


30c54ef16c ▾

...

emissary / src / main / java / emissary / server / mvc / internal / CreatePlaceAction.java / <> Jump to ▾

 ctubbsii Apply changes from formatter plugin update History 2 contributors

68 lines (60 sloc) | 3.3 KB

```
1 package emissary.server.mvc.internal;
2
3 import static emissary.server.mvc.adapters.PlaceStarterAdapter.CP_CLASS_NAME;
4 import static emissary.server.mvc.adapters.PlaceStarterAdapter.CP_DIRECTORY;
5 import static emissary.server.mvc.adapters.PlaceStarterAdapter.CP_LOCATION;
6
7 import javax.ws.rs.Consumes;
8 import javax.ws.rs.FormParam;
9 import javax.ws.rs.POST;
10 import javax.ws.rs.Path;
11 import javax.ws.rs.Produces;
12 import javax.ws.rs.core.MediaType;
13 import javax.ws.rs.core.Response;
14
15 import emissary.admin.PlaceStarter;
16 import emissary.place.IServiceProviderPlace;
17 import org.apache.commons.lang.StringUtils;
18 import org.slf4j.Logger;
19 import org.slf4j.LoggerFactory;
20
21 @Path("")
22 // context is /emissary, set in EmissaryServer
23 public class CreatePlaceAction {
24     private final Logger logger = LoggerFactory.getLogger(this.getClass());
25     public static final String EMPTY_PARAM_MSG = "Required query parameters were blank or null, verify the parameters";
26     public static final String PLACE_STARTER_ERR = "Cannot create the requested place";
27
28     /*
29     * <!-- Create and register a place on a remote machine --> <Use-Case source="" action="/CreatePlace.action"> <Work
30     * type="Bean" target="emissary.comms.http.worker.LogWorker"/> <Work type="Bean"
31     * target="emissary.comms.http.worker.CreatePlaceWorker"/> <View status="0" view="/createplace.jsp"/> <View status="-1"
32     * view="/error.jsp"/> </Use-Case>
33     */
34
35     @POST
36     @Path("/CreatePlace.action")
37     @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
38     @Produces(MediaType.TEXT_PLAIN)
39     public Response createPlace(@FormParam(CP_CLASS_NAME) String className, @FormParam(CP_LOCATION) String location,
40                                @FormParam(CP_DIRECTORY) String directory) {
41         if (StringUtils.isBlank(className) || StringUtils.isBlank(location) || StringUtils.isBlank(directory)) {
42             logger.error("{}: className - {}, location - {}, directory - {}", EMPTY_PARAM_MSG, className, location, directory);
43             return Response.serverError().entity(EMPTY_PARAM_MSG).build();
44         }
45
46         try {
47             final IServiceProviderPlace thePlace = PlaceStarter.createPlace(location, className, directory);
48             // previous CreatePlaceWorker returned like this
49             // Put new place on the request going forward
50             // req.setAttribute(PlaceStarterAdapter.PLACE_OBJECT, thePlace);
51             if (thePlace != null) {
52                 return Response.ok().entity(thePlace.toString()).build();
53             } else {
54                 logger.error(PLACE_STARTER_ERR + "class - {}, location - {}, directory - {}", className, location, directory);
55                 return Response.serverError()
56                     .entity(PLACE_STARTER_ERR + "class - " + className + ", location - " + location + ", directory - " + directory).build();
57             }
58         } catch (Exception e) {
59             // TODO Needed because we don't do any validation of the location string in the PlaceStarter, see TODOs
60             // there
61             logger.error("There was an error trying to create the place. class - {}, location - {}, directory - {}", className, location, directory,
62                 e);
63             return Response.serverError()
64                 .entity(PLACE_STARTER_ERR + "class - " + className + ", location - " + location + ", directory - " + directory).build();
65         }
66     }
67 }
68 }
```