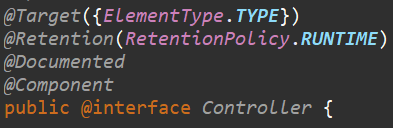
Annotation

* @Controller

返回一個View



變數僅有value,且此使用Component的value

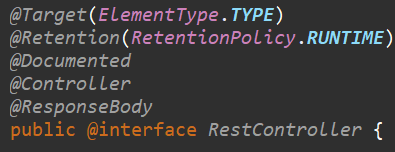
此為bean名稱,無設定時用class名稱(第一個字小寫)



* @RestController

= @Controller + @ResonseBody

直接響應http response (攔截器的postHandler無法用,需要實作ResponseBodyAdvice介面來修改響應內容)



變數與@Controller相同僅有value()

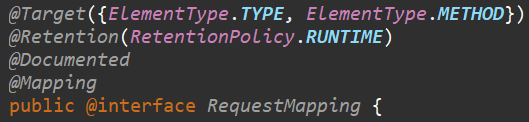
* @ControllerAdvice

待補

* @ResCOntrollerAdvice

待補

* @RequestMapping



變數：

名稱(標示符使用)，搭配AOP或URI組合時使用

String name() default "";

此表示URI路徑(可以用相對或絕對，且可以用通配符號或regex)

可以省略value或name,Ex:@RequestMapping("abc")

*@AliasFor*("path")

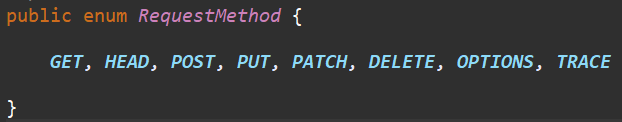
String[] value() default {};

*@AliasFor*("value")

String[] path() default {};

HTTP請求方法,使用RequestMethod.GET或RequestMethod.POST…

*RequestMethod*[] method() default {};



請求在包含這些參數且值匹配時才會被映射到這個方法

Ex: params= "query=myQuery" 或使用!=

String[] params() default {};

請求標頭匹配時才會被映射到這個方法

EX: headers = "content-type=application/json"

String[] headers() default {};

僅有Content-Type(Client發送的格式)匹配時才會被映射到這個方法

EX: consumes = "multipart/form-data"

String[] consumes() default {};

僅有Accept(Client希望Server回應的格式)匹配時才會被映射到這個方法

EX: produces = "application/json"

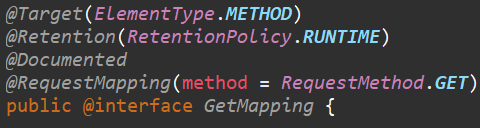
String[] produces() default {};

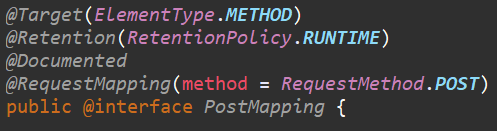
* @GetMapping、@PostMapping、@PutMapping、@DeleteMapping

等價@RequestMapping(method = RequestMethod.XXX)

XXX=GET、POST、PUT、DELETE

僅有這四個HTTP方法有這類annotation其他只能用@RequestMapping



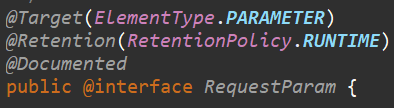


變數與@RequestMapping相同

* @RequestParam

PARAMETER僅能設定在方法傳入參數，不可設定在回傳參數

(java8設定回傳參數僅能用TYPE\_USE)



請求的參數名稱

EX: 前端:<input type='text' name='user'>

後端:

public void test(@RequestParam(name = "user") String userName

把前端的參數 user映射到userName (name帶的值傳到userName)

*@AliasFor*("name")

String value() default "";

*@AliasFor*("value")

String name() default "";

