Spring MVC Tutorial – Contact List Application

Step 11: Implementing the Search REST Controller Endpoints

The Learning House

427 S 4th Street #300

Louisville KY 40202

Step 11: Implementing the Search REST Controller Endpoints

Overview

In this step, we will create a new REST endpoint in the Search controller. This endpoint will translate the values in the Search form into a Map which will be passed on to the DAO searchContacts method. The List of Contacts returned from the searchContacts method will then be sent back to the caller.

Designing Our URL

We will design the endpoint for Search following the same approach used in Step 08 for the CRUD endpoints:

Searching for Contacts:

Verb: POST

URL: search/contacts

RequestBody: JSON object containing the values entered into the Search form

ResponseBody: Array of JSON Contact objects

Controller Implementation

Add the following method to your SearchController (see code comments for details):

```
    // This method will be invoked by Spring MVC when it sees a POST request for
    // ContactListMVC/search/contacts.  It translates the entered search terms
    // from the given Map<String, String> to a Map<SearchTerm, String>, passes
    // the search criteria to the DAO, and returns the search results to the
    // caller.
    @RequestMapping(value="search/contacts", method=RequestMethod.POST)
    @ResponseBody
    public List<Contact> searchContacts(@RequestBody Map<String, String> searchMap) {
        // Create the map of search criteria to send to the DAO
        Map<SearchTerm, String> criteriaMap = new HashMap<>();

        // Determine which search terms have values, translate the String
        // keys into SearchTerm enums, and set the corresponding values
        // appropriately.
        String currentTerm = searchMap.get("firstName");
        if (!currentTerm.isEmpty()) {
            criteriaMap.put(SearchTerm.FIRST_NAME, currentTerm);
        }
        currentTerm = searchMap.get("lastName");
        if (!currentTerm.isEmpty()) {
            criteriaMap.put(SearchTerm.LAST_NAME, currentTerm);
        }
        currentTerm = searchMap.get("company");
        if (!currentTerm.isEmpty()) {
            criteriaMap.put(SearchTerm.COMPANY, currentTerm);
        }
        currentTerm = searchMap.get("email");
        if (!currentTerm.isEmpty()) {
            criteriaMap.put(SearchTerm.EMAIL, currentTerm);
        }
        currentTerm = searchMap.get("phone");
        if (!currentTerm.isEmpty()) {
            criteriaMap.put(SearchTerm.PHONE, currentTerm);
        }

        return dao.searchContacts(criteriaMap);
    }
}
```

Wrap-up

In this step, we created a REST endpoint that takes the entered search terms, passes them to the DAO, and returns the search results to the caller. In the next step, we'll wire the search form to our new endpoint.