

## 4-3-6.データの受け渡しーPOSTー

# データの受け渡しーPOSTー

## はじめに

前回はGET送信について学習しました。  
GET送信はURLが明示的でわかりやすい反面、  
データが誰にでも閲覧可能という注意点がありました。  
そんな注意点を考慮したデータの送信方法がPOST送信です。

## Step1: 概念を知る

### POST (メソッド)

HTTPのメソッドの一つで、以下のような特徴があります。

- サーバからの情報取得時に使用
- パラメーター等の値がリクエストURLには表示されず、メッセージボディに格納される
- GET送信よりも多くのデータ量を扱える
- バイナリーデータ（2進数）が送信可能

## Step2: 使い方を知る

内容に入る前に、前回の復習です。

```
https://oooo.jp/login.php?lang=ja&args=
プロトコル://ドメイン/表示ファイル名?パラメータ = データ & パラメータ = なし
※わかりやすくするためにスペースを入れています。
```

どのURLもだいたいこのような構成になっていることを学びました。  
今回も取り上げる部分は、`?`以降の**パラメータ**と**データ**の部分です。

GET送信の際は、`=`以降に直接値を入力しました。  
一方、POST送信はパスワードなど、明示的にすべきでないもののデータの送信を行う際に使用するため、  
特にパラメータに対して値を入力するということはありません。  
GET送信との違いがわかればここでは問題ありません。

### パラメータの取得

パラメータの取得はGET送信の時と同じです。  
こちらも前回の復習となります。

```
String value = request.getParameter("args");
```

今回はもう1つ学んでいきます。  
上記の方法では、1つの引数 (`"args"`) に対して1つの値を取得しますが、  
1つの引数に対して複数の値が送られてくることがあります。  
その際に使用するのが `getParameterValues` メソッドです。  
複数の値が返されるので、配列を宣言しています。

?

```
String values[] = request.getParameterValues("args");
```

【サンプル: チェックボックス】

html/jsp

```
// スポーツのチェックボックス
<form action="送信先" method="post">
  <input type="checkbox" name="sports" value="tennis" />テニス
  <input type="checkbox" name="sports" value="baseball" />野球
  <input type="checkbox" name="sports" value="football" />サッカー
  <input type="checkbox" name="sports" value="basketball" />バスケットボール
  <input type="checkbox" name="sports" value="table_tennis" />卓球
  <input type="submit" value="送信" />
</form>
```

java

```
// 各チェックボックスのvalueの値が取得可能です
String sports[] = request.getParameterValues("sports");
```

## 補足

POST送信は、基本的には人の目に見えないところにデータが格納され送信されますが、リクエストを構成している要素の一つである、メッセージボディに格納されています。

```
/* リクエストライン */
POST /hoge/ HTTP/1.1

/* リクエストヘッダ */
Host: localhost:oooo
Connection: keep-alive
Content-Length: 22
Cache-Control: max-age=0
Origin: http://localhost:oooo
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4) Chrome/54.0.oooo.98
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://localhost:oooo/home/
Accept-Encoding: gzip, deflate, br
Accept-Language: ja,en-US;q=0.8,en;q=0.6

/* メッセージボディ */
lang=ja&args=password
```

## 課題

提出課題はありませんので、一通り学習が終わったら次の章に進んで下さい。

最終更新日時: 2022年 09月 10日(土曜日) 06:19