

개발 프로젝트 주요코드

■ 프로젝트 명 : Deus Ex Machina

| | | |
|------------------|--|--|
| 소스 코드 (하둡 구현) | class - Request Mapping * 맵 리듀스 | <pre>@RequestMapping(value="/mrCall", method=RequestMethod.POST) public void mrCall(HttpServletRequest req, HttpServletResponse resp) throws Exception { FileSystem hdfs = FileSystem.get(conf); Path path = new Path("/result"); if(hdfs.exists(path)) { hdfs.delete(path, true); } for(soCode = 0; soCode < 5; soCode++) { Job job = Job.getInstance(conf, "test"); URI inputUri = URI.create("/input/csv/" + fileName + ".csv"); URI outputUri = URI.create("/result/" + fileName + soCode FileInputFormat.addInputPath(job, new Path(inputUri)); job.setInputFormatClass(TextInputFormat.class); FileOutputFormat.setOutputPath(job, new Path(outputUri)); job.setOutputFormatClass(TextOutputFormat.class); job.setOutputKeyClass(Text.class); job.setOutputValueClass(IntWritable.class); job.setJarByClass(this.getClass()); job.setMapperClass(WebMapper.class); job.setReducerClass(WebReducer.class); job.waitForCompletion(true); HashMap<String, Object> resultMap = new HashMap<String, Object>(); resp.setCharacterEncoding("UTF-8"); resp.setContentType("text/json;charset=utf-8"); JSONObject json = JSONObject.fromObject(JSONSerializer.toJSON(resultMap)); resp.getWriter().write(json.toString()); } }</pre> |
| | 설명 | <p>AVA의 Hadoop Library를 import 하여 job을 생성한 뒤 linux에 접근, 공공데이터와 검찰청 데이터를 Map Reduce 하여 Linux Hadoop 서버에 저장하는 작업이다. 데이터가 여러 가지이므로 항목별로 나누기 위해 for문을 사용하여 작업하였다.</p> |

[계 속]

소스 코드
(데이터
시각화)

HTML
-
JavaScript

```
//성범죄자 데이터 그리기
function getHdfsData(){
    var hdfsList =[
        {"data":"/result/sexOffender2/part-r-00000"},
        {"data":"/result/sexOffender3/part-r-00000"}
    ];
    hdfsDataList = [];
    $.each(hdfsList, function(index, value) {
        setHdfsDataList(index, value);
    });
}
function setHdfsDataList(index, params){
    $.ajax({
        type : "get",
        url : "/con",
        data : params
    }).done(function(data){
        var resultMap = {"index": index, "data": data};
        hdfsDataList.push(resultMap);
        if(index == 0) drawChart(index);
    });
}
function drawChart(index){
    var result = hdfsDataList[index]["data"].result;
    var chartData = new google.visualization.DataTable();
    chartData.addColumn("string", "sido");
    chartData.addColumn("number", "numberOfcases");
    $.each(result, function(index, value){
        var row = [];
        for(var i = 0; i < 2; i++){
            row[i] = (i != 0) ? Number(value[i]) : value[i];
        }
        chartData.addRow([ row ]);
    });
    var option = [{
        chartType : 'PieChart',
        options : {title: '지역별 성범죄자 수', pieHole : 0.4},
        containerId : 'chartList1'
    },{
        chartType : 'ColumnChart',
        options : {title: '세부지역별 성범죄자 수'},
        containerId : 'chartList2'
    }];
    var op = option[index];
    op.dataTable = chartData;
    var wrapper = new google.visualization.ChartWrapper(op);
    wrapper.draw();
}
```

