



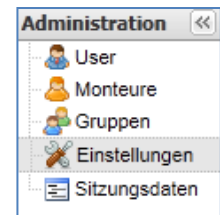
# Spezifische Anpassungen

Katalog zum eigenständigen Adaptieren von  
levatis Personalplanung für Bau und Handwerk





## Allgemein

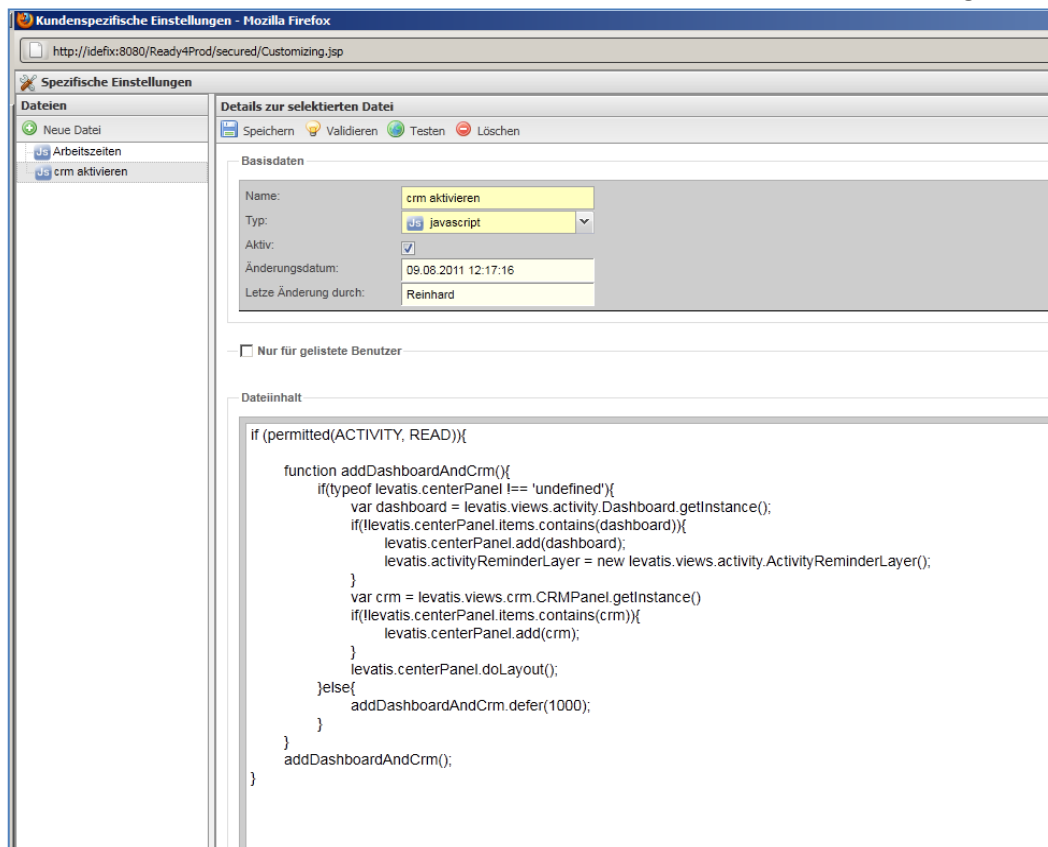
Es gibt umfangreiche Einstellungsmöglichkeiten, um sich levatis nach seinen individuellen Bedürfnissen anzupassen. Dazu gibt es im rechten Administrationsbereich den Menüpunkt „Einstellungen“.



Es öffnet sich ein eigenes Fenster – die kundenspezifischen Einstellungen. Hier können alle individuellen Einstellungen selbst konfiguriert werden. Es können beliebig viele Dateien erstellt werden. Dabei handelt es sich um sogenannte javascript (.cs) oder cascading stylesheets (.css). Das spielt aber keine große Rolle, denn mit Hilfe unseren Vorlagen müssen sie lediglich den Text in das entsprechende Feld kopieren, das war es dann auch schon.

