

解决跨域iframe问题

```
<script>
    document.domain=domainConfig.documentDomain;
</script>
```

```
var domainConfig = {
    documentDomain:'ahau.edu.cn',// 同域
}
```

ids密码加密

```
public static void main(String[] args) throws Exception {
    String loginName = "0211180535";
    String password = "abc383868";
    String md5Pwd = SUDY.idsPasswordEncrypt(loginName, password);
    System.out.println(md5Pwd);
    String field29 = SUDY.idsField29Encrypt(loginName, password);
    System.out.println(field29);
    // String aes = com.sudytech.security.ids.util.AES.decrypt(field29, "ids#s12" + loginName);
    // System.out.println(aes);
    String dbField29=
"205BDD62D83ACED1B9027DE9117C1591374D25824E6E176919E573F61B4597FB406E230C1FEED966027DD213ABD7669F";
    System.out.println(SUDY.verifyIdsField29(dbField29, loginName, password));
    //String ldapMd5 = "{MD5}" + SUDY.idsPasswordEncrypt("", password);
    // System.out.println(ldapMd5);
}
```

C:\Windows\System32\drivers\etc

192.168.120.80 www.cas.com

192.168.120.80 www.imp.com

Eclipse Class Decompiler Project Update

<http://raw.githubusercontent.com/cnfree/eclipse/master/decompiler/update/>

E:\Program Files (x86)\jad158g.win\jad.exe

E:\Program Files (x86)\jad158g.tmp

url中文转换 encodeURI(encodeURI(name)) decodeURI(name)

sudy加密

```
String md5Pwd = IdsServerAux.encryptByMd5(new
StringBuilder().append(user2.getLoginName()).append(oldpassword).toString());
```

sql中文排序 SELECT id,name FROM `T_USER` ORDER BY convert(name using gbk) ASC limit 10,100;

```

$('#passwordAnswer').textbox('textbox').validatebox('options').required = false;
$('#name').validatebox({required:false});

//$('#passwordAnswer').textbox('setValue','1');
//$('#password').passwordbox('hidePassword');

<filter>
<filter-name>CookieCheckFilter</filter-name>
<filter-class>com.sudytch.auth.basic.check.CookieCheckFilter</filter-class>
</filter>
<filter-mapping>
<filter-name>CookieCheckFilter</filter-name>
<url-pattern>*/</url-pattern>
</filter-mapping>

```

NetBeans快捷键

Alt+Insert get/set

代码格式化: 快捷键 Alt+Shift+F

修复导入: 快捷键 Ctrl+Shift+I

```

setTimeout(function() {
    SUI.init();
    init();
    layer.close(loading);
}, 500);

var obj = document.getElementById(field);
var demo = SUI.build(obj, 'select'); // 创建一个组件实例, 并执行初始化渲染
$(obj).addClass('col-xs-8');
var confi = $(obj).sui().getConfig();
var src = $("#messagePrimaryKey").sui().getConfig();
confi.properties.name = field;
confi.properties.title = field;
//confi.properties.dataSource = src.properties.dataSource;
confi.properties.selectStyle = src.properties.selectStyle;
confi.mode = src.mode;
confi.validate = $("#name").sui().getConfig().validate;
$(obj).sui().setConfig(confi);

```

===== web =====

超出部分感叹号表示

```
h3.app-name {
    display: inline-block;
    max-width: 160px;
    overflow: hidden; /*超出部分隐藏*/
    text-overflow: ellipsis; /*超出部分感叹号显示*/
    white-space: nowrap; /*一行显示*/
}
```

图片圆角

```
.user-info.user-avatar img {
    overflow: hidden;
    border-radius: 20px;
    width: 20px;
}
```

//获取浏览器宽度

```
function getBrowserWidth() {
    alert($(window).width());
}
```

//获取屏幕宽度

```
function getScreenWidth() {
    alert(window.screen.width);
}
```

//css 悬浮事件

```
li:hover,.sl:hover{
    opacity:0.6; //透明度
    filter:alpha(opacity=60); /* IE8 及更早版本 */
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    border-radius:20px;//边框圆角
    border:1px solid #aaa;
}

ul{
    list-style-type:none;
    margin:0;
    padding:0;
}

li{
    display:inline-block;
}
```

