

Online Mobile

From Servoy 7.3 onwards a Servoy developer can do a remote search for a specific foundset being in find mode and has findstates applied.

This allows to work with live data on the server, like foundsets with a million records.

In This Chapter

- [Concept](#)
- [Usage Client Side](#)
- [Needs Server Side](#)

Concept

The Servoy find/search concept is extended to start with a local find and utilize a server side search.

Usage Client Side

In the mobile solution you set a foundset into find mode and then add some criteria then instead of calling `search()` on the foundset, call `remoteSearch` on the mobile plugin:

```
if (foundset.find())
{
    foundset.anumbervalue = ">10";
    plugins.mobile.remoteSearch(foundset,successCallback,errorCallback);
}
```

The `successCallback` and `errorCallback` params are functions that are called when this `remoteSearch` is done because this call is an a-synchronize call to the server.

On success the retrieved foundset as argument is not the same foundset started the `find()` on, but a new one similar to the one from `databaseManager.getFoundSet()`.

This new foundset should be loaded into the UI by the success function as shown below. Note: returning a new foundset is intentionally to prevent remote search result to touch the UI.

The success and error callbacks have the following signature:

```
/**
 * @param {JSFoundSet} the foundset that was given to remoteSearch
 */
function successCallback(foundset)
{
    if (foundset.getSize() >0) {
        // something found on the server show it in the current form
        controller.showRecords(foundset);
    }
}

/**
 * @param {Number} statusCode The http status code
 * @param {String} message The message of the error
 * @param {JSFoundSet} foundset The foundset that was passed to remoteSearch
 */
function errorCallback(statusCode, message, foundset)
{
    //do what ever is needed, like showing alert
}
```

Needs Server Side

The `plugins.mobile.remoteSearch()` will call the service solutions 'offline_data' form 'ws_create()' method with the following signature:

```
/**
 * @param data The data that the client send over in the body
 * @param version The version number of this request
 * @param method What kind of create must be done
 * @param authenticateResult Object that is created in the ws_authenticate method.
 */
function ws_create(data,version,method,authenticateResult)
{
    // Test if it is a remoteSearch call
    if (method == "search") {
        // Create the FoundSet from the data that is send over from the mobile client
        var fs = plugins.mobileservice.createRemoteSearchFoundSet(data);
        // This FoundSet is in find mode, more stuff could be added to it if you want to filter even
more. Then call search()
        fs.search();
        // Create the OfflineDataDescription that will be returned for this remoteSearch
        var retval = plugins.mobileservice.createOfflineDataDescription('data_');
        if (fs.getDataSource() == "db:/server/test") {
            // if this was a foundset on the test table, traverse only the relation of
"test_to_test2".
            var traverse = new Array();
            traverse[0] = "test_to_test2";
            retval.addFoundSet(fs,traverse);
        }
        else {
            // for all other datasources just call addFoundSet which will traverse over all
relations found in the mobile shared module when encountered in the records of this foundset.
            retval.addFoundSet(fs);
        }
        return retval;
    }
}
```

This method is called by the `remoteSearch` call from the client and the `OfflineDataDescription` object that is returned will be merged into the current synced set the mobile client already has.

For all the pk's that are returned the service solution will be ask for to get the latest data for the Record.

If that record is changed on the client, the record data that the client had will be kept, to not delete changes the client made to it. If that is not the case then the local storage is updated with the new record data.

The foundset passed to remote search will go out of find mode and the remote filtered foundset will be given to the `successCallback` function.

All other global/shared foundsets will have all the new records and the existing records loaded, related foundset will be exactly the pk set that the server did return for it. Except new records on the client side, those will be inserted back in.

This does result in that the set of data in the mobile client does grow with every `remoteSearch()` call that gets new records, so calling `clearData()`, `loadData()` or doing a full `sync()` at certain times is recommended.