



## 12. EJS の基本

### テンプレートエンジン

テンプレートエンジンは、MVCフレームワークの View を出力するためのライブラリです。テンプレートHTMLにプログラムで取得したデータを埋め込んで動的に表示します。また、表示部分 (View) のプログラムが分離できるため、効率よく管理・表示することができます。

### コンポーネントとレイアウト

繰り返し同じ処理を共通化したファイルを、コンポーネントといいます。また、HTMLの骨格部分 (html, bodyタグなど) の共通化したファイルはレイアウトといいます。

### EJS とは

EJS は Node.js のテンプレートエンジンの 1つです。EJS のファイル拡張子は `.ejs` です。

```
テンプレート名.ejs
```

### テンプレートファイルの配置場所

EJS ではテンプレートファイルの配置場所が決まっています。デフォルトで `views` フォルダに設定されているので、事前にプロジェクト直下に views フォルダを作成しておきます。

### ファイル構成

```
├─ .env
├─ node_modules
├─ package-lock.json
└─ package.json
```

```
├─ public/
│   └─ login.html
├─ routes.js
├─ server.js
└─ views
    └─ index.ejs
```

## テンプレートファイルの作成

`views/` に `index.ejs` を作成して、HTMLを記述しておきます。このファイルを、**Express** でレンダリングします。

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet">
  <title>Document</title>
</head>

<body>
  <div class="container">
    <h1 class="h1">EJS</h1>
  </div>

</body>

</html>
```

## EJS の設定

### EJSモジュールインストール

**EJS** モジュールをインストールします。

```
% npm i ejs
```

### テンプレートエンジンの設定

テンプレートエンジンを選択するには `app.set()` でパラメータを設定します。

```
app.set('view engine', テンプレートエンジン名)
```

テンプレートエンジンを `EJS` にするには、`ejs` を指定します。

```
const express = require('express')
const routes = require('./routes')

require('dotenv').config()
const host = process.env.HOST
const port = process.env.PORT

const app = express()

// テンプレートエンジンを EJS にする
app.set('view engine', 'ejs')

app.use(routes)

app.listen(port, host, () => {
  console.log(`Server listen: http://${host}:${port}`)
})
```

## レンダリング

テンプレートファイルを表示することをレンダリングといいます。Express では `render()` メソッドでファイルを指定して、レンダリングします。

```
res.render(ファイルパス)
```

`routes.js` のGETリクエスト `/` に対して、`index.ejs` をレンダリングするよう修正します。

```
router.get('/', (req, res) => {
  res.render('index.ejs')
})
```

## サーバ起動

サーバを起動して、テンプレートファイルが表示できるか確認します。

```
% node server
```

# EJS

## ソース

### server.js

```
const express = require('express')
const routes = require('./routes')

require('dotenv').config()
const host = process.env.HOST
const port = process.env.PORT

const app = express()

app.set('view engine', 'ejs')

app.use(routes)

app.listen(port, host, () => {
  console.log(`Server listen: http://${host}:${port}`)
})
```

### routes.js

```
const express = require('express')
const router = express.Router()
const item = require('./item')

router.get('/', (req, res) => {
  let data = {}
  data.title = 'トップページ';
  res.render('index.ejs', data);
})

router.get("/profile", (req, res) => {
  res.send("This is Profile page.");
})

router.post('/auth', (req, res) => {
  const login_name = req.body.login_name;
  const password = req.body.password;
```

```

    let message = 'ログインできませんでした';
    if (login_name == process.env.LOGIN_NAME && password == process.env.PASSWORD) {
        message = 'ログインしました';
    }
    res.send(message);
})

router.get("/item/", (req, res) => {
    const items = item.get();
    res.send(message);
})

router.get('/item/:id', (req, res) => {
    let message = '商品が見つかりませんでした';
    const id = req.params.id;
    const _item = item.find(id);
    if (_item) {
        message = _item.name + 'の価格:' + _item.price + '円';
    }
    res.send(message);
})

module.exports = router

```

## models/item.js

```

exports.values = [
    { id: 1, name: 'コーヒー', price: 150 },
    { id: 2, name: '紅茶', price: 180 },
    { id: 3, name: 'ほうじ茶', price: 100 },
]

exports.get = () => {
    return this.values
}

exports.find = (id) => {
    return this.values.find((value) => value.id == id)
}

```

## index.ejs

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet">
    <title>Document</title>
</head>

```

```
<body>
  <div class="container">
    <div>
      <a href="/profile">プロフィール</a>
      <a href="/item">商品一覧</a>
    </div>
    <h1 class="h1"><%= title %></h1>
  </div>

</body>

</html>
```

## 演習

### 問題1

---

`/profile` ルーティングで **EJS** でテンプレート表示してみましょう。テンプレートファイルは `profile/index.ejs` とします。

#### profile/index.ejs

```
...
<div class="container">
  <div>
    <a href="/">トップページ</a>
  </div>
  <h1 class="h1"><%= title %></h1>
  <div>
    <dl>
      <dt>名前</dt>
      <dd>横浜  太郎</dd>
      <dt>出身地</dt>
      <dd>横浜</dd>
      <dt>趣味</dt>
      <dd>
        旅行 グルメ スポーツ
      </dd>
    </dl>
  </div>
</div>
...
```

### 問題2

---

`/login` ルーティングを作成して、ログインページを表示してみましょう。`login.html` は、`login.ejs` テンプレートに変更します。

当サイトの教材をはじめとするコンテンツ（テキスト、画像等）の無断転載・無断使用を固く禁じます。これらのコンテンツについて権利者の許可なく複製、転用等する事は法律で禁止されています。尚、当ウェブサイトの内容をWeb、雑誌、書籍等へ転載、掲載する場合は「ロジコヤ」までご連絡ください。