Design & Styling

- Email templates
  - Inserting data
  - System templates
  - Wrapping emails
- Grouping fields
  - Pages
- Icons/Buttons
- Media files
- Stylesheets
  - Styling cheatsheet
  - Using Google fonts
  - What are Stylesheets
- Tricks and hacks
  - CSS for form buttons in page footer
  - CSS for printing labels
  - Inline dashboard
  - JQuery / script cheatsheet
  - Many pages / long page name
  - Rounded corners missing in simple menu mode
- Updating to design version 5

Email templates

If you need to include graphics in the design we recommend using inline base64 images.

CSS is easily embeddable inside style tags

Inserting data

To insert data from a record,,into an email, the following syntax is used.

`{FIELDNAME}`

Custom functions

A couple of functions are implemented, allowing minor modification of data from fields, before inserting it.

Syntax: `@[FUNCTION]({FIELDNAME} [PARAMETER])`

Function: FALLBACK

Sample: `@FALLBACK({NAME} Friend)`

If the given field does not have a value, or is not found, the PARAMETER value will be used instead.

Function: DATEADD

Sample: `@DATEADD({DATE} 3 weeks)`

If the given field is a Date, Time or DateTime field, the value is modified by the given amount (positive or negative)

Supports wrapping inside FORMAT (see below).

Supported periods: minute(s), hour(s), day(s), week(s), month(s), year(s)

Function: DATESUB

Sample: `@DATESUB({DATE} 3 weeks)`

If the given field is a Date, Time or DateTime field, the value is modified by the given amount (positive or negative)

Supports wrapping inside FORMAT (see below).

Supported periods: minute(s), hour(s), day(s), week(s), month(s), year(s)

Function: FORMAT

Sample: `@FORMAT({NUMBER} 000.000)` or `@FORMAT({DATE} MMM, d YYYY)`

Formats a given number or date to the specified format.

Number formatting uses DecimalFormat and date formatting uses SimpleDateFormat.

It is possible to wrap this function around DATESUB and DATEADD like this:
`@FORMAT(@DATEADD({DATE} 3 weeks) MMM, d YYYY)`

System templates

Templates can be changed in the designer: **Modules > Configuration**

Invitation template

HTML can be edited in **Template.WelcomeUser** and **Template.WelcomeUserLink**

Supported tags

- {APPLICATION}
- {LOGINURL}
- {USERNAME}
- {PASSWORD}

Password reset template

HTML can be edited in **Template.PasswordReset** and **Template.PasswordResetLink**

Supported tags

- {LOGINURL}
- {PASSWORD}

Reset via link-templates

Additional supported tags

- {PASSWORDLINK}
- {LINKLIFETIME}

Wrapping emails

It is possible to wrap all emails sent by the system in an MJML or HTML wrapper.

Our implementation of the MJML engine is a bit unstable, so it is recommended to build the wrapper using MJML, then exporting the HTML version and uploading that to the platform.

Two tags are available: `[SUBJECT]` and `[BODY]`

SUBJECT will be replaced with the subject of the given email.

BODY will be replaced with the email-content of the given email.

To enable email-wrapping, set the Policy `defaultEmailTemplateID` to the ID of the template.

You can control whether the content of the email is inserted into the template before or after rendering, via the Policy `emailDoInsertBeforeRender`.

Grouping fields

Pages

Pages are grouping for fields, that will display corresponding tabs in record forms. Selecting the tab will filter out the fields not belonging to the page in question.

Example pages.PNG

In questionaires the pages will be displayed as an explanatory subject.

Icons/Buttons

Buttons are graphical elements that are tied to functions in Tempus Serva.

Media files

Media files allows upload of files for use in the solutions

The files are publicly available / does not requires users to log in.

Typical uses:

- Pictures for wrappers or stylesheets
- User manuals for solutions
- Templates for use record file storages

New files are uploaded using: "Ressources" > "Media files" > "Add"

Stylesheets

Styling cheatsheet

Shared page styling

All pages are structured in the following CSS class structure

- .TempusServaPage (root)
  - .menuList
    - .menuItem
  - ... content ...

