

# How To Use HABitatX

2023 年 12 月 4 日

SenoOh

## 1 はじめに

HABitatX は、openHAB[1] では煩雑になりがちな複数デバイスの一括管理を支援するツールである。本システムは、openHAB の一括管理操作を提供するインターフェースとして動作する。事前に openHAB が動作していることが条件である。本システムは単体で動作し、openHAB のデバイス設定を担う設定ファイルを一括で作成、変更、削除できる。

”HABitatX”は、”openHAB”、”habitat”をもとに作られた造語である。この名前は、openHAB を表す”HAB”と生息地を表す”habitat”，未来への展望を表す”X”を組み合わせたものである。

## 2 使い方

本節では、本システムの使い方の概要を説明する。本システムは openHAB のデバイス設定を担う設定ファイルを一括で作成、変更、削除できる。設定ファイルはスプレッドシートとテンプレートコードを用いて作成する。テンプレートコードは、openHAB の設定ファイルの構造を保ちながら一部を埋め込み式に置き換え、スプレッドシートから得られた情報を利用して埋め込み設定ファイルを作成する。今回はスプレッドシートに Excel、テンプレートコードに ERB を採用する。まず、テンプレートコードを作成する。ファイル名とコードを書いて保存をすることで作成できる。次にテンプレートコードに対応するスプレッドシートを作成する。その後、Things と Items を作成する。Things や Items に一意に識別できる名前をつけ、上記のスプレッドシートとテンプレートコードを選択して作成する。そのため、Things や Items より先にテンプレートコードを作成する必要がある。詳細設定については以降の章で説明する。

## 3 Template

### 3.1 Template の作成手法

本節では Template の作成手法について説明する。root で「Tempalte」を選択し、/tenplate で「Template を作成」を選択することで Template を作成するページに移動できる。ここでファイル名とテンプレートコードを記述して保存することで Template を作成できる。ファイル名は一意に識別できる名前をつける。テンプレートコードは openHAB の設定ファイルの構造を保ちながら一部を埋め込み式に置き換えて記述する。以降の章で openHAB の設定ファイルの記法について説明する。

### 3.1.1 設定ファイルで Things を設定する手法

本節では、設定ファイルで Things を設定する手法について説明する。ファイルの拡張子は `.things` である。ファイルの構文は以下である。

```
Thing <things_type_id>:<thing_id> "Label" @ "Location" [ <parameters> ]
```

Things の設定ファイルの構文について表 2 にまとめる。

表 1 Things の設定 (設定ファイル)

セグメント	説明	備考
things_type_id	Things のタイプ (mqtt:topic など) を指定	Things の uid
thing_id	A-Z, a-z, 0-9, _, - の範囲で自由に定義	
Label	Thing を識別しやすくするための名前を自由に設定	
Location	Thing が物理的に設置されている場所を指定	
parameters	Things の設定パラメータを定義	

また、Things では Bridge の設定も可能である。構文は以下である。

```
Bridge <bridge_type_id>:<bridge_id> "Label" @ "Location" [ <parameters> ]{  
  Thing <type_id> <thing_id> "Label" @ "Location" [ <parameters> ]  
}
```

Bridge の設定ファイルの構文について表 3 にまとめる。

表 2 Bridge の設定 (設定ファイル)

セグメント	説明	備考
bridge_type_id	Bridge のタイプ (mqtt:broker など) を指定	Bridge の uid
bridge_id	A-Z, a-z, 0-9, _, - の範囲で自由に定義	
Label	Binding を識別しやすくするための名前を自由に設定	
Location	Thing が物理的に設置されている場所を指定	
parameters	Bridge の設定パラメータを定義	

また、Things では Channel の設定も可能である。構文は以下である。

```
Bridge <bridge_type_id>:<bridge_id> "Label" @ "Location" [ <parameters> ]{  
  Thing <type_id> <thing_id> "Label" @ "Location" [ <parameters> ]{  
    Channels:  
    Type <channel_type>:<channel_id> "Label" [ <parameters> ]  
  }  
}
```

}

Channels の設定ファイルの構文について表 4 にまとめる。

表 3 Channels の設定 (設定ファイル)

セグメント	説明	備考
channel_type	Channels のタイプ (switch number など) を指定	
channel_id	A-Z, a-z, 0-9, _, - の範囲で自由に定義	
Label	Channels を識別しやすくするための名前を自由に設定	
parameters	Channels の設定パラメータを定義	stateTopic などを設定

### 3.1.2 設定ファイルで Item を設定する手法

本節では、設定ファイルで Item を設定する手法について説明する。ファイルの拡張子は.items である。ファイルの構文は以下である。

```
item_type item_name "labeltext" <iconname> (Group1,Group2,...)
["tag1","tag2",...] {bindingconfig}
```

