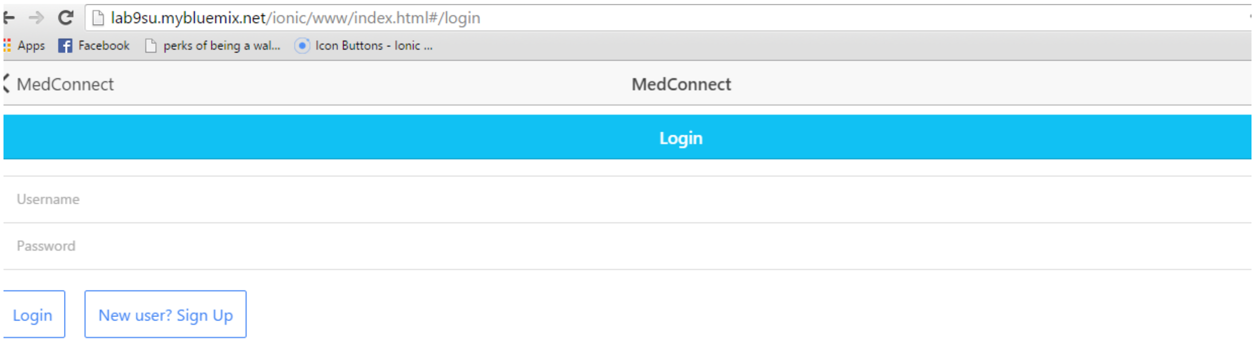
Lab 9

Suramya Chukka

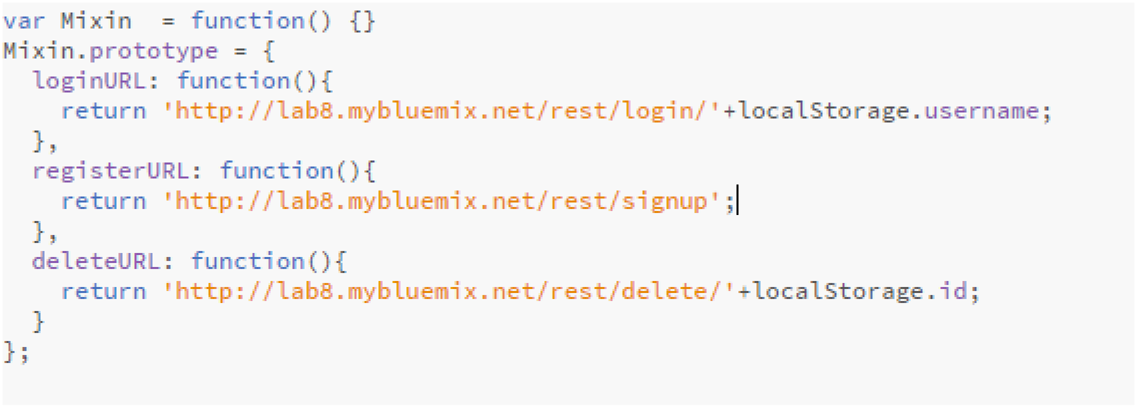
Class Id :11

1. Login screen from app deployed in bluemix



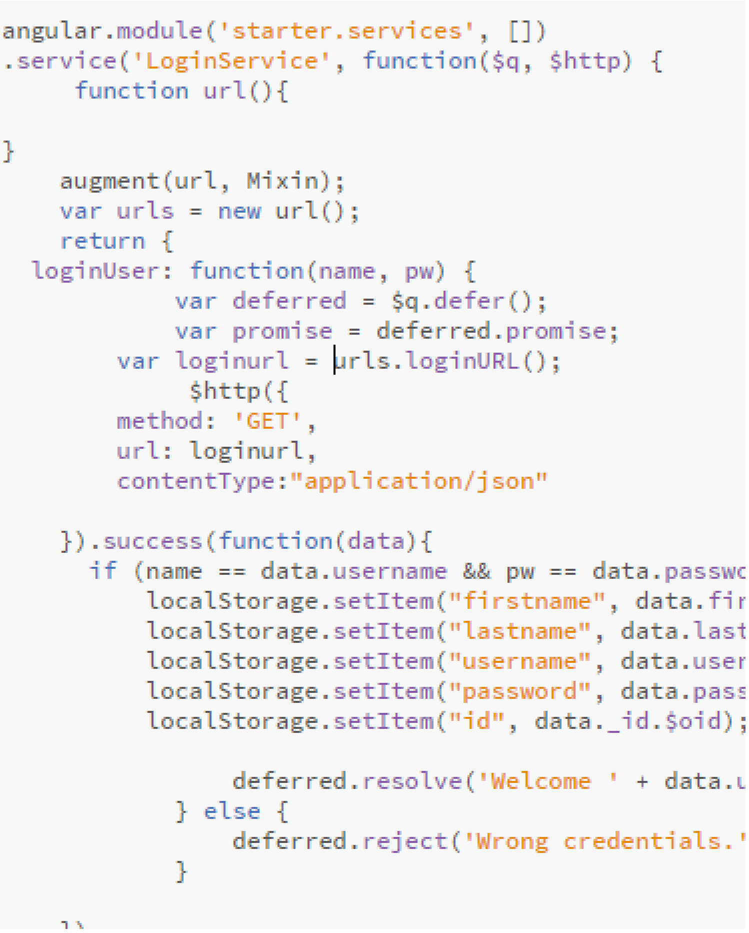
Login , register and delete functionalities make use of Mixin design pattern to fetch the respective URLs to be hit in their http requests.