設定true (預設值),請求要有此參數,否則拋出錯誤

設定false,請求沒有此參數時,參數帶入null或帶入defaultValue()的值

Ex:

required = true 或沒有寫

/test?param=123 =>後端param對應的變數值為123帶入方法

沒有設定defaultValue()情況

/test 或 /test?param =>後端拋出錯誤

有設定defaultValue()

/test 或 /test?param =>帶入defaultValue()

required = false

/test?param=123 =>後端param對應的變數值=123

/test或/test?param =>後端param對應的變數值=null或defaultValue()

boolean required() default true;

有設定此，請求無帶此參數或帶此參數卻無值時會使用此

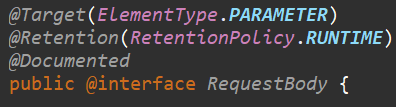
Ex:public void test(@RequestParam(defaultValue = "0")String value)

前端/test 或/test?value =>後端value帶入0

String defaultValue() default ValueConstants.***DEFAULT\_NONE***;

* @RequestBody

用在Http請求的Body參數綁定到傳入參數中



設定true (預設值為此)，請求body中缺少對應參數會拋出錯誤

設定false，請求body中缺少對應參數,該參數帶入null

Ex:

前端:body 無passwd參數

@RequestBody String passwd => 拋錯誤

前端:body帶passwd 或body帶passwd="1"

@RequestBody String passwd => 值帶入

boolean required() default true;

* @ResponseBody

方法返回http響應,非view

常用於RESTful API

Ex:

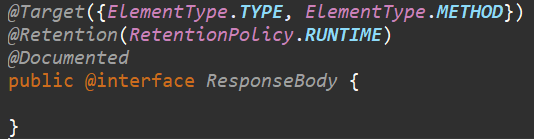
@GetMapping("/user")

@ResponseBody public User getUser() {

return new User("John", "Doe");

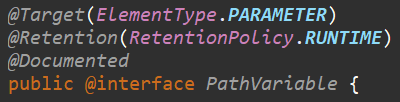
}

另外@RestControll = @Controller + @ResponseBody



* @PathVariable

URL路徑的變數綁定到傳入參數上



指定URL中的哪一個變數的名稱，綁定到傳入的參數

@PathVariable無宣告參數名稱時自動綁定value()和name()

可以使用Map<String,String> 一次綁定所有參數

Ex: @GetMapping("/users/{userId}")

public void users(@PathVariable(name = "userId") String id)或

public void users(@PathVariable("userId") String id)

Ex: @GetMapping("/users/{userId}/{userName}")

public void users(@PathVariable("userId") String id,

@PathVariable("userName") String userName)或

public void users(@PathVariable Mapping<String,String> user)

*@AliasFor("name")*

String value() default "";

*@AliasFor*("value")

String name() default "";

設定true (預設值) 沒有對應的參數在URL中會報錯誤

設定false會報null

EX:參考其他annotation required範例

boolean required() default true;

* @SessionAttributes

待補

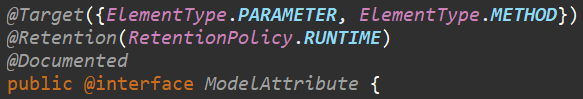
* @ModelAttribute

將Model屬性綁到參數中

如果是用在方法上,每次這個class的其他方法被呼叫時,會在執行方法前先執行帶有@ModelAttribute的方法,再執行對應的方法。

會觸發WebDataBinder(數據綁定)

(參考後面@InitBinder或其他word的WebDataBinder)



模型名稱 在view中可以直接 ${模型名稱.屬性}

*@AliasFor("name")*

String value() default "";

*@AliasFor*("value")

String name() default "";

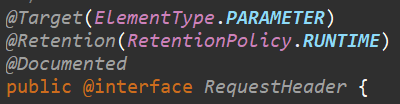
設定true (預設值) 會自動映射對應的屬性

設定false 不會自動映射對應屬性，需要自己一個一個set

boolean binding() default true;

* @RequestHeader

獲取HTTP標頭的特定值綁定到參數



取得的標頭名稱 (可使用Map<String,String>一次取得全部)