Folgende Schritte müssen der Reihe nach ausgeführt werden:

1. Im linken Bereich → Neue Datei
2. Bei den Basisdaten → Der Datei einen Namen geben
3. Bei den Basisdaten → Typ entsprechend Katalog auswählen:  oder 
4. Bei den Basisdaten → Datei auf aktiv setzen
5. Wenn gewünscht können die Änderungen nur für einzelne User freigeschaltet werden. In diesem Fall müssen die entsprechenden User ausgewählt werden
6. Die gewünschten Einstellungen aus dem Katalog kopieren und in den Bereich „Dateiinhalte“ kopieren
7. Auf „Validieren“ klicken
8. Wenn die Validierung positiv ist, kann mittels „Testen“ die Änderung in einem eigenen Fenster kontrolliert werden
9. Ist das Ergebnis zufriedenstellend, so bitte mittels „Speichern“ die Einstellungen speichern.
10. Die Änderungen werden im eigenen Fenster erst nach einem erneuten Laden im Browser sichtbar. Wenn Sie diese sofort haben möchten, müssen Sie im Wochenkalender einmal Strg+F5 drücken.



```

if (permitted(ACTIVITY, READ)){
    function addDashboardAndCrm(){
        if(typeof levatis.centerPanel !== 'undefined'){
            var dashboard = levatis.views.activity.Dashboard.getInstance();
            if(!levatis.centerPanel.items.contains(dashboard)){
                levatis.centerPanel.add(dashboard);
                levatis.activityReminderLayer = new levatis.views.activity.ActivityReminderLayer();
            }
            var crm = levatis.views.crm.CRMPanel.getInstance()
            if(levatis.centerPanel.items.contains(crm)){
                levatis.centerPanel.add(crm);
            }
            levatis.centerPanel.doLayout();
        }else{
            addDashboardAndCrm.defer(1000);
        }
    }
    addDashboardAndCrm();
}
    
```

## Grundsätzliche Einstellungen

### *Arbeitszeiten pro Tag*

Pro Wochentag ist eine Zahl einzugeben, maximal 2 Nachkommastellen mit Beistrichen getrennt. [Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag, evtl. Sonntag].

```
levatis.defaults.workloads = [ 4.0, 4.0, 8.5, 8.5, 8.5, 8.5, 8.5 ];
```

### *Abrunden der Tagesarbeitszeit (Anzeige der Auslastung in den Farben)*

Mit dieser Prozedur werden die Auslastungen abgerundet, d.h. Orange wird erst mit dem vollen Tagesarbeitspensum angezeigt (standardmäßig wird kaufmännisch gerundet).

```
function getWorkloadCSS(wl, weekDay) {
    var defaultWl = levatis.defaults.workloads[weekDay];
    var rel = (wl / defaultWl);
    var wlId = 8 * rel;
    if (wlId < 1 && wlId > 0) {
        return 'workload1';
    }
    wlId = Math.floor(wlId);
    if (wl > defaultWl) {
        return 'workload9';
    }
    return 'workload' + Math.floor(wlId);
}
```

### *Sonntag als Arbeitstag hinzunehmen*

Mit diesem Befehl kann der Sonntag als zusätzlicher Tag im Wochenkalender aktiviert werden. Zu beachten ist, dass die jeweiligen Arbeitszeiten pro Tag ebenso als siebenter Tag hinterlegt ist.

```
levatis.defaults.sundayAsWorkday = true;
```

### *Aufgabenfenster: zuschalten von Feldern*

Mit Hilfe des Befehls können sowohl Aufgaben-, als auch die Auftragsnummer hinzugefügt werden.

```
levatis.defaults.taskwindow = {
    USE_FF_INTERNALID : true ,           // Aufgabennummer aus dem
    Warenwirtschaftssystem

    USE_FF_JOBNUMBER : true ,           // Auftragsnummer aus dem
    Warenwirtschaftssystem

    USE_FF_JOBNUMBER_LOCK : true,       // true bedeutet, dass die
    Auftragsnummer nicht geändert werden kann
}
```

```

        USE_FF_INTERNALID_LOCK : true    // true bedeutet, dass die
Aufgabennummer nicht geändert werden kann
    };

