**Rest ( Representational State Transfer )**

**==>** 전통적인 어플리케이션은
하나의 시스템에 모든 서비스를 다 구현하였습니다.
그러나 현재는 하나의 시스템에 모든 기능을 구현하기 보다는 기능별로 따로
따로 분리시켜서 시스템을 구현합니다.

예를 들어 고객관리, 배송, 주문처리를 위한 위한 기능을 예전에는 하나의 시스템에 모두
구현했다는 현재는 기능별로 따로 시스템 구현합니다.

이때에 배송시스템에서 고객의 정보가 필요할 때
고객관리 시스템에게 고객의 정보를 요청을 해야 합니다.

또, 주문시스템에서도 고객의 정보가 필요할 때 고객관리 시스템에게 고객의 정보를
요청해야 합니다. 각각의 시스템의 서로에게 데이터를 주고 받을 수 있도록 해야 합니다.

이때에 각각의 시스템에서는
요청에 따른 데이터를 동일한 형태로 응답합니다.

이와 같은 모든요청에 따른 동일한 데이터 형식으로 응답하는
시스템을 Restful 시스템 이라고 합니다.

스프링에서는 Rest를 위하여
컨트롤러에서 RestController 어노테이션을 이용합니다.
또, RestController에 요청을 위해서는 Ajax통신을 이용합니다.

그리고 Restful 시스템에서는 예전방식처럼 쿼리스트링에 의한 데이터 전송 보다는 URI에
의한 데이터 전송을 권장합니다.

예들들면 시스템에게 이름과 나이를 전송해야 하는 경우 전통적인 방법은 다음과 같습니다.
**http://localhost:8080/list.do?name=**이순신**&age=20**

그러나 rest에서는 다음과 같이 URI 자체에 데이터 전송방식을 사용합니다.
**http://localhost:8080/list/20/**이순신

URI 방식의 데이터를 받기 위해서는
컨트롤러에서는 @PathVariable 어노테이션을 이용합니다.

```
@RestController
public class HelloController {

	@RequestMapping("/list.do")
	public String hello(String name, int age) {
		return "no:"+no+",name:"+name;
	}
}
```

```
@RestController
public class HelloController {

	@RequestMapping("/list/{age}/{name}")
	public String hello(@PathVariable int age, @PathVariable String name) {
		return "no:"+no+",name:"+name;
	}
}
```