@ModelAttribute і Spring MVC

квітня

В Spring MVC є дуже корисна анотація - @ModelAttribute. Завдяки використанню цієї анотації можна істотно зменшити код контролера і полегшити його розуміння, а заодно і поліпшити його. :)

Анотацію @ModelAttribute можна використовувати як з методами, так і з параметрами методів.

**Додавання:**

**- одного атрибута в модель**

Якщо анотація @ModelAttribute використовується з методом, то значення, що повертається методом, автоматично додається в модель. Розглянемо простий приклад:

**@ModelAttribute**

**public Person addPerson(@PathVariable String name) {**

**return new Person(name);**

**}**

В даном випадку створюється об'єкт типу Person, що додається в модель. Ім'я цього об'єкта в моделі збігається з назвою класу об'єкта, який додається якщо явно не вказано ім'я атрибута моделі. Тобто, тепер в jsp-файлах можна звертатися до новоствореного об'єкта як:

**${person.name}**

Для полегшення розуміння коду представимо код класу Person:

package com.seostella.spring.domain;

**public class Person {**

**private String name;**

**public Person(String name) {**

**super();**

**this.name = name;**

**}**

**public String getGreeting(){**

**return "Hi " + name;**

**}**

**getter/setter**

**}**

*У прикладі використовувався найпростіший метод отримання об'єкта шляхом його створення. Так само можливо отримання об'єкта іншими способами. Наприклад, з бази або з файлової системи.*

Плюс в цьому підході в тому, що немає необхідності в контролері тримати логіку отримання об'єкту за параметрами запиту (або іншими властивостями). Кожен об'єкт ініціалізується в окремому методі, що полегшує читання коду і пошук можливих помилок.

Щоб явно вказати назву атрибуту, необхідно скористатися властивістю value анотації @ModelAttribute:

**@ModelAttribute(value = "personGagarin")**

**public Person addGagarinPerson() {**

**return new Person("Gagarin");**

**}**

В модель додасться атрибут personGagarin, до якого можна буде звернутися наступним способом:

**${personGagarin.name}**

**- декількох атрибутів в модель**

Щоб додати декілька атрибутів в модель, використовуючи один метод, скористайтеся анотацією @ModelAttribute з методом, який не повертає значення. Наприклад:

**@ModelAttribute**

**public void populateModel(Model model) {**

**model.addAttribute( new Sheldon() );**

**model.addAttribute( "personLeonard", new Person("Leonard") );**

**}**

Як і в попередньому випадку з додаванням одного атрибута в модель, якщо не вказувати ім'я атрибута, то він додається по імені класу. Тобто, об'єкт Sheldon додасться в модель під ім'ям sheldon. Також можна явно вказувати ім'я атрибута, як це зроблено з атрибутом personLeonard.

Код класу Sheldon:

**package com.seostella.spring.domain;**

**public class Sheldon extends Person{**

**public Sheldon() {**

**super("Sheldon");**

**}**

**}**

**Використання атрибутів моделі**

Якщо Вам необхідно скористатися атрибутом моделі (наприклад, змінити його), то використовуйте анотацію @ModelAttribute як аргумент методу. Наприклад:

**@RequestMapping(value = "/mod", method = RequestMethod.GET)**

**public ModelAndView showModifiedPerson(@ModelAttribute Person person) {**

**person.setName( person.getName() + " (modified using @ModelAttribute method attribute)" );**

**ModelAndView modelAndView = new ModelAndView("person");**

**return modelAndView;**

**}**

Вищенаведеним кодом ми змінимо значення атрибута person, додавши в кінець його імені фразу "(modified using @ModelAttribute method attribute)".

Якщо необхідно явно вказати назву атрибуту, змініть оголошення методу наступним чином, додавши до анотації @ModelAttribute параметр "person":

**public ModelAndView showModifiedPerson(@ModelAttribute("person") Person person) {**

**// ...**

**}**

Завантажити проект з прикладами зі статті можна за наступним посиланням - [Завантажити spring-modelattribute.zip](http://www.seostella.com/media/post/spring/modelattribute/spring-modelattribute.zip)