```

### CRM-light hinzufügen

Mit diesem Befehl kann die Funktion des internen Aufgabenstapel hinzugefügt werden. Standardmäßig ist er nicht eingeschaltet. Das CRM kann auf Berechtigungsebene danach auch pro User in den Administrationseinstellungen aktiviert/deaktiviert werden (Administration → User → Tätigkeiten).

```

if (permitted(ACTIVITY, READ)){
    function addDashboardAndCrm(){
        if(typeof levatis.centerPanel !== 'undefined'){
            var dashboard =
levatis.views.activity.Dashboard.getInstance();
            if(!levatis.centerPanel.items.contains(dashboard)){
                levatis.centerPanel.add(dashboard);
            }
            var crm = levatis.views.crm.CRMPanel.getInstance()
            if(!levatis.centerPanel.items.contains(crm)){
                levatis.centerPanel.add(crm);
            }
            levatis.centerPanel.doLayout();
        }else{
            addDashboardAndCrm.defer(1000);
        }
    }
    addDashboardAndCrm();
}

```

### Individuelle Ablagen hinzufügen

Standardmäßig sind die *Allgemeine* und die *Private Ablage* aktiviert. Die *Allgemeine Ablage* kann von allen Usern gesehen und benutzt werden, die *Private Ablage* nur vom jeweiligen User. Werden diese Zeilen kopiert, so müssen stets die obersten zwei („Allgemein“ u. „Private Ablage“) gleich bleiben. Im Folgenden können beliebig viele Ablagen hinzugefügt werden. Im Feld „style“ kann die Hintergrundfarbe der jeweiligen Ablage gewählt werden. Dazu muss die Farbe als HTML-Hexcode eingegeben werden.

```

levatis.wv.clipboardDrawersDef = [{
    id : 'public',
    title : 'Allgemein',
    style : 'background:#FFFFFF;'
}, {
    id : levatis.portaluser.login,
    title : 'Private Ablage',
    style : 'background:#F6E3CE;'
}, {

```

```

        id : 'ablagedrei',
        title : 'Ablage 3',
        style : 'background:#00FF00;'
    }, {
        id : 'ablagevier',
        title : 'Ablage 4',
        style : 'background:#2ECCFA;'
    }, {
        id : 'ablagefuenf',
        title : 'Ablage 5',
        style : 'background:#FF0000;'
    }
    ];

```



### *Sonntag als Arbeitstag hinzunehmen*

Mit diesem Befehl kann der Sonntag als zusätzlicher Tag im Wochenkalender aktiviert werden. Zu beachten ist, dass die jeweiligen Arbeitszeiten pro Tag ebenso als siebenter Tag hinterlegt ist.

```
levatis.defaults.sundayAsWorkday = false;
```



### *Der Tag beginnt um wieviel Uhr?*

Wann der Tag beginnen soll, bestimmt folgende Syntax. Die Zahl sind Minuten nach Mitternacht. Standardmäßig beginnt der Tag um 7:00.

```
levatis.defaults.dayStartsAt = 540;
```



### *Zusätzliche Sonderzeiten*

Sollen andere Sonderzeiten gelten, muss folgender Befehl eingegeben werden. Es werden damit alle bisherigen Sonderzeiten überspielt, d.h. möchte man nur eine zusätzliche Sonderzeit, müssen die bestehenden Sonderzeiten ebenso hinzugefügt werden. Es gibt eine Vielzahl an Icons, bitte wenden Sie sich an uns, wir helfen Ihnen gerne bei der Auswahl.

```

levatis.defaults.addExtraTasksMenuItems = function(menu){
    menu.addItem({
        text : 'Urlaub',
        icon : 'images/icons/world.png'
    });
    menu.addItem({
        text : 'Krankenstand',
        icon : 'images/icons/pill.png'
    });
    menu.addItem({
        text : 'Lager',
        icon : 'images/icons/transmit_go.png'
    });
}

```

```

    });
    menu.addItem({
        text : 'Zeitausgleich',
        icon : 'images/icons/time.png'
    });
    menu.addItem({
        text : 'Schulung',
        icon : 'images/icons/calculator_edit.png'
    });
    menu.addItem({
        text : 'Bürotätigkeit',
        icon : 'images/icons/page_white_edit.png'
    });
    menu.addItem({
        text : 'Bereitschaft',
        icon : 'images/icons/status_away.png'
    });
};