Mixin:



```
var Mixin  = function() {}
Mixin.prototype = {
  loginURL: function(){
    return 'http://lab8.mybluemix.net/rest/login/'+localStorage.username;
  },
  registerURL: function(){
    return 'http://lab8.mybluemix.net/rest/signup';
  },
  deleteURL: function(){
    return 'http://lab8.mybluemix.net/rest/delete/'+localStorage.id;
  }
};
```

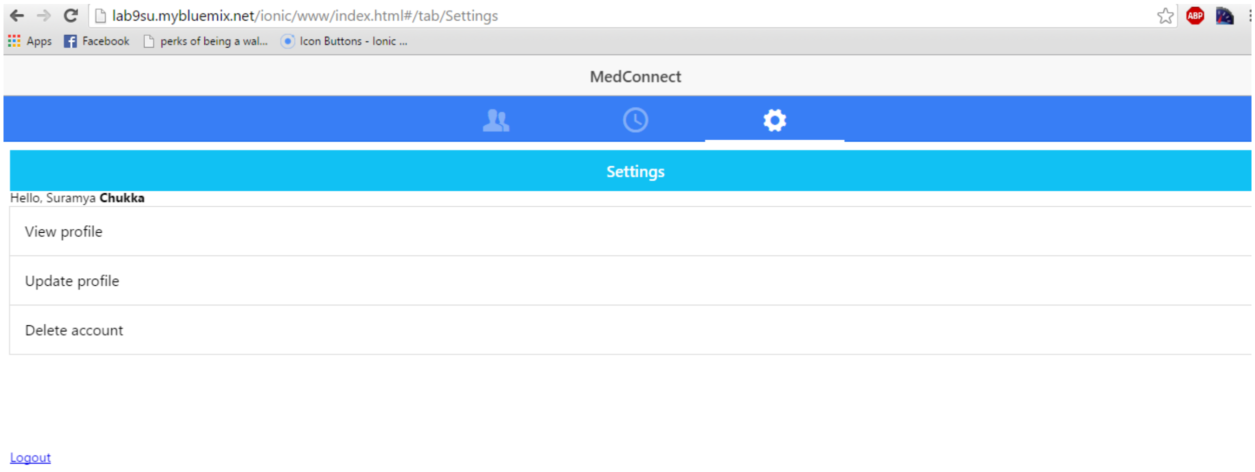
Login Service :



```
angular.module('starter.services', [])
.service('LoginService', function($q, $http) {
     function url(){

}
    augment(url, Mixin);
    var urls = new url();
    return {
  loginUser: function(name, pw) {
            var deferred = $q.defer();
            var promise = deferred.promise;
        var loginurl = urls.loginURL();
             $http({
        method: 'GET',
        url: loginurl,
        contentType:"application/json"

    }).success(function(data){
      if (name == data.username && pw == data.passwo
          localStorage.setItem("firstname", data.fir
          localStorage.setItem("lastname", data.last
          localStorage.setItem("username", data.user
          localStorage.setItem("password", data.pass
          localStorage.setItem("id", data._id.$oid);

                deferred.resolve('Welcome ' + data.u
            } else {
                deferred.reject('Wrong credentials.'
            }

    })
```

Delete and Register functionalities are performed in similar manner.