===== java时间格式 =====

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
sdf.format(new Date(System.currentTimeMillis()));
```

```
String useDeptEmail = WebplusEnv.getConfigFactory().getConfig("config").getString("ucp.useDeptEmail");
```

```
boolean isRunning = ThreadConsole.getInstance().isRunningThead(OrigImportThread.class);
```

```
$.fn.validatebox.defaults.rules.email.message = '请输入有效的<br/>电子邮件地址';
```

===== 服务器布署多个tomcat =====

1.HTTP端口，默认8080，如下改为8081

[html] view plain copy

```
<Connector port="8081" protocol="HTTP/1.1"
            connectionTimeout="60000"
            redirectPort="8443" disableUploadTimeout="false"
```

```
executor="tomcatThreadPool" URIEncoding="UTF-8"/>
```

2. 远程停服务端口，默认8005，如下改为8006

[html] view plain copy

```
<Server port="8006" shutdown="SHUTDOWN">.....
```

3.AJP端口，默认8009，如下改, 8010

[html] view plain copy

```
<Connector port="8010" protocol="AJP/1.3" redirectPort="8443" />
```

alert回调

```
$.messenger.alert("提示:", msg, "info",function() {
    $('#userDeptEmailFrom').focus();
});
```

一秒刷新

```
setTimeout(function() {
    window.location.reload();
},1000);
```

滚动条设置在顶端

```
$('#body').scrollTop( $('#body').height() );
```

```

===== js radio选中事件=====
$("input[name='radioName']").on('ifChecked', function(event){
    test();
});
$.ready(function() {
    $("input[name='statisticsTimeRadio']").on('click', function(){
        var type = $("input[name='statisticsTimeRadio']:checked").attr("value");
        if(type==0){
            $('#statisticsTime').hide();
        }else{
            $('#statisticsTime').show();
        }
    });
});

$("#rdol").prop("checked", "checked");

$(".selector").find("option[text='pxx']").attr("selected", true);

```

```

function initCheckBox() {
    $('input[type="radio"], input[type="checkbox"]')
    .iCheck({
        checkboxClass : 'icheckbox_minimal-blue',
        radioClass : 'iradio_minimal-blue'
    });
    $('input[name="checkall"]').on('ifChecked', function(event) {
        $('input[type="checkbox"]').iCheck('check');
    });
    $('input[name="checkall"]').on('ifUnchecked', function(event) {
        $('input[type="checkbox"]').iCheck('uncheck');
    });
}

```

```

===== js 正则表达式=====
/*是否带有小数*/
function isDecimal(strValue) {
    var objRegExp= /\d+\.\d+$/;
    return objRegExp.test(strValue);
}

/*校验是否中文名称组成 */
function ischina(str) {
    var reg=/^[\u4E00-\u9FA5]{2,4}$/;    /*定义验证表达式*/
    return reg.test(str);    /*进行验证*/
}

/*校验是否全由8位数字组成 */

```

```
function isStudentNo(str) {
    var reg=/^[0-9]{8}$/;    /*定义验证表达式*/
    return reg.test(str);    /*进行验证*/
}
```

/*校验电话码格式 */

```
function isTelCode(str) {
    var reg= /^(0\d{2,3}-\d{7,8})|(1[3584]\d{9}))$/;
    return reg.test(str);
}
```

```
var reg = /^(-){1}\d+|\d+$/;
```

/*校验邮件地址是否合法 */

```
function IsEmail(str) {
    var reg=/^([a-zA-Z0-9_-])+@([a-zA-Z0-9_-])+(\.[a-zA-Z0-9_-])+/;
    return reg.test(str);
}
```