```



### *Sonderzeiten im Wochenkalender inklusive Icons*

Möchte man spezifische Sonderzeiten im Wochenkalender auch mit einem entsprechendem Icon angezeigt haben, dann kann mit folgendem Befehl diese anzeigen lassen:

```

.template-sonderzeit-Monteurbesprechung {

    background-image:url(../images/icons/group.png);
    background-repeat: no-repeat;
    background-position:center bottom;
    width :100%;
    height:20px;
}

```



### *Projektmarkierung hinzufügen*

Im Projektdetailfenster gibt es die Möglichkeit ein individuelles Auswahlfeld zu hinterlegen. Dieses nennt sich projectmarker und ist wie folgt in die Specifics einzutragen:

```
levatis.defaults.projectmarker = ['offen','bezahlt','abgeschlossen'];
```



### *Aufgabenmarkierung hinzufügen*

Äquivalent zur Projektmarkierung kann die Aufgabenmarkierung individuelle Auswahlfelder zur Verfügung stellen. Mit folgendem Befehl kann diese aktiviert werden:

```
levatis.defaults.taskmarker=['Markierung1','Markierung2'];
```



### *Einen Report als Standardarbeitsblatt hinterlegen*

Man kann einen spezifischen Report als eigenes Arbeitsblatt hinterlegen. Damit wird das Standard-Arbeitsblatt ignoriert. Die Syntax erfordert zuerst die ID des Berichtes (Beim Support zu erfragen) und mit einem Beistrich getrennt und unter einfachem Hochkomma die Bezeichnung des Arbeitsblattes.

```
levatis.reporting.worksheets = [ [123456, 'Standardarbeitsblatt'] ];
```

Müssen mehrere unterschiedliche Arbeitsblätter zum Druck hinzugefügt werden, so sind diese folgendermaßen zu trennen:

```
levatis.reporting.worksheets =  
[[123456, 'Standardarbeitsblatt'], [987654, 'Kompr.Arbeitsblatt']];
```

Hier gilt es zu beachten, dass nur das erste Arbeitsblatt bei der Druckfunktion des Wochenkalenders verwendet wird (rechte Maustaste auf Termin => drucken).



### *Kundenspezifische Fenstereinstellungen*

Seit der Version 2.9 können individuelle Fenstereinstellungen hinterlegt werden. Als Beispiel wird hier eine Regex hinterlegt. In Grün das Default-JS, danach das Beispiel für die Specifics.

```
levatis.defaults.taskwindow = {
```

```
...
```

```
USE_FF_INTERNALID_LOCK : false,
```

```
onTaskWindowShow : Ext.EmptyFn,
```

```
ProjectMembersCombo : {allowBlank : true }
```

```
};
```

```
levatis.defaults.taskwindow.onTaskWindowShow = function(taskwindow) {  
levatis.task.generalTabLeftColumn.items.items[10].setDisabled(false);  
levatis.task .generalTabLeftColumn.items.items[10].regex = /^[0-9]/;  
if(Ext.isEmpty( levatis.task .id)) // new task  
this.generalTabLeftColumn.items.items[10].setValue("Hello World as  
default Text");  
}
```

## Einstellungen im Wochenkalender – Templates

Die Anzeige des Wochenkalenders kann über sogenannte Templates individualisiert werden. Grundsätzlich können alle Attribute, die übermittelt werden, auch angezeigt werden.

### Standard-Templates

Mit Hilfe dieser Vorlagen kann der Wochenkalender auf einfache Weise individualisiert werden. Einfach die entsprechenden Templates in den Einstellungen kopieren. Die unterschiedlichen Ausprägungen überschneiden sich zum Teil, das ist allerdings beabsichtigt um ein einfaches Kopieren zu ermöglichen.