Ex:@RequestHeader("User-Agent") String userAgent, @RequestHeader("Accept-Language") String acceptLanguage 或

@RequestHeader Map<String, String> headers

*@AliasFor("name")*

String value() default "";

*@AliasFor*("value")

String name() default "";

設定true (預設值) 請求缺少對應的標頭，會拋出錯誤

設定false 參數會是null(如果有不確定有沒有的參數可設定此)

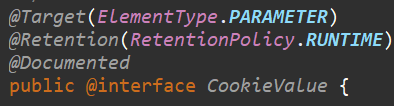
boolean required() default true;

當標頭無設定所需的參數時,給定一個預設值

String defaultValue() default ValueConstants.***DEFAULT\_NONE***;

* @CookieValue

HTTP請求中的cookies提取特定的值傳給參數



取得輸入的cookie 傳入參數

Ex:

Set-Cookie: userSession=abc123; Path=/user; Expires=Wed, 21 Oct 2025 07:28:00 GMT

設定@CookieValue("userSession") String userSession

*@AliasFor("name")*

String value() default "";

*@AliasFor*("value")

String name() default "";

設定true (預設值) 請求缺少對應的cookie，會拋出錯誤

設定false 參數會是null

boolean required() default true;

如果設定取得的cookie name存在就用原本的值,

如果不存在就使用此預設值

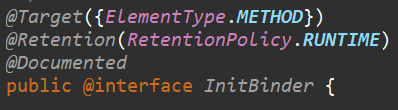
String defaultValue() default ValueConstants.***DEFAULT\_NONE***;

* @InitBinder

主要**初始化**WebDataBinder(為class)

僅初始未使用WebDataBinder,使用的為@ModelAttribute或@Valid(要相依validation實作Valid調整校驗)…

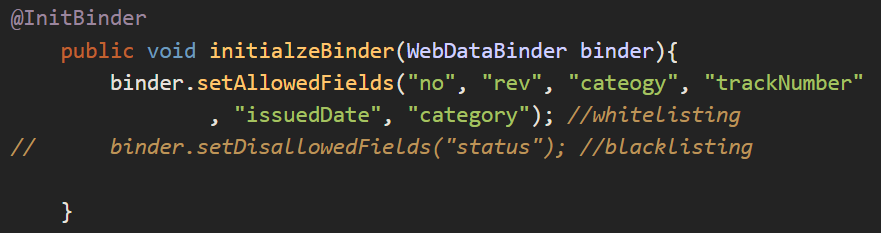
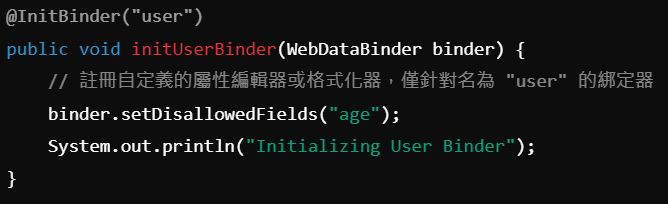
可把此設定在宣告@ControllerAdvice的class裡或設定在xml([spring xml設定連結](https://docs.spring.io/spring-framework/reference/web/webmvc/mvc-config/validation.html))中變全域配置



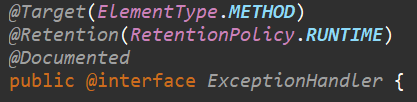
設定model的名稱 (預設為無)

String[] value() default {};

範例如下:

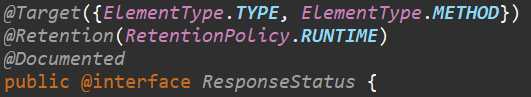
 

* @ExceptionHandler



Class<? extends Throwable>[] value() default {};

* @ResponseStatus



*@AliasFor*("code")

*HttpStatus* value() default *HttpStatus*.***INTERNAL\_SERVER\_ERROR***;

*@AliasFor*("value")

*HttpStatus* code() default *HttpStatus*.***INTERNAL\_SERVER\_ERROR***;

String reason() default "";