Items の設定ファイルの構文について表 5 にまとめる。

表 4 Items の設定 (設定ファイル)

セグメント	説明
item_type	Item のタイプ (Switch, String など) を指定
item_name	A-Z, a-z, 0-9, _, - の範囲で自由に定義
labeltext	Item を識別しやすくするための名前を自由に設定
iconname	Item の Category を設定
(Group1,Group2,...)	Parent Group(s) を設定
["tag1","tag2",...]	Semantic Class と Semantic Property を設定
bindingconfig	Channel の uid を指定して Things と link

### 3.1.3 テンプレートコードの記述例

本節ではテンプレートコードの記述例を示す。今回はスイッチの Things と Items のテンプレートコードを記述する。Things のテンプレートコードの記述例は以下である。

```
Thing mqtt:topic:MQTTBroker:<%= data['thingID'] %> "<%= data['label'] %>"
(mqtt:broker:MQTTBroker) [
  host = "localhost"
```

```

] {
    Channels:
<% data['number'].times do |i| %>
    <% switch_number = i + 1 %>
    Type switch : switch<%= switch_number %> [
        stateTopic = "dt/switch<%= switch_number %>",
        commandTopic = "cmd/switch<%= switch_number %>
    ]
<% end %>
}

```

thingsID, label, number をスプレッドシートで設定することでスイッチの Things を作成できるテンプレートコードを作成した。

Items のテンプレートコードの記述例は以下である。

```

Switch <%= data['itemID'] %> "<%= data['label'] %>" (<%= data['parentID'] %>)
["Switch","Presence"] {channel= "<%= data['channel'] %>"}

```

itemID, label, parentID, channel をスプレッドシートで設定することでスイッチの Things を作成できるテンプレートコードを作成した。

その他の具体的なテンプレートコードの記法は openHAB のドキュメントを参照する。

## 3.2 Template の変更手法

本節では Template の変更手法について説明する。root で「Tempalte」を選択し、/tenplate で作成した Template の中から 1 つを選択することで Template を閲覧、編集、削除するページに移動できる。その後「編集」を選択することで Template を変更できるページに移動できる。ここでファイル名とコードを書き換えることで Template を変更できる。

## 3.3 Template の削除手法

本節では Template の削除手法について説明する。root で「Tempalte」を選択し、/tenplate で作成した Template の中から 1 つを選択することで Template を閲覧、編集、削除するページに移動できる。その後「削除」を選択することで Template を削除できる。

## 4 Things

### 4.1 Things の作成手法

本節では Things の作成手法について説明する。root で「Things」を選択し、/things で「Things を作成」を選択することで Things を作成するページに移動できる。ここで Things 名、Excel ファイル、テンプレートコードを選択することで Things を作成できる。テンプレートコードは Template で作成したコードのみ使用できるため、Things を作成する前に Template を作成する必要がある。

### 4.2 Things の変更手法

本節では Things の変更手法について説明する。root で「Things」を選択し、/things で作成した Things の中から 1 つを選択することで Things を閲覧、編集、削除するページに移動できる。その後「編集」を選択することで Things を変更できるページに移動できる。ここで Things 名、Excel ファイル、テンプレートコードを新たに選択することで Things を変更できる。

### 4.3 Things の削除手法

本節では Things の削除手法について説明する。root で「Things」を選択し、/things で作成した Things の中から 1 つを選択することで Things を閲覧、編集、削除するページに移動できる。その後「削除」を選択することで Things を削除できる。

## 5 Items

### 5.1 Items の作成手法

本節では Items の作成手法について説明する。root で「Items」を選択し、/items で「Items を作成」を選択することで Items を作成するページに移動できる。ここで Items 名、Excel ファイル、テンプレートコードを選択することで Items を作成できる。テンプレートコードは Template で作成したコードのみ使用できるため、Items を作成する前に Template を作成する必要がある。

### 5.2 Items の変更手法

本節では Items の変更手法について説明する。root で「Items」を選択し、/items で作成した Items の中から 1 つを選択することで Items を閲覧、編集、削除するページに移動できる。その後「編集」を選択することで Items を変更できるページに移動できる。ここで Items 名、Excel ファイル、テンプレートコードを新たに選択することで Items を変更できる。

## 5.3 Items の削除手法

本節では Items の削除手法について説明する。root で「Items」を選択し、/items で作成した Items の中から 1 つを選択することで Items を閲覧、編集、削除するページに移動できる。その後「削除」を選択することで Items を削除できる。

## 参考文献

- [1] openHAB: openHAB, openHAB Foundation e.V. (online), available from <https://www.openhab.org/> (accessed 2023-05-22).