### *Mit interner Projektnummer, inkl. technischen und kaufmännischen Verantwortlichen*

```

var teTemplate = new Ext.Template(

'<div class="taskentry{taskDone} {timeerror}" active="{active}" id="te{id}" teId="{id}"
order="{indayorder}" taskId="{taskId}" customer="{customer}" workload="{workload}"
fixedtime="{fixedtime}" version="{version}">',

'<table class="teTable" cellspacing="0">',

/*      ' <tr class="{frontcolor} teHeadRow">',

'<td class="teHeadName{taskDone} fixed{fixed}"><div
style="overflow:hidden;">{customer}&nbsp;{firstname}</div></td>',

'<td class="teHeadZip{taskDone}">{zip}</td>',

'<td class="teWorkload{taskDone}" wl="{workload}">{workload}h</td>',

'</tr>', */

'<tr class="{frontcolor} teHeadRow">',

'<td class="fixed{fixed} widthCss{fixed}" width="15px;"></td>',

'<td class="teHeadName{taskDone}" padding="-1px;"><div
style="overflow:hidden;">{customer}&nbsp;{firstname}</div></td>',

'<td class="teHeadZip{taskDone}" style="width:32px;">{zip}</td>',

'<td class="teWorkload{taskDone}" wl="{workload}">{workload}h</td>',

'</tr>',

'<tr>',

'<td colspan="4" class="teHeadZip{taskDone}"
style="color:black;"><b>{projectinternalId}</b></td>', /*<td colspan="2"
class="teContent{taskDone}">{projectinternalId}</td>',

'</tr>',

'<tr>',

'<td colspan="4" class="teContent{taskDone}">{headerInfo}</td>',

'</tr>',

'<tr><td colspan="4" class="teComment">{comment}</td></tr>',

'<tr><td colspan="4" style="text-
align:right;"><b>{tasktechuser_short}|{taskorguser_short}</b><span style="margin-left:15px;"
class="teTime">{formattedTimeStart}{formattedTimeEnd}</span> </td></tr>',

'</table>', '</div>'); /*<td colspan="1" style="text-align:left;"><span
class="teTime"></span></td>

teTemplate.compile();

var teTemplate = new Ext.Template(
    
```



## Mit Projektstraße

```

        '<div class="taskentry{taskDone} {timeerror}" active="{active}" id="te{id}"
        teId="{id}" order="{indayorder}" taskId="{taskId}" customer="{customer}" workload="{workload}"
        fixedtime="{fixedtime}" version="{version}">',
        '<table class="teTable" cellspacing="0">',
        '<tr class="{frontcolor} teHeadRow">',
        '<td class="teHeadName{taskDone} fixed{fixed}"><div
        style="overflow:hidden;">{customer}&nbsp;{firstname}</div></td>',
        '<td class="teHeadZip{taskDone}">{zip}</td>',
        '<td class="teWorkload{taskDone}" wl="{workload}">{workload}h</td>',
        '</tr>',
        '<tr>',
        '<td colspan="3" class="teContent{taskDone}"><b>{street}</b></td>', //'<td
        colspan="2" class="teContent{taskDone}">{projectinternalId}</td>',
        '</tr>',
        '<tr>',
        '<td colspan="3" class="teContent{taskDone}">{headerInfo}</td>',
        '</tr>',
        '<tr><td colspan="3" class="teComment">{comment}</td></tr>',
        '<tr><td colspan="3" style="text-align:right;"><span
        class="teTime">{formattedTimeStart}{formattedTimeEnd}</span> </td></tr>',
        '<tr><td colspan="3" class="teTaskMarker-value-{taskmarker}">&nbsp;</td></tr>',
        '</table>', '</div>'); //'<td colspan="1" style="text-align:left;"><span
        class="teTime"></span></td>
teTemplate.compile();
    
```



## Mit Projektstraße und Projektbezeichnung

```

var teTemplate = new Ext.Template(
    '<div class="taskentry{taskDone} {timeerror}" active="{active}" id="te{id}"
    teId="{id}" order="{indayorder}" taskId="{taskId}" customer="{customer}" workload="{workload}"
    fixedtime="{fixedtime}" version="{version}">',
    '<table class="teTable" cellspacing="0">',
    '<tr class="{frontcolor} teHeadRow">',
    '<td class="teHeadName{taskDone} fixed{fixed}"><div
    style="overflow:hidden;">{customer}&nbsp;{firstname}</div></td>',
    '<td class="teHeadZip{taskDone}">{zip}</td>',
    '<td class="teWorkload{taskDone}" wl="{workload}">{workload}h</td>',
    '</tr>',
    '<tr>',
    '<td colspan="3" class="teContent{taskDone}">{street}</td>', //'<td colspan="2"
    class="teContent{taskDone}">{projectinternalId}</td>',
    '</tr>',
    '<tr>',
    '<td colspan="3" class="teContent{taskDone}"><b>{projectsubject}</b></td>',
    //'<td colspan="2" class="teContent{taskDone}">{projectinternalId}</td>',
    '</tr>',
    '<tr>',
    '</tr>',
    '</table>', '</div>');
    
```

```

        '<td colspan="3" class="teContent{taskDone}">{headerInfo}</td>',
        '</tr>',
        '<tr><td colspan="3" class="teComment">{comment}</td></tr>',
        '<tr><td colspan="3" style="text-align:right;"><span
class="teTime">{formattedTimeStart}{formattedTimeEnd}</span> </td></tr>',
        '</table>', '</div>'); //<td colspan="1" style="text-align:left;"><span
class="teTime"></span></td>
teTemplate.compile();
    
```