Content can easily be wrapped around the "TempusServaPage" by use of wrappers

Command bases styling

Menu mode

Simple / Standard

Advanced / Accordion

List mode

- .TempusServaPage
  - .tableListHeader
  - .viewList
    - .tableList
      - tr
        - td (field names)
      - **tr [repeat]**
        - td (field values)
    - .pager
      - .pagerLink

        - .pagerPagesize

type	class	content
tr	customStyle_[item status name]	Container for a single record line

Variant views

Most list related operations will have a structure like this.

- .TempusServaPage
  - .tableListHeader
  - **[.viewHeat | .tableStatistics | ...]**
    - .tableList

Note the inner table "tableList", that is present in all list operations. Specific dialogue styling should use the outer tag - example: "tableStatistics".

Item mode

- .TempusServaPage
  - *.tableForm*
    - .PAGE_[field PageID]
      - .fieldLabel
      - .fieldValue
      - .fieldNotes

Allthough the ".tableForm" is persent in the default template, there is no guarantee taht the element can be found within the page.

type	class	id	content	template
tbody	PAGE_[field PageID]		Container for a single field	{default_start} / {default_end}
div	FieldLabel	NB_[system fieldname]	Label for field	{default_name}
div	FieldValue	VB_[system fieldname]	Value of field / Input for field	{default_value}
div	FieldNotes	HB_[system fieldname]	Optional help text for the field	{default_help}

Further information on: Form templates

Other element styling

ITEM page selector

The page selector may be placed anywhere within the template

- formpageselector
  - tablePageSelector
    - <node>
      - tablePageSelectorElementLeftActive
      - tablePageSelectorElementActive
      - tablePageSelectorElementRightActive
    - <node>
      - tablePageSelectorElementLeft
      - tablePageSelectorElement
      - tablePageSelectorElementRight

Targeting mobile devices

The following example shows a **script** that dynamically assigns a red background to the menu, but only while using mobile devices.

```
if (/Android|webOS|iPhone|iPad|iPod|BlackBerry/i.test(navigator.userAgent)) {
  $('.menuList').css('background','red');
}
```

Hiding form update elements

As of build 2345 there is no easy way to refer ALL the update elements (some browsers are not ble to collapse cells with hidden contents).

```
<script>
  $('#NB_DATA_StatusID').parent().parent().hide();
  $('#NB_DATA_Revision').parent().parent().hide();
  $('.updateSubmit').parent().parent().parent().hide();
</script>
```

Using Google fonts

Find a font you alike

http://www.google.com/webfonts

Click "Quick use" on the font you want to use and follow the directions.

Change your stylesheet like this example

```
...
@import url(http://fonts.googleapis.com/css?family=Molle:400italic);
...
.TempusServaPage h3 { font-family: 'Molle'; }
...
```

The import method is needed, but embedding by a link inside a wrapper is also possible.

What are Stylesheets

Stylesheets can be applied in three ways

- Attached to solution
- Attached to solution interface
- Default application stylesheet

Stylesheets are defined in: Designer > Ressources > Stylesheet

Stylesheets contain CCS code to markup the page elements. The stylesheets may inherit content from each other, which is spcecified by the *Parent stylesheet*. Parent stylesheets are automatically preprended to the stylesheet in question, so definitions may be overloaded if needed.

