

Password Policy Code Changes for SuperSite and PartnerSite

ui/supersite/en/templates/misc/login/includes/customer_signup.html

Line 40 – 46

From

```
<input type="password" id="passwd" name="passwd" class="frm-field required" size="35" value="">
  <p class="info-small">
    <t>8 character minimum. No special characters.</t>
  </p>
```

To

```
<input type="password" id="passwd" name="passwd" class="frm-field" size="35" value="">
  <div class="info-small" id="password-policy">
    <p>Your password must have:</p>
    <ul class="passwordpolicy-required">
      <li id="pwdlength"><t>Between 9-16 characters</t></li>
      <li id="upperchar"><t>Uppercase characters (A-Z)</t></li>
      <li id="lowerchar"><t>Lowercase characters (a-z)</t></li>
      <li id="digit"><t>Numbers (0-9)</t></li>
      <li id="specialchar"><t>Special characters ~*!@$#%_+.?.:.,{}</t></li>
    </ul>
  </div>
```

=====

Line 259

From:

```
<script language="javascript">
```

```

$(document).ready(function () {
    $.validator.addMethod("alphanumeric", function (value, element) {
        return /[a-z]+/i.test(value) && /[0-9]+/i.test(value);
    }
    , "<t>Letters or numbers only please</t>");

    $.validator.addMethod("validcountrycode", function (value, element) {
        return validCountryCode(value);
    }
    , "<t>Please enter a valid country code.</t>");

    $.validator.addMethod("validcity", function (value, element) {
        return validCity(value);
    }
    , "<t>Please enter a valid city.</t>");
    $.validator.addMethod("signup_zipcode_handler", function (value, element) {
        validator = this;
        validZip(value, $("#country").val(), function (is_valid) {
            console.log(is_valid);
            if (is_valid == 'NA') {
                $("#input_zip").attr('readonly', true);
                $("#input_zip").val('N/A');
                submitted = validator.formSubmitted;
                validator.resetInternals();
                validator.toHide = validator.errorsFor( element );
                validator.formSubmitted = submitted;
                validator.successList.push( element );
                validator.invalid[ element.name ] = false;
                validator.showErrors();
            }
            else {

                if (is_valid) {
                    submitted = validator.formSubmitted;
                    validator.resetInternals();
                    validator.toHide = validator.errorsFor( element );
                    validator.formSubmitted = submitted;
                    validator.successList.push( element );
                    validator.invalid[ element.name ] = false;
                    validator.showErrors();
                }
                else {

                    message = validator.defaultMessage(element, {method: 'remote', parameters: value});
                    errors = {};
                    errors[element.name] = message;
                    validator.invalid[element.name] = true;
                    validator.showErrors(errors);
                }
            }
        });
        return "pending";
    }, get_translated_message('contacts_invalidzip'));

    // validate signup form on keyup and submit
    $("#sign").validate({
        rules: {
            name: "required",
            address1: "required",
            city:
            {

```

```

        required: true,
        validcity: true
    },
    telnoccc:
    {
        required: true,
        validcountrycode: true
    },
    telno:
    {
        required: true,
        number: true,
        minlength: 6
    },
    mobileno:
    {
        required: false,
        number: true,
        minlength: 6,
        maxlength: 12
    },
    passwd:
    {
        required: true,
        minlength: 8,
        alphanumeric: true
    },
    conf_passwd:
    {
        required: true,
        minlength: 8,
        equalTo: "#passwd",
        alphanumeric: true
    },

    email: {
        required: true,
        email: true
    },
    vat_id: {
        remote: {
            url: $.url("/misc/eu_country.php"),
            type: "GET",
            cache: false,
            dataType: "json",
            data: {
                vat_id: function () {
                    return $("#input_vat").val();
                },
                country_id: function () {
                    return $("#country").val();
                }
            }
        }
    },

    zip: {
        required: true,
        signup_zipcode_handler: true
    },
},
messages: {
    name: "<t>Please specify your name</t>",