### Mit Projektbezeichnung

```

var teTemplate = new Ext.Template(
    '<div class="taskentry{taskDone} {timeerror}" active="{active}" id="te{id}"
teId="{id}" order="{indayorder}" taskId="{taskId}" customer="{customer}" workload="{workload}"
fixedtime="{fixedtime}" version="{version}">',
    '<table class="teTable" cellspacing="0">',
    '<tr class="{frontcolor} teHeadRow">',
    '<td class="teHeadName{taskDone} fixed{fixed}"><div
style="overflow:hidden;">{customer}&nbsp;{firstname}</div></td>',
    '<td class="teHeadZip{taskDone}">{zip}</td>',
    '<td class="teWorkload{taskDone}" wl="{workload}">{workload}h</td>',
    '</tr>',
    '<tr>',
    '<td colspan="3" class="teContent{taskDone}"><b>{projectsubject}</b></td>',
    //<td colspan="2" class="teContent{taskDone}">{projectinternalId}</td>',
    '</tr>',
    '<tr>',
    '<td colspan="3" class="teContent{taskDone}">{headerInfo}</td>',
    '</tr>',
    '<tr><td colspan="3" class="teComment">{comment}</td></tr>',
    '<tr><td colspan="3" style="text-align:right;"><span
class="teTime">{formattedTimeStart}{formattedTimeEnd}</span> </td></tr>',
    '</table>', '</div>'); //<td colspan="1" style="text-align:left;"><span
class="teTime"></span></td>
teTemplate.compile();
    
```



### Mit Projektnummer und Projektbezeichnung

```

var teTemplate = new Ext.Template(
    '<div class="taskentry{taskDone} {timeerror}" active="{active}" id="te{id}"
teId="{id}" order="{indayorder}" taskId="{taskId}" customer="{customer}" workload="{workload}"
fixedtime="{fixedtime}" version="{version}">',
    '<table class="teTable" cellspacing="0">',
    '<tr class="{frontcolor} teHeadRow">',
    '<td class="teHeadName{taskDone} fixed{fixed}"><div
style="overflow:hidden;">{customer}&nbsp;{firstname}</div></td>',
    '<td class="teHeadZip{taskDone}">{zip}</td>',
    '<td class="teWorkload{taskDone}" wl="{workload}">{workload}h</td>',
    '</tr>',
    '<tr>',
    '<td colspan="3" class="teContent{taskDone}"><b>{projectsubject}</b></td>',
    //<td colspan="2" class="teContent{taskDone}">{projectinternalId}</td>',
    '</tr>',
    '<tr>',
    '<td colspan="3" class="teContent{taskDone}">{headerInfo}</td>',
    '</tr>',
    '<tr><td colspan="3" class="teComment">{comment}</td></tr>',
    '<tr><td colspan="3" style="text-align:right;"><span
class="teTime">{formattedTimeStart}{formattedTimeEnd}</span> </td></tr>',
    '</table>', '</div>'); //<td colspan="1" style="text-align:left;"><span
class="teTime"></span></td>
teTemplate.compile();
    
```

```

        '<td colspan="3" class="teHeadZip{taskDone}" style="color:black;font-weight:
lighter;"><b>{projectinternalId}</b></td>', // '<td colspan="2"
class="teContent{taskDone}">{projectinternalId}</td>',

        '</tr>',

        '<tr>',

        '<td colspan="3" class="teHeadZip{taskDone}"
style="color:black;"><b>{projectssubject}</b></td>', // '<td colspan="2"
class="teContent{taskDone}">{projectinternalId}</td>',

        '</tr>',

        '<tr>',

        '<td colspan="3" class="teContent{taskDone}">{headerInfo}</td>',

        '</tr>',

        '<tr><td colspan="3" class="teComment">{comment}</td></tr>',

        '<tr><td colspan="3" style="text-align:right;"><span
class="teTime">{formattedTimeStart}{formattedTimeEnd}</span> </td></tr>',

        '</table>', '</div>'); // '<td colspan="1" style="text-align:left;"><span
class="teTime"></span></td>

teTemplate.compile();
    
```



