

Abstract classes and Spring MVC @ModelAttribute/@RequestParam

You need to provide your own Binding methods and then create the correct subtypes. Spring wouldn't know otherwise which of the subtypes should be instantiated for which element.

Example: Url address can be like this: /proje-basvuru/ilk-uzun-metrajli-film or /proje-basvuru/kisa-film

```
1  @ModelAttribute("project")
2  public Project getProject(final HttpServletRequest request){
3      return createProject(getProjeBasvuruUrl(request.getRequestURI()));
4  }
5
6  private Project createProject(String projectType) {
7      if (projectType.equals("ilk-uzun-metrajli-film"))
8          return new FirstFeatureFilm();
9      if (projectType.equals("kisa-film"))
10         return new ShortFilm();
11         return new Project();
12     }
13
14     private String getProjeBasvuruUrl(String url){
15         return RegularExpressionUtil
16             .createNewInstance(url)
17             .getSingleWord("proje-basvuru/([\\w-]{0,})",1);
18     }
19
20     @RequestMapping(value = {"/kullanici/home/proje-basvuru/{projectType}/save"}, method =
21         RequestMethod.POST)
22     public ModelAndView saveProject(@Valid @ModelAttribute("project") Project project,
23         BindingResult bindingResult, @PathVariable("projectType") String projectType) {
24         ModelAndView modelAndView = new ModelAndView("kullanici/home/proje-basvuru");
25         modelAndView.addObject("projectCategory", projectCategoryService.get("url",projectType));
26         initForm(modelAndView, project);
27         if (!bindingResult.hasErrors()) {
28             projectService.saveOrUpdate(project);
29         }
30         return modelAndView;
31     }
32 }
```

When you access the form page or post the form, Spring will run `getProject()` method and init subtype. After post request Project model attribute will point to related sub type.

RegularExpressionUtil Class

```
1  import java.util.ArrayList;
2  import java.util.List;
3  import java.util.regex.Matcher;
4  import java.util.regex.Pattern;
5
6  public class RegularExpressionUtil {
7      private String searchText;
8      public static RegularExpressionUtil instance;
9
10     private RegularExpressionUtil(String searchText) {
11         this.searchText= searchText;
12     }
13
14     private RegularExpressionUtil() {
15     }
16 }
```

```

17 public static RegularExpressionUtil createInstance() {
18     if(instance == null) {
19         instance = new RegularExpressionUtil();
20     }
21
22     return instance;
23 }
24
25 public static RegularExpressionUtil createNewInstance(String searchText) {
26     return new RegularExpressionUtil(searchText);
27 }
28
29 public boolean isMatch(String regex) {
30     Pattern pattern = Pattern.compile(regex);
31     this.searchText= this.searchText.replace("\\Q", "").replace("\\E", "");
32     Matcher matcher = pattern.matcher(this.searchText);
33     return matcher.find();
34 }
35
36 public boolean isMatch(String str, String regex) {
37     this.searchText= str;
38     return this.isMatch(regex);
39 }
40
41 public boolean isEscapeMatch(String regex) {
42     Pattern pattern = Pattern.compile(regex);
43     Matcher matcher = pattern.matcher(this.searchText);
44     return matcher.find();
45 }
46
47 public boolean equals(String regex) {
48     return this.isMatch("^(" + regex + ")$");
49 }
50
51 public List<String> getSubStrings(String regex, int groupId) {
52     ArrayList result = new ArrayList();
53     Pattern pattern = Pattern.compile(regex);
54     this.searchText= this.searchText.replace("\\Q", "").replace("\\E", "");
55     Matcher matcher = pattern.matcher(this.searchText);
56
57     while(matcher.find()) {
58         result.add(matcher.group(groupId));
59     }
60
61     return result;
62 }
63
64 public String getSingleWord(String regex, int groupId) {
65     List result = this.getSubStrings(regex, groupId);
66     return result != null && result.size() > 0?(String)result.get(0):null;
67 }
68
69 public void setSearchText(String searchText) {
70     this.searchText= searchText;
71 }
72 }

```