```

```

        address1: "<t>Please specify your address</t>",
        zip: "<t>Please specify your zip/postcode</t>",
        telno: {
            required: "<t>Please provide us your contact number</t>"
        },
        mobileno: {
            required: "<t>Please provide us your mobile number</t>"
        },
        email: {
            required: "<t>You need a valid email address to log into the website</t>",
            email: "<t>Your email address must be in the format name@domain.com</t>"
        },
        passwd: {
            required: "<t>Please specify your password</t>",
            minlength: "<t>Your password must be at least 8 characters long</t>",
            maxlength: "<t>Your password can contain a maximum of 15 characters</t>",
            alphanumeric: "<t>The password should contain alpha numeric characters</t>"
        },
        conf_passwd: {
            required: "<t>Please confirm your password</t>",
            minlength: "<t>Your password must be at least 8 characters long</t>",
            equalTo: "<t>Please enter the same password as above</t>",
            alphanumeric: "<t>The password should contain alpha numeric characters</t>"
        },
        vat_id: {
            remote: get_translated_message('common_invalid_vat_id')
        }
    },
    success: function (label) {
        // set &nbsp; as text for IE
        label.html("&nbsp;").addClass("checked");
    }
});

});

var countryCodeList = new Array();
<#loop:country_tel_code#>
countryCodeList['<#tag:country_tel_code[].key /#>'] = '<#tag:country_tel_code[].value /#>';
</loop:country_tel_code#>

</script>

```

To

```

<script language="javascript">
$(document).ready(function () {
    $.validator.addMethod("alphanumeric", function (value, element) {
        return /[a-z]+/i.test(value) && /[0-9]+/i.test(value);
    }
    , "<t>Letters or numbers only please</t>");

    $.validator.addMethod("validcountrycode", function (value, element) {
        return validCountryCode(value);
    }
    , "<t>Please enter a valid country code.</t>");

    $.validator.addMethod("validcity", function (value, element) {
        return validCity(value);
    }
    , "<t>Please enter a valid city.</t>");

```

```

    }
    , "<t>Please enter a valid city.</t>");
$.validator.addMethod("signup_zipcode_handler", function (value, element) {
    validator = this;
    validZip(value, $("#country").val(), function (is_valid) {
        console.log(is_valid);
        if (is_valid == 'NA') {
            $("#input_zip").attr('readonly', true);
            $("#input_zip").val('N/A');
            submitted = validator.formSubmitted;
            validator.resetInternals();
            validator.toHide = validator.errorsFor( element );
            validator.formSubmitted = submitted;
            validator.successList.push( element );
            validator.invalid[ element.name ] = false;
            validator.showErrors();
        }
        else {

            if (is_valid) {
                submitted = validator.formSubmitted;
                validator.resetInternals();
                validator.toHide = validator.errorsFor( element );
                validator.formSubmitted = submitted;
                validator.successList.push( element );
                validator.invalid[ element.name ] = false;
                validator.showErrors();
            }
            else {

                message = validator.defaultMessage(element, {method: 'remote', parameters: value});
                errors = {};
                errors[element.name] = message;
                validator.invalid[element.name] = true;
                validator.showErrors(errors);
            }
        }
    });
    return "pending";
}, get_translated_message('contacts_invalidzip'));

// validate signup form on keyup and submit
$("#sign").validate({
    rules: {
        name: "required",
        address1: "required",
        city:
            {
                required: true,
                validcity: true
            },
        telno:
            {
                required: true,
                validcountrycode: true
            },
        telno:
            {
                required: true,
                number: true,
                minlength: 6
            },
        mobileno:

```

```

        {
            required: false,
            number: true,
            minlength: 6,
            maxlength: 12
        },
        conf_passwd:
        {
            required: true,
            minlength: 9,
            equalTo: "#passwd",
            alphanumeric: true
        },

        email: {
            required: true,
            email: true
        },
        vat_id: {
            remote: {
                url: $.url("/misc/eu_country.php"),
                type: "GET",
                cache: false,
                dataType: "json",
                data: {
                    vat_id: function () {
                        return $("#input_vat").val();
                    },
                    country_id: function () {
                        return $("#country").val();
                    }
                }
            }
        },
        zip: {
            required: true,
            signup_zipcode_handler: true
        }
    },
    messages: {
        name: "<t>Please specify your name</t>",
        address1: "<t>Please specify your address</t>",
        zip: "<t>Please specify your zip/postcode</t>",
        telno: {
            required: "<t>Please provide us your contact number</t>"
        },
        mobileno: {
            required: "<t>Please provide us your mobile number</t>"
        },
        email: {
            required: "<t>You need a valid email address to log into the website</t>",
            email: "<t>Your email address must be in the format name@domain.com</t>"
        },
        conf_passwd: {
            required: "<t>Please confirm your password</t>",
            minlength: "<t>Please enter the same password as above</t>",
            equalTo: "<t>Please enter the same password as above</t>",
            alphanumeric: "<t>Please enter the same password as above</t>"
        },
        vat_id: {
            remote: get_translated_message('common_invalid_vat_id')
        }
    }

