

W3D2: Quiz

CS544 Enterprise Architecture

1. Explain what the Front Controller Pattern is:

Spring Dispatcher. Receives all requests, and sends them to our mapped methods (in our controller beans).

It also does a bunch of processing if we want

2. Explain what a Flash Attribute is in the context of Spring MVC:

Attribute used for temporary storage with redirects. Stores it in the Session for just the duration of the redirect (next request).

3. Write the code for a Spring Data JPA interface called PizzaDao. This is not an open book or laptop quiz! Just do the best you can from memory.

```
public interface PizzaDao extends JpaRepository<Pizza, Long> {}
```

4. Write the code for a Spring MVC controller class called PizzaController that has the PizzaDao injected and has the following methods: getAll(), viewAdd(), add(), get(), update(), delete()
Hint: these are the same methods as that were on the Car and Book controller in the homework.

```
@Controller
```

```
public class PizzaController {  
    @Autowired  
    private PizzaDao pizzaDao;
```

```
    @GetMapping("/pizzas")  
    public String getAll(Model model) {  
        model.addAttribute("pizzas", pizzaDao.findAll());  
        return "pizzaList";  
    }
```

```
    @GetMapping("/pizzas/add")  
    public String viewAdd(Pizza pizza, Model model) {  
        model.addAttribute("pizza", new Pizza());  
        return "pizzaDetail"  
    }
```

```
    @PostMapping("/pizzas")  
    public String add(Pizza pizza) {  
        pizzaDao.save(pizza);  
        return "redirect:/pizzas";  
    }
```

```
@GetMapping("/pizzas/{id}")
public String get(@PathVariable long id, Model model) {
    Pizza pizza = pizzaDao.findById(id).get();
    model.addAttribute("pizza", pizza);
    return "pizzaDetail";
}

@PostMapping("/pizzas/{id}")
public String update(Pizza pizza) {
    pizzaDao.save(pizza);
    return "redirect:/pizzas";
}

@PostMapping("/pizzas/delete")
public String delete(@RequestParam long id) {
    pizzaDao.deleteById(id);
    return "redirect:/pizzas";
}
}
```