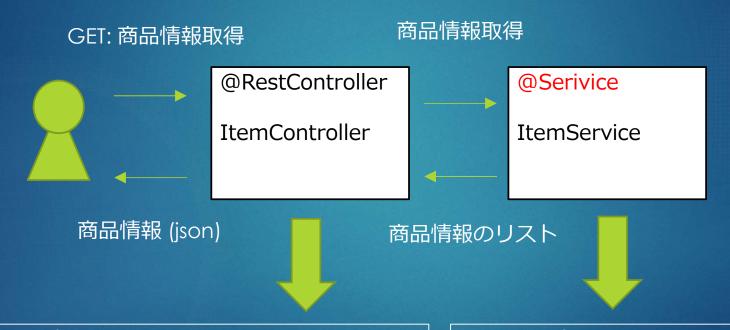
#### Rest API GETを実装する ~ @Serviceクラスの説明

HelloWorldの演習では単純に文字を返すだけで、Controllerクラスで完結しました。実際の業務処理は、 @Serviceアノテーションを付与した**サービスクラスに実装**します。



- ■Controllerクラス リクエストの入口で、必要な業務処理を呼び出す。 複雑な処理は基本的には書かない。
- ■Serviceクラス 業務処理の元となる箇所。ビジネスロジックはサー ビスクラスに実装する。

## Rest API GETを実装する ~@Autowiredを使う

Controllerクラスから、Serviceクラスを呼ぶ時は、<u>@Autowired アノテーション</u>を使用します。 SpringBootが<u>サービスクラスのインスタンスを自動で注入</u>してくれます。 @Service、@RestController等、SpringBoot管理のコンポーネントのインスタンスを取得できます。



フレームワークがインスタンスを自動で注入してくれている。そのままメソッドを呼ぶ事ができる!

## Rest API GETを実装する ~@PathVariable

特定の商品番号情報を取得するといったケースで、動的にURLに値を指定する仕様を実装します。値の受け取り方として、@PathVariableアノテーションを指定する事で受け取ることができます。 @GetMappingアノテーションのURLマッピングを指定する箇所には、動的に変わる部分は中括弧で表します。

GET: 商品情報10003番を取得 http://localhost:8080/items/10003

10003の商品情報 (json)

```
@RestController
ItemControllerクラス

...
@GetMapping("/items/{id}")
public Item getItem(@PathVariable("id") String id) {
   return itemService.getItem(id);
}
...
```

# 実機演習の流れ

【その①】 モデルを作成・コントローラクラスでListを作成し、そのままクライアント(Rest Client)に返却

Rest Client

全商品情報 (json)

controller パッケージ

@RestController ItemControllerクラス

@GetMapping("/items") public List getAllItemsメソッド model パッケージ

Itemクラス

Item

#### 【その②】 サービスを作成・サービスクラスでListを作成し、コントローラからはサービスを呼ぶだけに変更



<u>controller パッケージ</u> @RestController

@RestController ItemControllerクラス

@Autowired

private ItemService itemservice;

@GetMapping("/items") public List getAllItemsメソッド

@GetMapping("/items/{id}")
public Item getItem
(@PathVariable("id") String id)

@Service ItemServiceクラス

public List
getAllItemsメソッド

public Item getItem
メソッド

Create/Use

service パッケージ

model パッケージ

Itemクラス

Item

Create