```

```

    },
    success: function (label) {
        // set &nbsp; as text for IE
        label.html("&nbsp;").addClass("checked");
    }
});

$("#sign").click(function(){
    runStrongPassword($("#passwd").val());
});

$("#passwd").on ('input', function (e) {
    runStrongPassword($(this).val());
});

});

var countryCodeList = new Array();
<#loop:country_tel_code#>
countryCodeList['<#tag:country_tel_code[].key /#>'] = '<#tag:country_tel_code[].value /#>';
<#/loop:country_tel_code#>
</script>

```

=====

ui/supersite/en/templates/misc/login/login.html

Line 20

From:

```
var regExp = new RegExp("(?=.{8,15}$)(?=.*\\d)(?=.*[a-zA-Z])(?!.*\\s).*$");

var regExp = new RegExp("(^[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[~!@#$%_+.?:{}],)[A-Za-z0-9~!@#$%_+.?:{}],{9,16}$");
```

=====

Add before </script> on line 45:

```
function runStrongPassword(strPassword) {

    //console.log(strPassword);
    var specialCharCount = strPassword.match(/[~!@#$%_+.?:{}],/g);
```

```

var capCount = strPassword.match(/[A-Z]/g);
var lowerCount = strPassword.match(/[a-z]/g);
var digitCount = strPassword.match(/\d/g);
var invalidCharacters = strPassword.match(/[&()'\-=[\];""\|<>]/g);
//console.log(invalidCharacters != null && invalidCharacters.length > 0);

passwordLengthCheck(strPassword);
specialCharacterCheck(specialCharCount);
upperCaseCheck(capCount);
lowerCaseCheck(lowerCount);
digitCheck(digitCount);
invalidCharacterCheck(invalidCharacters);

if(passwordLengthCheck(strPassword) && specialCharacterCheck(specialCharCount) &&
upperCaseCheck(capCount) && lowerCaseCheck(lowerCount) && digitCheck(digitCount))
    return true;
else
    return false;
}
function passwordLengthCheck(strPassword) {
    if (strPassword.length == 0) {
        setTextColorToRed('pwdlength');
        setTextColorToRed('specialchar');
        setTextColorToRed('upperchar');
        setTextColorToRed('lowerchar');
        setTextColorToRed('digit');
    }
    if (strPassword.length >= 9 && strPassword.length <= 16) {
        setTextColorToGreen('pwdlength');
        return true;
    }
    if (strPassword.length > 16 || strPassword.length < 9) {
        setTextColorToRed('pwdlength');
    }
}
function specialCharacterCheck(specialCharCount) {
    if (specialCharCount != null && specialCharCount.length >= 1) {
        setTextColorToGreen('specialchar');
        return true;
    }
    else {
        setTextColorToRed('specialchar');
    }
}
function upperCaseCheck(capCount) {
    if (capCount != null && capCount.length >= 1) {
        setTextColorToGreen('upperchar');
        return true;
    }
    else {
        setTextColorToRed('upperchar');
    }
}
function lowerCaseCheck(lowerCount) {
    if (lowerCount != null && lowerCount.length >= 1) {
        setTextColorToGreen('lowerchar');
        return true;
    }
    else {
        setTextColorToRed('lowerchar');
    }
}

```

```
}  
function digitCheck(digitCount) {  
  if (digitCount != null && digitCount.length >= 1) {  
    setTextColorToGreen('digit');  
    return true;  
  }  
  else {  
    setTextColorToRed('digit');  
  }  
}  
function invalidCharacterCheck(invalidCharacters) {  
  if (invalidCharacters != null && invalidCharacters.length > 0) {  
    setTextColorToRed('specialchar');  
  }  
}  
  
function setTextColorToRed(ele){  
  document.getElementById(ele).className = "invalid";  
}  
function setTextColorToGreen(ele){  
  document.getElementById(ele).className = "valid";  
}
```

=====

[ui/partnersite/en/templates/misc/signup/signup_form.html](#)

Line 13

From

```
var regExp = new RegExp("(?=^.{8,15}$)(?=.*\\d)(?=.*[a-zA-Z])(?!.*\\s).*$");
```

To

```
var regExp = new RegExp("(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[~!@#$%_+.?:{}]) [A-Za-z0-9~!*!@#$%_+.?:{}]{9,16}$");
```

Line 195

From

<t>(8-15 alphanumeric characters)</t>

To

<t>The password should be 9 to 16 characters long and should have at least 1 lower case (a-z), 1 upper case (A-Z), 1 number (0-9) and 1 special character (~!*!@#\$%_+.?:, { })</t>

=====