### Mit Projektstraße und Auftragsnummer

```

var teTemplate = new Ext.Template(

        '<div class="taskentry{taskDone}" active="{active}" id="te{id}"
teId="{id}" order="{indayorder}" taskId="{taskId}" customer="{customer}" workload="{workload}"
version="{version}" fixedtime="{fixedtime}">',

        '<table class="teTable" cellspacing="0">',

        '<tr class="{frontcolor} teHeadRow">',

        '<td class="teHeadName{taskDone} fixed{fixed}"><div
style="overflow:hidden;">{customer}</div></td>',

        '<td class="teHeadZip{taskDone}">{zip}</td>',

        '<td class="teWorkload{taskDone} {tasktype}"
wl="{workload}">{workload}h</td>',

        '</tr>',

        '<tr>',

        '<td colspan="3"
class="teContent{taskDone}"><b>{street}</b></td>',

        '</tr>',

        '<tr>',

        '<td colspan="3"
class="teContent{taskDone}"><b>{jobnumber}</b></td>',

        '</tr>',

        '<tr>',

        '<td colspan="3" class="teContent{taskDone}">{headerInfo}</td>',

        '</tr>',

        '<tr><td colspan="3" class="teComment">{comment}</td></tr>',

        '<tr><td colspan="3" style="text-align:right;"><span
class="teTime {timeerror}">{formattedTimeStart}{formattedTimeEnd}</span> </td></tr>',

        '<tr><td colspan="3" class="teTaskMarker-value-
{taskmarker}">&nbsp;</td></tr>',

        '</table>', '</div>');
    
```

```
teTemplate.compile();
```

## Spezialitäten

Es gibt Ideen, die über die normale Funktionalität des Wochenkalenders hinausgeht. Diese Extravaganzen werden in diesem Abschnitt behandelt.

