



10. POSTリクエスト

URL エンコード

POSTデータの取得はURLエンコードが必要です。ver 4.16 以降 では `urlencoded()` メソッドで `extended: true` を指定します。

```
app.use(express.urlencoded({ extended: true }));
```

ver 4.16 以前は `body-parser` モジュールのインストールが必要ですが、ここでは説明を省略します。

POSTデータの取得

POST データを取得するには、リクエストオブジェクトの `body` プロパティで取得します。ログイン名、パスワードを取得して認証処理を実装します。

```
app.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);
})
```

.env の修正

`.env` に認証用の `LOGIN_NAME` と `PASSWORD` を設定します。

```
HOST=localhost
PORT=3000
```

LOGIN_NAME=test
PASSWORD=1234

認証確認

Express サーバを再起動して、認証確認をしてみましょう。

ログイン名	<input type="text" value="test"/>
パスワード	<input type="password" value="...."/>
<input type="button" value="Loigin"/>	

`.env` に設定したログイン名、パスワードでログインできました。

ログインしました

ソース

server.js

```
const express = require('express');

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

const app = express();

app.use(express.urlencoded({ extended: true }));
app.use(express.static(__dirname + '/public'));

//Webルート GETリクエスト
app.get("/", (req, res) => {
  console.log(req.body);
  console.log(req.url);
  console.log(req.query);

  res.send("Hello Express!");
});
```

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

app.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);
})

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

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。