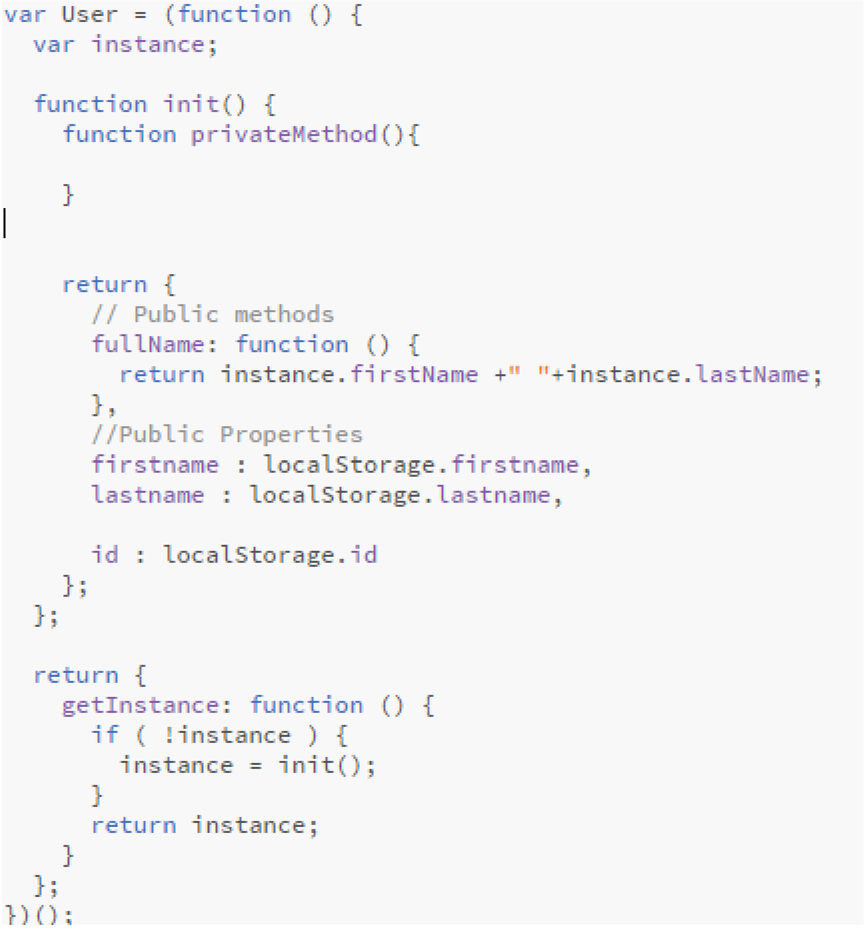
2.Settings screen from app deployed in bluemix



Hello, Suramya **Chukka**

The name displayed on the top of the screen is fetched using Singleton design pattern. Whenever a user logs in , his corresponding first and last name are displayed .

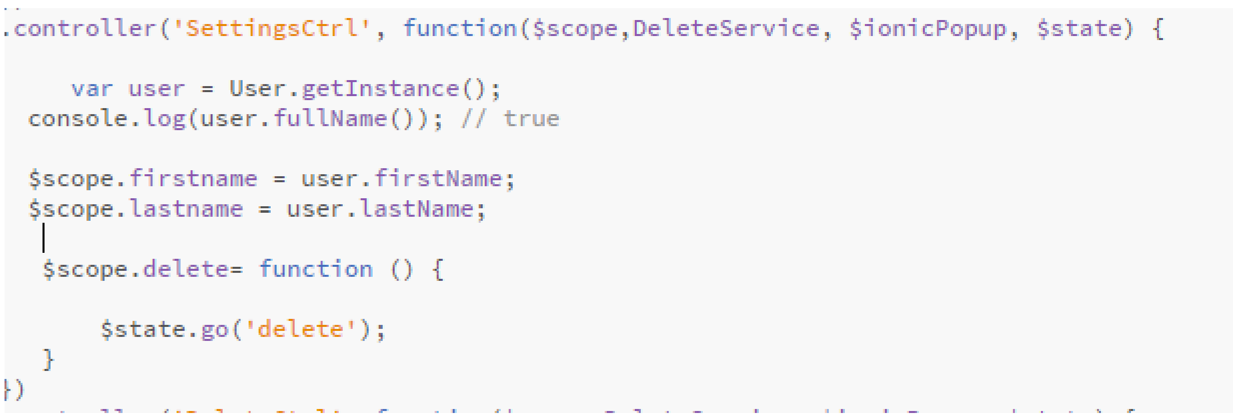
Singleton:



```
var User = (function () {
  var instance;

  function init() {
    function privateMethod(){

    }


    return {
      // Public methods
      fullName: function () {
        return instance.firstName +" "+instance.lastName;
      },
      //Public Properties
      firstname : localStorage.firstname,
      lastname : localStorage.lastname,

      id : localStorage.id
    };
  };

  return {
    getInstance: function () {
      if ( !instance ) {
        instance = init();
      }
      return instance;
    }
  };
})();
```