[계 속]

| | | |
|--------------------------------|----------------------------------|---|
| <p>소스 코드 (데이터 시각화)</p> | <p>HTML - JavaScript</p> | <pre> /*****성범죄자 데이터 시각화*****/ \$("#sexOffender").on("click", function(){ crCheck = 0; mainIndex = 0; \$(".chartList").empty(); \$(".labelMenu").css("display", "none"); \$("#fourthTest").css("display","none"); \$("#label1").text("지역별 성범죄자 수"); \$("#label2").text("세부지역별 성범죄자 수"); \$("#slide-1-trigger").prop("checked", true); google.charts.load('current'); google.charts.setOnLoadCallback(getHdfsData); \$("#label label").off().on("click", function(){ mainIndex = \$("#label label").index(this); drawChart(mainIndex); }); }); /*****범죄 데이터 시각화*****/ \$("#criminalAnalytics").on("click", function(){ crCheck = 1; mainIndex = 0; \$(".chartList").empty(); \$(".labelMenu").css("display", "inline-block"); \$("#label1").text("범죄발생장소"); \$("#label2").text("연령 및 성별"); \$("#label3").text("범죄자 교육수준"); \$("#label4").text("범죄발생요일"); \$("#label5").text("범죄자 직업"); \$("#label6").text("범죄발생시간"); \$("#label7").text("범행동기"); \$("#label8").text("범행 후 은신처"); \$("#fourthTest").css("display","block"); \$("#slide-1-trigger").prop("checked", true); google.charts.load('current'); google.charts.setOnLoadCallback(drawVisualization); \$("#label label").off().on("click", function(){ mainIndex = \$("#label label").index(this); subIndex = 0; createChartView(dataUrlList[mainIndex]["Option"], dataList[mainIndex]["data"]); }); }); </pre> |
|--------------------------------|----------------------------------|---|

[계 속]

| | | |
|--------------------------------|----------------------------------|---|
| <p>소스 코드 (데이터 시각화)</p> | <p>HTML - JavaScript</p> | <pre> /*****차트 타입 버튼*****/ \$("#fourthTest a").on("click", function(){ subIndex = \$("#fourthTest a").index(this); createChartView(dataUrlList[mainIndex]["Option"], dataList[mainIndex]["data"]); }); function createChartView(option, data){ // json 데이터 변수 담기 var result = data.result; var columns = data.columns; var size = data.size; // 컬럼 정보 확인 (예외처리) if(columns.length == 0){ alert("데이터가 잘못되었습니다."); return false; } // 구글 차트에 사용 할 데이터 만들기 var chartData = new google.visualization.DataTable(); // 컬럼 데이터 설정 \$.each(columns, function(index, value) { chartData.addColumn(value.type, value.value); }); // 차트 데이터 설정 \$.each(result, function(index, value) { var row = []; for(var i = 0; i < columns.length; i++){ row[i] = value[columns[i].column]; } chartData.addRow([row]); }); option.dataTable = chartData; if(subIndex == 0){ option.chartType = 'BarChart'; }else if(subIndex == 1){ option.chartType = 'ColumnChart'; }else if(subIndex == 2){ option.chartType = 'PieChart'; }else if(subIndex == 3){ option.chartType = 'ComboChart'; }else if(subIndex == 4){ option.chartType = 'BubbleChart'; } wrapper = new google.visualization.ChartWrapper(option); wrapper.draw(); } /*****차트 크기 조절*****/ \$(window).resize(function() { if(crCheck == 0){ drawChart(mainIndex); }else { createChartView(dataUrlList[mainIndex]["Option"], dataList[mainIndex]["data"]); } }); </pre> |
|--------------------------------|----------------------------------|---|

[계 속]

소스 코드
(검색 기능)