### *Zusammenfassung aller Mitarbeiter auf einen Mitarbeiter*

Mit diesem Zusatz werden alle Aufgaben eines Tages auf einen Mitarbeiter zusammengefasst. Dies macht dann Sinn, wenn man eine große Anzahl von Mitarbeitern hat, aber nur wenige Aufgaben pro Tag. Es sind zwei Dateien einzufügen, eine .js und eine .css. Die jeweilige ID der Mitarbeiter ist in der Zeile mit Stern zu setzen!



```
Ext.ns('levatis.vw.resources');

levatis.vw.resources = {

    isActive : true,

    stackItems: true,

    resourceToWorkerHTML_ID : 'teUseResource-workername', // used in OWVTemplates
    workerToResourceHTML_ID : 'cell_2427_',
    resourceempty_css : "teUseResource-isempty",
    defaults_css : "te-resources-default",

    isResource : function(worker){
        return worker.name.match('Zusammenfassung') != null;
    },

    renderResourceToWorker : function(resources){
        var values;

        if(!levatis.vw.resources.stackItems)
            values = new Array();
        else
            values = "";

        for(var i = 0; i < resources.length;i++){
            var val = resources[i].name.substring(2,resources[i].name.length)
            if(levatis.vw.resources.stackItems)
                values += ('<p>'+val.replace(/,/g,"")+ '</p>');
            else
                values.push(' '+val);
        }

        return '<div class="'+levatis.vw.resources.defaults_css+'"'+values+'</div>';
    },

    renderWorkerToResource : function(workers){
        var values;

        if(!levatis.vw.resources.stackItems)
            values = [];
        else
            values = "";

        for(var i = 0; i < workers.length;i++){
```



```

if(levatis.vw.resources.stackItems)
values += ('<p>'+workers[i].name.replace(/,/g,"")+ '</p>');
else
values.push(' '+workers[i].name);
}
return values.length === 0 ? '<div class="'+levatis.vw.resources.resourceempty_css+'"><p>Kein
MA zugewiesen!</p></div>' : '<div
class="'+levatis.vw.resources.defaults_css+'">'+values+'</div>';
},
render : function() {
window.setTimeout(' levatis.vw.resources.privateRender()',1000, levatis.vw.resources);
},
privateRender : function(){
if(!levatis.vw.resources.isActive)return;
var rec = levatis.visualWorkloadWindow.storeTaskEntries.data.items;
//flush divs
for(var i = 0; i < rec.length;i++){
var ti= rec[i].data.taskId ;
var di = rec[i].data.dateId;
var el = document.getElementById(ti+'_'+di);
if(el != null)
el.remove();
}
for(var i = 0; i < rec.length;i++){
var teI = rec[i].data;
summary = "";
if(teI.workerId === -1 )continue; // ablage
var workerI = levatis.visualWorkloadWindow.storeWorkers.getById(teI.workerId).data;
if(parseInt(workerI.id) === 1871)continue;
for(var j = 0; j < rec.length;j++){
var teJ = rec[j].data;
if(teJ.workload === 0 || teJ.workerId <= 0)continue;
if(teI.dateId === teJ.dateId){
if(teI.taskId === teJ.taskId ){
var workerJ = levatis.visualWorkloadWindow.storeWorkers.getById(teJ.workerId).data;
if(!levatis.vw.resources.isResource(workerJ)){
var taskId,detailInfo,workload,workloadCss = '',comment = "",commentCss = '',timeCss =
'',formattedTimeStart = '',formattedTimeEnd = '',customer = undefined,customerCss = 'summary-
customer';
var worker = levatis.visualWorkloadWindow.storeWorkers.getById(teJ.workerId).data;
if( parseInt(worker.id) === 1871) continue;
var teContainer = document.getElementById(levatis.vw.resources.workerToResourceHTML_ID +
teI.dateId);
if( teContainer === null) // worker (Zusammenfassung doent exist)
return;
teContainer.removeClassName('tecontainer');

```



```

taskId = teI.taskId;
detailInfo = rec[j].data.detailInfo;
if(detailInfo === undefined)
detailInfo = '';
if(rec[j].data.comment != null){
comment = rec[j].data.comment;
commentCss = 'summary-comment';
}
if(comment === undefined || comment === null){
commentCss = '';
comment = '';
}
if(rec[i].data.formatedTimeStart != undefined && rec[i].data.formatedTimeEnd != undefined){
formatedTimeStart = rec[j].data.formatedTimeStart;
formatedTimeEnd = rec[j].data.formatedTimeEnd;
timeCss = 'summary-time';
}
if(formatedTimeStart === undefined)
formatedTimeStart = '';
if(formatedTimeEnd === undefined)
formatedTimeEnd = '';
workload = rec[j].data.workload;
workloadCss = Math.floor(workload);
if(workload < 10)workload = '&nbsp;&nbsp;&nbsp;'+workload;
customer = rec[j].data.customer
if(customer === undefined){ // sonderzeit
customer = rec[j].data.description;
customerCss = 'template-sonderzeit-'+customer;
}
if(taskId === -1){ // sonderzeit
taskId = rec[j].data.description;
customer = rec[j].data.description;
customerCss = 'template-sonderzeit-'+customer;
}
var el = document.getElementById(taskId+'_'+teI.dateId);
if(el != null){
if(el.innerHTML.match(worker.name) === null){
if(el.innerHTML.match(customer) === null){
el.lastChild.innerHTML += '<tr><td class="'+customerCss+'">'+customer+'</td></tr><tr><td colspan=2 class="summary-detailinfo" colspan=3>'+detailInfo+'</td></tr>'
}
el.lastChild.innerHTML += '<tr class="workloadX'+workloadCss+'">'+
'<td><b class="workloadX'+workloadCss+' ">'+workload+'</b><a class="summary-worker" href="#worker_'+workerJ.id +' ">'+workerJ.name +'</a></td>'+

```