```
String path = this.getRequest().getSession().getServletContext().getRealPath("/");
String demoPath = "_web/ids/importUserToOrg/importDemo.xls";
realPath = path+demoPath;
settFileName("importUserDemo.xls");
```

//联动单选框

```
$('#radio21').on('ifChecked', function(event) {
    $('#radio22').iCheck('check');
```

```

        $('#radio24').iCheck('uncheck');
    });
    $('#radio23').on('ifChecked', function(event) {
        $('#radio22').iCheck('uncheck');
        $('#radio24').iCheck('check');
    });

    //时间选择框
    timebox('#datebox');
    //demo必填文本框：添加清空文本框信息
    $('#tt').textbox().textbox('addClearBtn', 'icon-clear');

    //demo文本域：添加事件
    $('#textarea').textbox('textbox').on('click', function() {
        console.log(1);
    });
    //复选框事件
    $(".checkbox input").each(function(i, el) {
        $(this).on('ifChecked', function(event) {
            switch(i)
            {
                case 0:
                    console.log('第一个按钮已选中');
                    break;
                case 1:
                    console.log('第二个按钮已选中');
                    break;
                default:
                    //console.log(event.type);
            }
        });
        $(this).on('ifUnchecked', function(event) {
            switch(i)
            {
                case 0:
                    console.log('第一个按钮已移除');
                    break;
                case 1:
                    console.log('第二个按钮已移除');
                    break;
                default:
                    //console.log(event.type);
            }
        });
    });
    //复选框, 单选框：添加皮肤
    $('#input[type="checkbox"], input[type="radio"]').iCheck({
        checkboxClass: 'icheckbox_minimal-blue',
        radioClass: 'iradio_minimal-blue',
        increaseArea: '20%'
    });

```

```
});
```

```
$("#mailbox").html('');  
var confi = $("#mailbox").sui().getConfig();  
confi.source = source + node.id;  
$("#mailbox").sui().setConfig(confi);  
addRemoveButton("mailbox", "添加<%= type%>", "false", '<%=sureUrl%>');
```

```
$("#mailbox").html(html);  
SUI.init();  
var confi = $("#mailbox").sui().getConfig();  
confi.source = source + node.id;  
$("#mailbox").sui().setConfig(confi);  
extendParam();
```

```
function extendParam() {  
    var url = '<%=bindMessageBoxUser%>';  
    var params = '';  
    addSureButton("mailbox", "绑定账号", "请先选择将要绑定的人员!", url, params, function() {  
        callback_Url("/_web/appMessageBox/bindUserManage.jsp", "refresh");  
        //parent.closeLayer();  
    });  
};
```

```
$("#search").on('click', function(event) {  
    event.preventDefault();  
    $("#mailbox").sui().tableApi().ajax.reload(function() {  
        }, false);  
});
```

```
$("#reset").on('click', function(event) {  
    event.preventDefault();  
    $("#artState").sui().setValue("0");  
    $("#keyword").sui().setValue('');  
    $("#mailbox").sui().tableApi().ajax.reload(function() {  
        }, false);  
});  
}
```



```
var str = $("#textLimit").sui().getValue();
```

```
private void checkDomainManagerPerm() throws Exception {
    SSOUser ssoUser = CoreRequestContextUtil.getCoreRequestContext().getCurrentUser();
    int domainId = Integer.parseInt(CoreRequestContextUtil.getCoreRequestContext().getDomainId());
    PermissionManager permManager = new PermissionManager(ssoUser, domainId);
    if (!permManager.hasDomainManagePerm()) {
        throw new Exception("无权限操作");
    }
}
```

角色管理-角色人员-导入

SUI.settings().uploadify

//单选框、复选框样式

```
function checkboxAndRadioStyle() {
    $('input[type="checkbox"],input[type="radio"]').iCheck({
        checkboxClass: 'icheckbox_minimal-blue',
        radioClass: 'iradio_minimal-blue',
        increaseArea: '20%'
    });
}
```

--- radio