A good ressources for learning CSS can be found here: http://www.w3schools.com/css/

The default stylesheet for the application is defined in the policy: **defaultStylesheetID**

Tricks and hacks

Tricks and hacks

CSS for form buttons in page footer

```
.tsUpdateButtons {
  position: fixed;
  bottom: 0px;
  width: 92%;
  max-width: 1200px;
  padding: 0px 0px 2px 0px;
  z-index: 26;
  background-color: transparent;
  opacity: 1;
}
.submitOption, .updateSubmit {
  max-width: 22%;
  line-height: 1;
  height: 26px;
}
```

Tricks and hacks

CSS for printing labels

The below case will style a certain dashboard widget outputting LI elements

```
@page {
  margin: 0cm; /* page reset */
}
@media print {
  html * { margin: 0px; padding: 0px; } /* page reset */
  #Widget23 ul { padding: 12mm 6mm 12mm 6mm; margin-bottom: 0; }
  #Widget23 li { border: 0px; padding: 5mm; height: 33.9mm; width: 63.5mm; margin-right: 2mm; }
  #Widget23 li { border: 1px solid #EEEEEE; }   /* disable after print is validated */
}
```

The page setup is

- left margin 6mm
- top margin 12mm

The label layout is

- size 63.5mm x 33.9mm
- spacing 2mm

Inline dashboard

In order to display a dashboard on a page for instance on an item add the following code to the custom script

```
$('.mainContent form').append('<p class=inlineDashboard></p>');
$('.inlineDashboard').load('main?command=board&Dashboard=4&HideLinks=1');
```

JQuery / script cheatsheet

Navigation

Redirecting non administrators away from list views

```
var params =  window.location.href.split("?")[1];
if( params == "SagID=251&command=list" ) {
  if( ! $( "#TempusServaPage" ).hasClass( "IsAdministrator" ) ) {
    console.log("redirecting to main from list");
    window.location.href = "main";
  }
}
```

Prevent edit as default command from lists

```
$(".tableList a").each(function() {
  url = $(this).attr("href").replace("=edit","=show");
  $(this).attr("href",url);
});
```

Files

Make all file links WebDAV enabled

```
$().ready( function() {
  $(".webdavFile").each( function() {
    $(this).next().attr("href", $(this).attr("href") );
  });
});
```

Input manipulation

Building multiselect values from existing services

In the following example the field TERRITORY has autocomplete wuth Country values. Output format example: "Denmark, Finland, Sweden"

```
function split( val ) {
  return val.split( /,\s*/ );
}
function extractLast( term ) {
  return split( term ).pop();
}
$(function() {
  $("#DATA_TERRITORY").autocomplete({
       source: function( request, response ) {
         $.getJSON( "autocomplete?type=dk.p2e.blanket.form.fields.ajax.FieldAjaxCountry&subtype=0", {
           term: extractLast( request.term )
         }, response );
       },
      autoFocus: true,
      min_length: 2,
      delay: 300,
       search: function() {
         // custom minLength
         var term = extractLast( this.value );
         if ( term.length < 2 ) {
           return false;
         }
       },
       focus: function() {
         // prevent value inserted on focus
         return false;
       },
       select: function( event, ui ) {
         var terms = split( this.value );
         // remove the current input
         terms.pop();
         // add the selected item
         terms.push( ui.item.value );
         // add placeholder to get the comma-and-space at the end
         terms.push( "" );
```

```
        this.value = terms.join( ", " );
        return false;
      }
  });
});
```

Making radiobuttons unselectable

Clicking on a radiobutton a second type will remove the selection

```
$("input[type=radio]").click( function() {
 if( $(this).attr("checked") == 'checked' )
    $(this).removeAttr("checked").button("refresh");
else
    $(this).attr("checked",true).button("refresh");
});
```

Transform text input to selectbox

Include the following function

```
function transformToSelectField( fieldName, valueList) {
  var select = "<select class='form-control' id='DATA_" + fieldName + "' name='DATA_" + fieldName + "'
tabindex='2'>";
  var optionsList = valueList.split(" ");
  for (var i = 0; i < optionsList.length; i++) {
   var isSelected = ( getValue(fieldName) == optionsList[i] ) ? " selected " : "";
   select += "<option " + isSelected + " value='" + optionsList[i] + "'>" + optionsList[i] + "</option>";
  }
  select += "</select>";
  console.log( select );
  $("#DATA_" + fieldName).replaceWith( select );
}
```