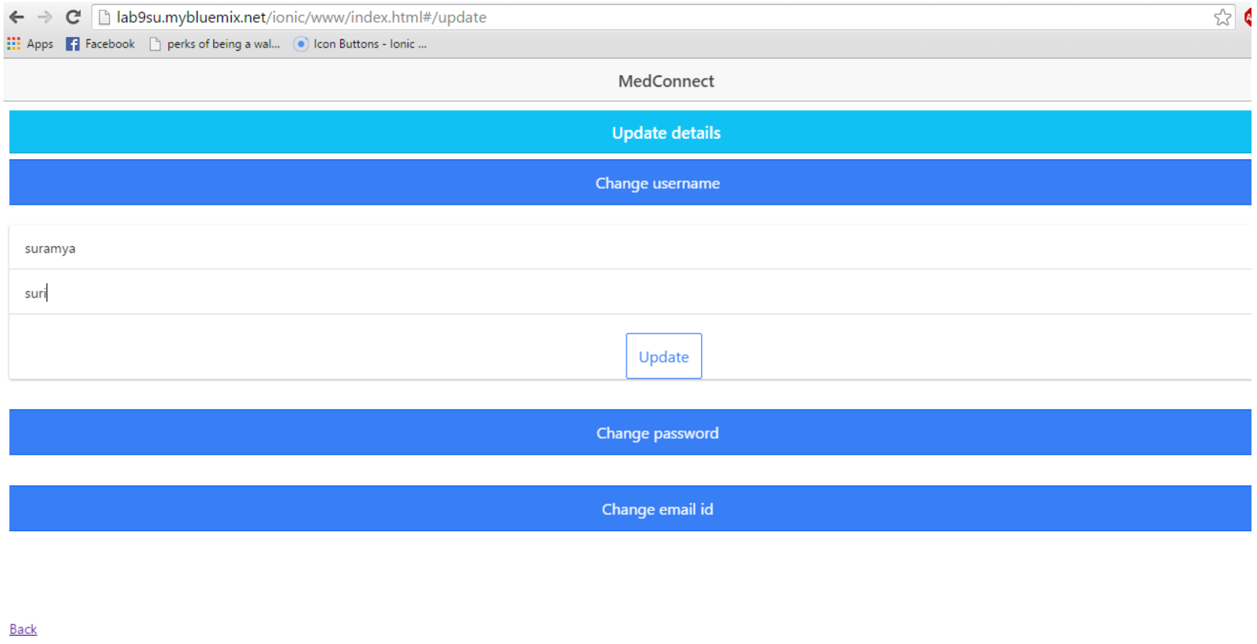
The function defined above is called in Settings controller , to display name at the top of settings screen.

Settings Controller:



```
.controller('SettingsCtrl', function($scope,DeleteService, $ionicPopup, $state) {

     var user = User.getInstance();
  console.log(user.fullName()); // true

  $scope.firstname = user.firstName;
  $scope.lastname = user.lastName;

   $scope.delete= function () {

       $state.go('delete');
   }
})
```