```
$("#input[name='display']").on('ifChecked', function(event) {
    if($("#display2").prop('checked')) {
        $("#perEdit1").iCheck('unCheck');
        $("#perEdit2").iCheck('check');
    }
});
```

--- checkbox

```
$('#field6Box').on('ifChecked', function(event) {
    $('#field1Box').iCheck('check');
});
/* $('#field6Box').on('ifUnchecked', function(event) {
    if(!$('#field7Box').prop('checked')) {
        $('#field1Box').iCheck('unCheck');
    }
}); */
```

```
/* function hoverRowSrc(obj) {
    var img = '';
```

```

        if(!$(obj).is(':has(img)')) ){
            $(obj).append$(img);
        }
    }
}

function outRowSrc(obj) {
    if($(obj).is(':has(img)')) ){
        setTimeout(function() {
            $(obj).find('img').remove();
        }, 50)
    }
}

function initCheckAll() {
    $('input[name="input1"]').on('ifChecked', function(event) {
        $('div[name="divSrc"]').each(function() {
            $(this).addClass('checked');
        });
        if($('div[name="divSrc"]').length>0) {
            $('button[data-tool="right"]').removeAttr("disabled");
        }
    });
    $('input[name="input1"]').on('ifUnchecked', function(event) {
        $('div[name="divSrc"]').each(function() {
            $(this).removeClass('checked');
        });
        $('button[data-tool="right"]').attr("disabled", "");
    });
}

function selectSrc(obj) {
    var st = $(obj).attr('class');
    setTimeout(function() {if(st.indexOf('checked')>-1) {
        $(obj).removeClass('checked');

        var flag = true;
        $('div[name="divSrc"]').each(function() {
            if($(this).hasClass("checked")) {
                flag = false;
            }
        })
        if(flag) {
            $('button[data-tool="right"]').attr("disabled", "");
        }

    }else{
        $(obj).addClass('checked');
        $('button[data-tool="right"]').removeAttr("disabled");
    }}, 20);
}
}
*/

```

```
var bb = 'function play() {alert("getplay");}';
```

```
var ex = {getplay:eval("("+ss+")")};
```

JSON.parse(string) : 接受一个 JSON 字符串并将其转换成一个 JavaScript 对象。
JSON.stringify(obj) : 接受一个 JavaScript 对象并将其转换为一个 JSON 字符串。

特殊字符正则

http://blog.csdn.net/al_ assad/article/details/76383107
tomcat配置JAVA_HOME

```
export JAVA_HOME=/usr/lib/jvm/java-openjdk
export JRE_HOME=/usr/lib/jvm/java-openjdk/jre
```

之后再重新启动Tomcat，返回信息正确：

```
Using CATALINA_BASE:   /apache-tomcat-8.5.16
Using CATALINA_HOME:   /apache-tomcat-8.5.16
Using CATALINA_TMPDIR: /apache-tomcat-8.5.16/temp
Using JRE_HOME:        /usr/lib/java-1.8.0-opensdk/jre
Using CLASSPATH:        /apache-tomcat-8.5.16/bin/bootstrap.jar:/apache-tomcat-8.5.16/bin/tomcat-juli.jar
```

Tomcat started.

<https://www.cnblogs.com/kyosusan/p/5198934.html>

修改root密码

一、知道原来的mysql数据库的root密码;

①: 在终端命令行输入 `mysqladmin -u root -p password "新密码"` 回车, Enter password: 【输入原来的旧密码】

②: 登录mysql系统修改, `mysql -uroot -p` 回车 Enter password: 【输入原来的密码】

```
mysql>use mysql;
```

```
mysql> update user set password=password("新密码") where user=' root';
```

 【密码注意大小写】

```
mysql> flush privileges;
```

```
mysql> exit;
```

layer.close

```
APP.preload(['_docs/js/docs.min.js', '_docs/css/docs.css',
'_docs/js/highlight/styles/default.css', '_libs/plugins/zTree/css/metroStyle/metroStyle.css', '_libs/plugins/personnel-
selection/ztree.css', '_libs/plugins/personnel-selection/ztree.css', '_libs/plugins/personnel-selection/app.css']); //
启动预加载, 参数为预览加载的插件, string | array
    var adjustHeightSwitch = false;
```

t_sns_friend

oracle查询:分组查询, 取出每组中的第一条记录

<https://blog.csdn.net/yatou5211/article/details/53764676>

