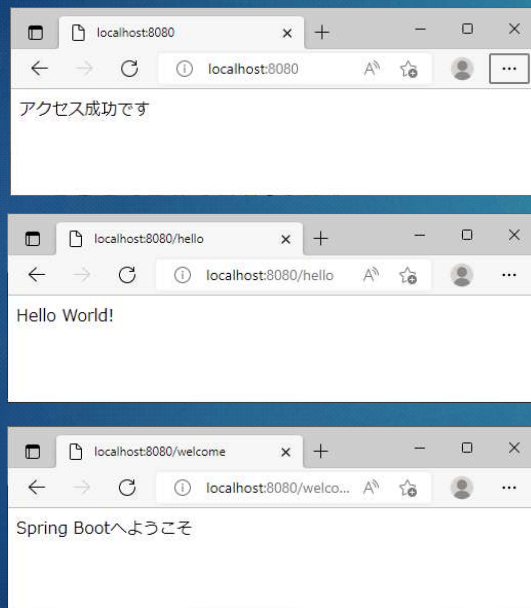


# Spring BootでHello World!

ブラウザからHTTP GETを利用してSpring Bootに対してリクエストを送り、単純に文字列を返すだけの簡単なAPIを作成します。

## ブラウザからHTTP GET



http://hostname:8080/

文字列 : アクセス成功です

http://hostname:8080/hello

文字列 : Hello World!

http://hostname:8080/welcome

文字列 : Spring Bootへようこそ

## Spring Boot

```
@RestController  
Hello Controller
```

# 作成するControllerクラス

## 作成するControllerクラス

```
6 @RestController
7 public class HelloController {
8
9     @GetMapping(value="/")
10    public String index() {
11        return "アクセス成功です";
12    }
13
14    @GetMapping(value="/hello")
15    public String hello() {
16        return "Hello World!";
17    }
18
19    @GetMapping(value="/welcome")
20    public String welcome() {
21        return "Spring Bootへようこそ";
22    }
23 }
```

## ポイント

### 【@RestController】

- ✓ 本アノテーションでSpring BootはControllerクラスとして認識
- ✓ メソッドの戻り値が、レスポンスの戻り値となる。
- ✓ Web API用のControllerで利用する

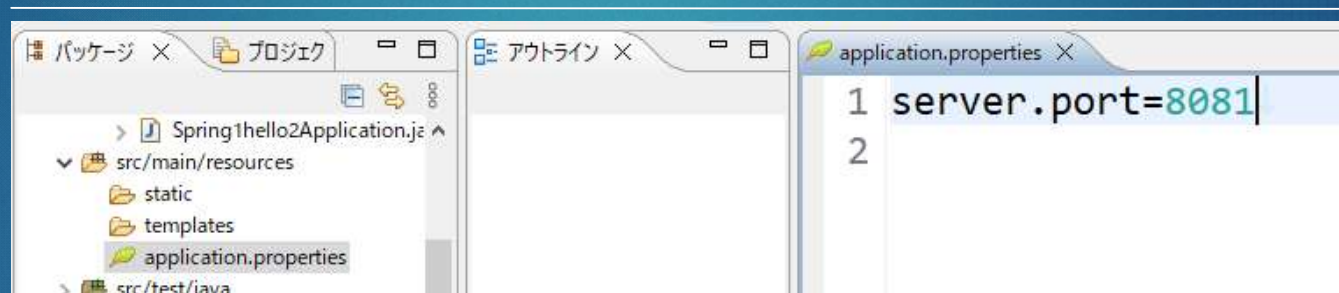
### 【@GetMapping】

- ✓ HTTP Getリクエスト用のアノテーション
- ✓ value 属性にURLのパスを指定する
- ✓ 紐づいているメソッドが実行される

# 実行するポートを変更する

Spring Boot はデフォルトで8080ポートで起動しますが、application.propertiesに設定する事で任意のポートに変更することができます。

## 設定例



**http://localhost:8081** で起動する



# 実機演習の流れ

## ■ Hello Worldを作成する

1. 以前のパートで作成した「spring1hello2」を使用します
2. controllerを管理する「com.udemy.spring1hello2.controller」パッケージを作成します
3. @RestController を付与したControllerクラスを作成します
4. 実行し、挙動を確認します

## ■ Spring Bootの実行ポートを変更する

5. Spring Bootを一旦終了します
6. application.properties に「server.port=8081」と追加します
7. 再度実行し、実行ポートが8081となることを確認します