3. Update screen from app deployed in bluemix:



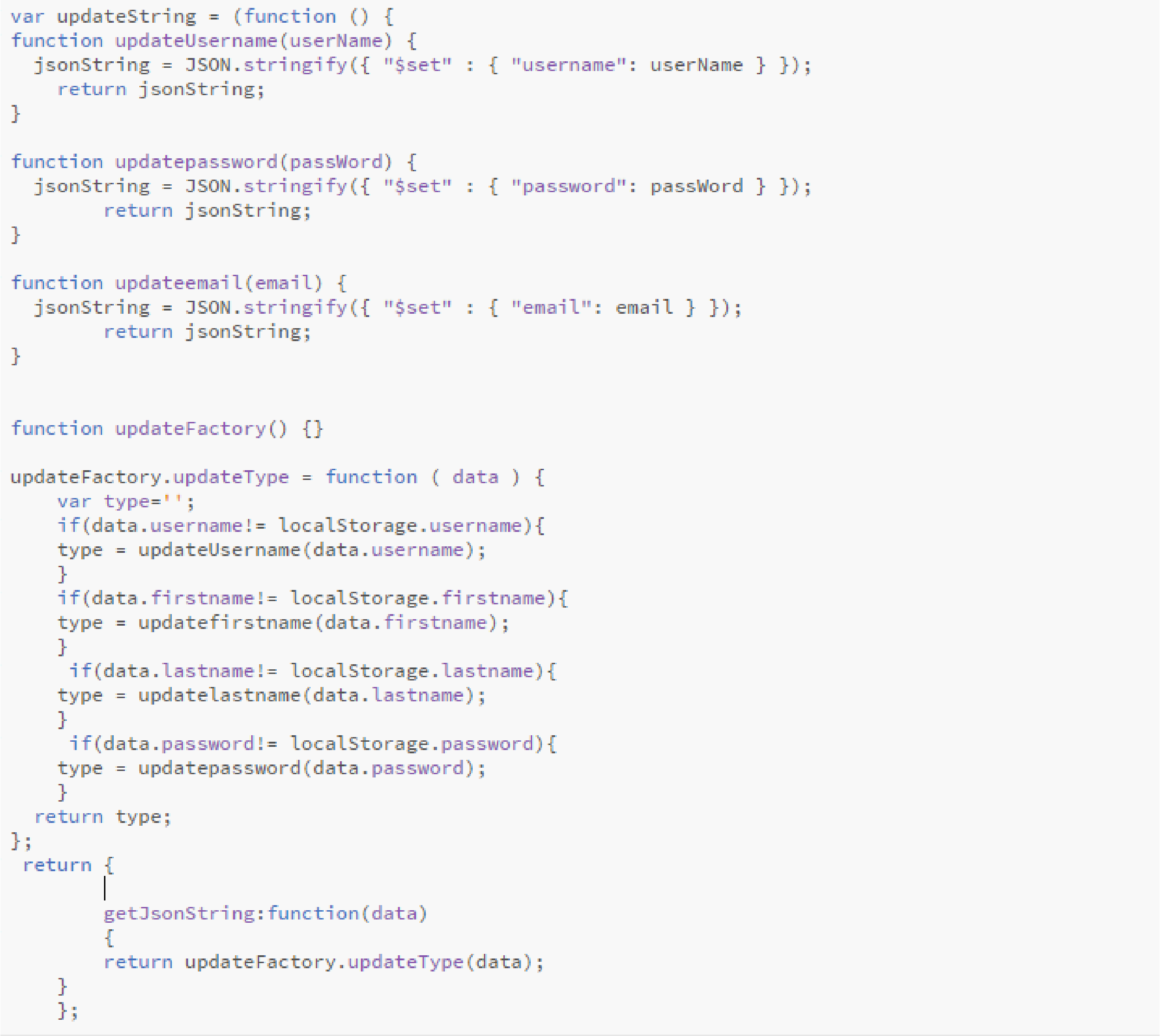
lab9su.mybluemix.net/ionic/www/index.html#/update

suramya

suri

Factory design pattern is used to perform corresponding update operations based on the type to be updated.