```
SELECT * FROM(
SELECT z.type , z.code ,ROW_NUMBER()
OVER(PARTITION BY z.type ORDER BY z.code) AS code_id
FROM group_info z
)
WHERE code_id =1;
```

```
$("#tt").tabs({
    onSelect : function( title ){
        <!-- if(title == '短信'){
            if(!$("#smsDetail").attr("src")){
                $("#smsDetail").attr("src","<%=smsDetailPage%>");
            }
        }
    }
});
```

```

    }
} else if(title == '邮件'){
    if(!$("#emailDetail").attr("src")){
        $("#emailDetail").attr("src", "<%=emailDetailPage%>");
    }
} else if(title == '即时消息'){
    if(!$("#imDetail").attr("src")){
        $("#imDetail").attr("src", "<%=imDetailPage%>");
    }
} else if(title == '微信'){
    if(!$("#wxDetail").attr("src")){
        $("#wxDetail").attr("src", "<%=wxSendDetail%>");
    }
} --%>
<%-- if(!$("#smsDetail").attr("src")){
    $("#smsDetail").attr("src", "<%=smsDetailPage%>");
} --%>
var pp = $('#tt').tabs('getSelected');
var ifr = $("#iframe", pp)[0];
if(!ifr.attr("src")){
    var url = ifr.attr("tarsrc");
    ifr.attr("src", url);
}
}

```

java日期

获取前几天、后几天的日期

```

public String getSpecifiedDay(String specifiedDay,int num){
    Calendar c = Calendar.getInstance();
    Date date=null;
    try {
        date = new SimpleDateFormat("yy-MM-dd").parse(specifiedDay);
    } catch (ParseException e) {
        e.printStackTrace();
    }
    c.setTime(date);
    int day=c.get(Calendar.DATE);
    c.set(Calendar.DATE, day+num);

    String dayAfter=new SimpleDateFormat("yyyy-MM-dd").format(c.getTime());
    return dayAfter;
}

```

日期相隔多少天

```

private static SimpleDateFormat sdf1 = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
private static SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy-MM-dd");

```

```

Date begin = sdf2.parse(beginTime);
Date end = sdf2.parse(endTime);
long day = (end.getTime()-begin.getTime())/(24*60*60*1000);

```

日期相隔多少分钟

```

protected String decodeOnlineTime(Date loginDateTime) {
    // TODO Auto-generated method stub
    if(loginDateTime==null){
        return "";
    }

    Date end = new Date(System.currentTimeMillis());
    long between=(end.getTime()-loginDateTime.getTime())/1000;//除以1000是为了转换成秒
    long min=between/60;

    return String.valueOf(min);
}

```

如何删除ssh secure shell中的profile

C:\Users\sudytech\AppData\Roaming\SSH

```
String md5Pwd = IdsServerAux.encryptByMd5(new StringBuilder().append(loginName).append(password).toString());
```

```

<div><input id="email" name="email" value="" style="padding:4px;float:left;width: 250px;height:18px;border-
radius:4px;border:1px solid #ccc;"

```

```

    validType="validEmail" class="easyui-validatebox validatebox-text" panelHeight="auto"></div>

```

validBirthDate

```

$.extend($.fn.validatebox.defaults.rules, {
    validEmail: {
        validator: function(value, param) {
            var regEx = /^[a-zA-Z0-9_]{1,}@((([a-zA-z0-9]){1,}\.){1,3}(com|cn|net)$|/;
            var a = regEx.test(value);
            return a;
        },
        message: '请输入正确邮箱格式'
    }
});

```

```

$('#loginNameInput').textbox('textbox').keyup(function(event) {
    $('#userName').textbox("setValue", '');
    $('#userId').val('0');
    var text = $('#loginNameInput').combobox("getText");
    val = text || val ;

```