HTML
-
JavaScript

```

/*****검색 기능*****/
$("#containerSearch").submit(function(event){
    var pageNum = 0;
    event.preventDefault();
    if($("#so-search").text() == "다음"){
        if($("#so-name").val() == "" || $("#so-name").val() == null){
            alert("검색할 내용이 없습니다. 정확히 입력해주세요.");
        } else {
            soName2 = $("#so-name").val();
            $("#so-name").val("");
            $("#so-name").attr("placeholder", "이름을 입력하세요.");
            $("#so-search").text("검색");
        }
    } else {
        if($("#so-name").val() == "" || $("#so-name").val() == null){
            alert("검색할 내용이 없습니다. 정확히 입력해주세요.");
        } else {
            createHTML(1, pageNum, soCheck);
        }
    }
});
/*****좌표 표시 이벤트 *****/
var firstCd = $(".soSer").eq($(this).index());
var finalCd = firstCd.children(".tdAl").text();
if(finalCd == null || finalCd == ""){
    alert("잘못된 좌표입니다. 다시 클릭해주세요.");
} else {
    mapDraw(finalCd, firstCd.children(".roAl").text());
}

/*****지도 함수*****/
function mapDraw(msoCoordinate, msoName){
    var mapContainer = document.getElementById('map'),
    mapOption = {
        center: new daum.maps.LatLng(33.450701, 126.570667),
        level: 1
    };
    var map = new daum.maps.Map(mapContainer, mapOption);
    var geocoder = new daum.maps.services.Geocoder();
    geocoder.addressSearch(msoCoordinate, function(result, status) {
        if (status === daum.maps.services.Status.OK) {
            var coords = new daum.maps.LatLng(result[0].y, result[0].x);
            var marker = new daum.maps.Marker({
                map: map,
                position: coords
            });
            var infowindow = new daum.maps.InfoWindow({
                content: '<div
style="width:150px;text-align:center;padding:6px 0;">' + msoName + '</div>'
            });
            infowindow.open(map, marker);
            map.setCenter(coords);
        }
    });
}
mapDraw(msoCoordinate, msoName);

```

[계 속]

| | | |
|----------------------|-------------------------|--|
| 소스 코드 (페이징 처리) | HTML - JavaScript | <pre> function createHTML(page, pageNum, soCheck){ var soName = \$("#so-name").val(); var d = {"soName" : soName, "soName2" : soName2, "pageNum" : pageNum, "soCheck" : soCheck}; \$.ajax({ type : "post", url : "/soName", data : d }).done(function(data){ var d = JSON.parse(data); var totalData = d.result.cnt; var result = d.result2; if(totalData == null totalData == 0){ alert("성범죄자로 등록되지 않은 이름입니다."); }else { paging(totalData, dataPerPage, page); \$(".roAl").empty(); \$(".tdAl").empty(); \$(".centerAl").empty(); for(var i = 0; i < result.length; i++){ \$.each(result[i],function(key,value){ if(key == "name"){ \$(".roAl").eq(i).text(value); } else if(key == "realResidence"){ \$(".tdAl").eq(i).text(value); } else if(key == "date") { \$(".centerAl").eq(i).text(value); } }); } \$(".soSer").css("cursor", "pointer"); \$(".soSer").hover(function() { \$(this).css("color", "red"); }, function(){ \$(this).css("color", "black"); }); } }); } function paging(totalData, dataPerPage, currentPage){ var totalPage = Math.ceil(totalData/dataPerPage); var first = 1; var last = 0; if(totalPage >= pageCount){first = Number(currentPage); last = (first + pageCount) - 1; }else {last = totalPage;} if(first < pageCount){first = 1;last = 5; if(totalPage < last){last = totalPage;} }else {first = first - 2; last = (first + pageCount) - 1; if(totalPage < last){last = totalPage;} if(last >= (totalPage - 4)){first = last - 4;} } var \$pagingView = \$("#paging"); var html = ""; for(var i=first; i <= last; i++){ html += "" + i + " "; } \$("#paging").html(html); \$("#paging a").css({"color":"black", "text-decoration":"none"}); \$("#paging a#" + currentPage).css({"color":"red", "font-weight":"bold"}); \$("#paging a").off().on("click",function(){ var \$item = \$(this); var \$id = \$item.attr("id"); pageNum = (Number(\$id) - 1) * dataPerPage; createHTML(\$id, pageNum, soCheck); }); } </pre> |
|----------------------|-------------------------|--|