Example usage

```
transformToSelectField( "NAME", "Alice Bob" );
```

Copy / paste

Selecting text for easy copying

```
//Get Exising Select Options
$('form#product select').each(function(i, select){
  var $select = $(select);
  $select.find('option').each(function(j, option){
    var $option = $(option);
    // Create a radio:
    var $radio = $('<input type="radio" />');
    // Set name and value:
    $radio.attr('name', $select.attr('name')).attr('value', $option.val());
    // Set checked if the option was selected
    if ($option.attr('selected')) $radio.attr('checked', 'checked');
    // Insert radio before select box:
    $select.before($radio);
    // Insert a label:
    $select.before(
      $("<label />").attr('for', $select.attr('name')).text($option.text())
    );
    // Insert a
:
    $select.before("
");
  });
  $select.remove();
});
```

credit

Selecting text for easy copying

```
var textNode = document.getElementById('IdOfNodeToBeSelected');
if (document.selection) {
  var range = document.body.createTextRange();
  range.moveToElementText(textNode);
  range.select();
} else if (window.getSelection) {
  var selection = window.getSelection();
  var range = document.createRange();
  range.selectNodeContents(textNode);
```

```
  selection.removeAllRanges();
  selection.addRange(range);
}
```

Many pages / long page name

The following code will shorten page selectors, unless they are hovered or active

```
.tablePageSelector td a {
   max-width: 80px;
   overflow: hidden;
   text-overflow: ellipsis;
   font-size: 80%;
}
.tablePageSelector .tablePageSelectorElementActive a,
.tablePageSelector td a:hover {
  max-width: none;
}
.tablePageSelectorElementLeft, .tablePageSelectorElementRight { width: 2px; }
```

Rounded corners missing in simple menu mode

When a user is running in the context of a simple user certain buttons will be hidden, and on displayed in the expanded menu ("burger"). Previous buttons might have their corners correctly, which can be mitigated by removing the items (they are normally just hidden with CSS).

```
$('.menuSimple .menuPlus').remove();
```

Note: Removing the elements will also remove them from the burger menu.

Updating to design version 5

Step by step guide

The following steps are required to update your design

1. Activate version 5 layout + bootstrap 4
    1. Set configuration > layoutNewestVersion = true
2. Update stylesheet
    1. Copy new stylesheet (see below)
    2. Change color codes in new sheet
    3. Set configuration > defaultStylesheet = <ID>
3. Update wrapper
    1. Copy new wrapper (see below)
    2. Set configuration > defaultWrapper = <ID>

Additionally you might want to update how Dashboards work too

1. Set configuration > dashboardMain = true
2. Change widget layouts (for each)
    1. Set width to 1/12 parts (6 = half, 12 = full)

Sample stylesheet

```
:root {
  --themePrimary: #007bff;
  --themePrimaryLight: #cbe4ff;
  --themePrimaryDark: #312783;
  --themePrimarySpecial: #648bb7;
  --themeSecondary:#8c2db3;
  --themeTeritary: #8e9aa7;
  --themeDark: #333333;
  --themeGreyDark: #5a626b;
  --themeGrey: #ced4da;
  --themeLightGrey: #f8f9fc;
  --themeLight: #ffffff;
}
```

```
.logo {
  background: url(https://alpha.tempusserva.dk/TempusServa/media/korsbaek-white.13.svg) no-repeat
!important;
  background-position-x: left;
  background-position-y: center;
  background-size: contain !important;
  opacity:0.6;
  width: 300px;
  margin-top: 8px;
  height: 44px;
}
```

Sample wrapper

```
<html>
  <head>
    <link rel="icon" type="image/png" href="https://www.tempusserva.dk/favicon.ico" />
    <meta name="application-name" content="Tempus Serva" />
    <meta name='viewport' content='initial-scale=1.0, maximum-scale=1.0'>
    <meta name="mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-capable" content="yes">
  </head>
  <body>
```

```
  </body>
</html>
```