```

        text = val;
    if (event.keyCode == "13") {
        $.ajax({
            url : 'getUsersByLoginName.do?domainId=1&loginName='+text,
            type: 'get',
            dataType:'json',
            success:function(result) {
                var data = [];
                data = result;
                $('#loginNameInput').combobox("loadData", data);
                $('#loginNameInput').combobox("showPanel");
                $('#loginNameInput').combobox("setText", text);
            }
        });
    }
});

```

身份证验证

```

var idCard = function (value) {
    //if (value.length == 15) return true;
    var number = value.toLowerCase();
    var d, sum = 0, v = '10x98765432', w = [7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2], a =
'11,12,13,14,15,21,22,23,31,32,33,34,35,36,37,41,42,43,44,45,46,50,51,52,53,54,61,62,63,64,65,71,81,82,91';
    var re = number.match(/^(\\d{2})\\d{4}(((\\d{2})(\\d{2})(\\d{2})(\\d{3}))|((\\d{4})(\\d{2})(\\d{2})(\\d{3}[x\\d]))))$/);
    if (re == null || a.indexOf(re[1]) < 0) return false;
    if (re[2].length == 9) {
        number = number.substr(0, 6) + '19' + number.substr(6);
        d = ['19' + re[4], re[5], re[6]].join('-');
    } else d = [re[9], re[10], re[11]].join('-');
    if (!isDateTime.call(d, 'yyyy-MM-dd')) return false;
    for (var i = 0; i < 17; i++) sum += number.charAt(i) * w[i];
    return (re[2].length == 9 || number.charAt(17) == v.charAt(sum % 11));
}

```

```

$(document).ready(function() {
    $("#fm").form({
        onSubmit: function() {
            return $(this).form("validate");
        },
        success: function(result) {
            if (result == 'success') {
                $.messenger.alert('提示', '保存成功!', 'info');
            } else {
                alertMsg(result);
            }
        }
    });
});
});

```

```

function check()
{
    $('#fm').submit();
}

$.extend($.fn.validatebox.defaults.rules, {
    phoneValidType: {
        validator: function(value, param) {
            return /^1[3|4|5|6|7|8|9][0-9]{9}$/.test(value);
        },
        message: '请输入正确的手机号码'
    }
});

$.extend($.fn.validatebox.defaults.rules, {
    emailValidType: {
        validator: function(value, param) {
            if(value.indexOf('*')>0){
                return false;
            }
            //var reg = /^\\w+((-\\w+)|(\\.\\.\\w+))*\\@[A-Za-z0-9]+((\\.|-)[A-Za-z0-9]+)*\\. [A-Za-z0-9]+$/;
            //return !reg.test(value);
            return /^((([a-z]|\d|!#$%&'*\+\-\/=\?^_`{|}~)|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])+(\.([a-z]|\d|!#$%&'*\+\-\/=\?^_`{|}~)|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])+)*)|((\x22)((\x20|\x09)*(\x0d\x0a))?(\x20|\x09)+)?(((\x01-\x08\x0b\x0c\x0e-\x1f\x7f)|\x21|[\x23-\x5b]|[\x5d-\x7e]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(\.([\x01-\x09\x0b\x0c\x0d-\x7f]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF]))))*(((\x20|\x09)*(\x0d\x0a))?(\x20|\x09)+)?(\x22))@((([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|((([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|([a-z]|\d|_|~)|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])*)|([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF]))\.)+((([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|((([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|([a-z]|\d|_|~)|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])*)|([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF]))\.)?$/i.test(value);
        },
        message: '请输入有效的电子邮件地址'
    }
});

$.extend($.fn.validatebox.defaults.rules, {
    telephoneValidType: {
        validator: function (value, param) {
            return /^(0((\d{2,3})-(\d){7,8})|(\d{2,3})(\d){7,8}))$/i.test(value);
        },
        message: '联系电话不正确'
    }
});

```