_water_saved  number  
total_money_saved  number  
sessions_count    integer
```

GET /dashboard/summary

▼

Response samples

200

Content type

application/json

Copy

```
{  
  "total_water_saved": 0,  
  "total_money_saved": 0,  
  "sessions_count": 0  
}
```

Get System Wide Stats (Admin)

AUTHORIZATIONS: > *CognitoAuth*

Responses

✓ 200

Global system statistics

RESPONSE SCHEMA: application/json

total_users	integer
total_devices	integer
activity_data	Array of arrays

GET /admin/stats



Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{  
  "total_users": 0,  
  "total_devices": 0,  
  "activity_data": []  
}
```

List All Users (Admin)

AUTHORIZATIONS: > CognitoAuth

Responses

✓ 200

List of all users

RESPONSE SCHEMA: application/json

— users > Array of objects (User)

GET /admin/users

▼

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{  
  - "users": [  
    + { ... }  
  ]  
}
```