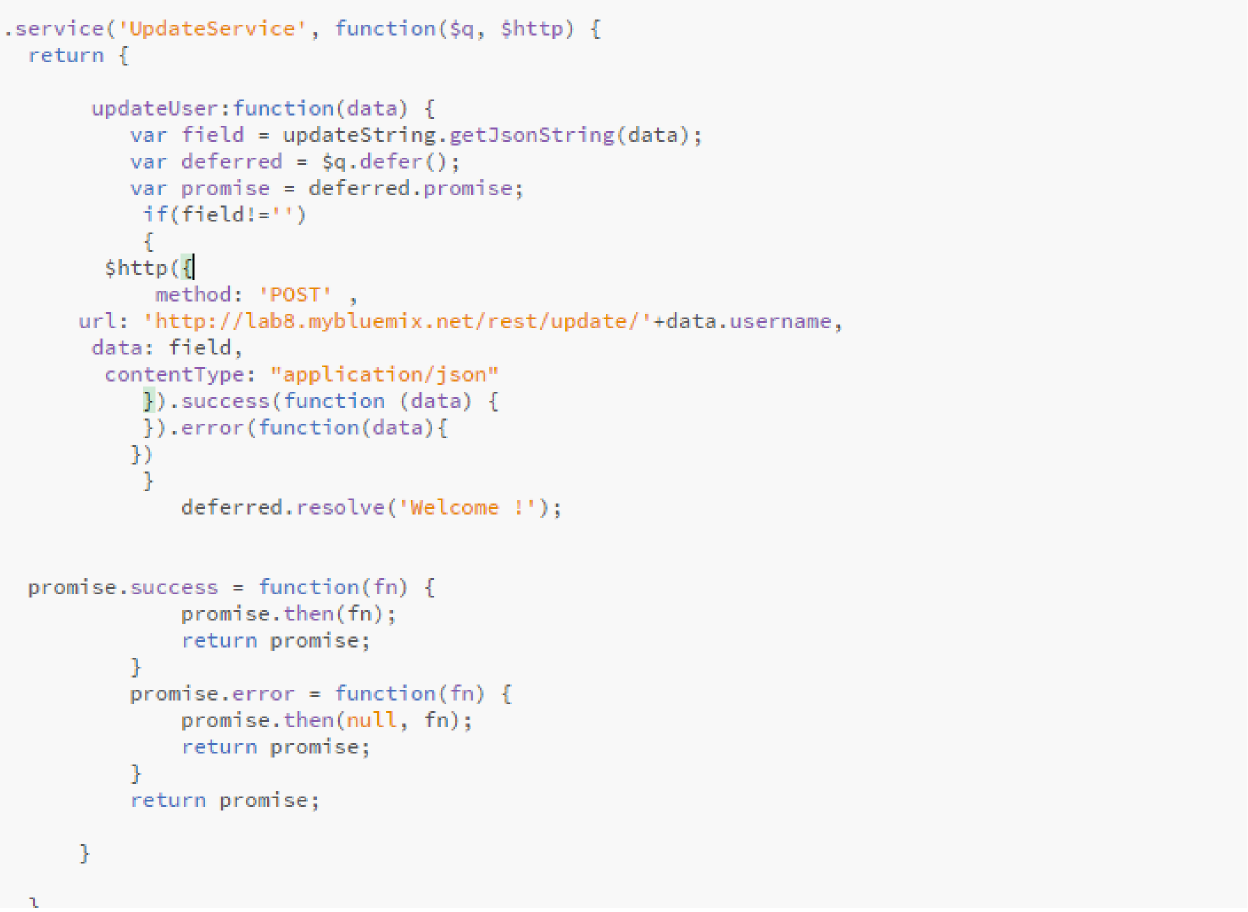
Factory :



```
var updateString = (function () {
function updateUsername(userName) {
  jsonString = JSON.stringify({ "$set" : { "username": userName } });
    return jsonString;
}

function updatepassword(passWord) {
  jsonString = JSON.stringify({ "$set" : { "password": passWord } });
        return jsonString;
}

function updateemail(email) {
  jsonString = JSON.stringify({ "$set" : { "email": email } });
        return jsonString;
}


function updateFactory() {}

updateFactory.updateType = function ( data ) {
    var type='';
    if(data.username!= localStorage.username){
    type = updateUsername(data.username);
    }
    if(data.firstname!= localStorage.firstname){
    type = updatefirstname(data.firstname);
    }
     if(data.lastname!= localStorage.lastname){
    type = updatelastname(data.lastname);
    }
     if(data.password!= localStorage.password){
    type = updatepassword(data.password);
    }
  return type;
};
 return {

        getJsonString:function(data)
        {
        return updateFactory.updateType(data);
    }
    };
```

This factory method is used in the update service of the application. Depending on the updation selected , corresponding data is fetched from factory method to the service and sent through the URL method.

Update Service:



```
.service('UpdateService', function($q, $http) {
  return {

       updateUser:function(data) {
          var field = updateString.getJsonString(data);
          var deferred = $q.defer();
          var promise = deferred.promise;
           if(field!='')
           {
        $http({
            method: 'POST' ,
      url: 'http://lab8.mybluemix.net/rest/update/'+data.username,
       data: field,
        contentType: "application/json"
           }).success(function (data) {
           }).error(function(data){
          })
           }
              deferred.resolve('Welcome !');


  promise.success = function(fn) {
              promise.then(fn);
              return promise;
          }
          promise.error = function(fn) {
              promise.then(null, fn);
              return promise;
          }
          return promise;

      }

  }
```

Bluemix URL:

http://lab9su.mybluemix.net/ionic/www/index.html