Buttons auf DocType um unterschiedliche Listen zu öffnen

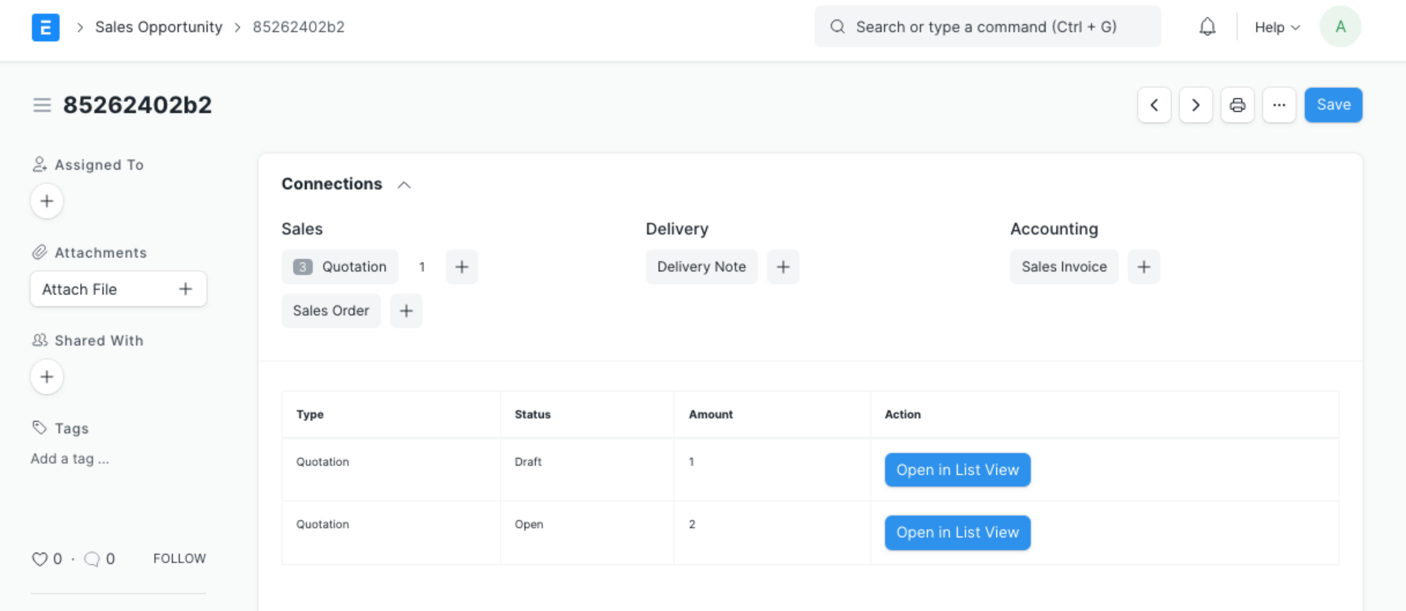
Einleitung

In diesem Beispiel versuchen wir Buttons auf unserem DocType Sales Opportunity zu erstellen hinter welchen jeweils die gefilterete Listenansicht einer bestimmten Verknüpfung, hier Quoation, zu finden ist. Die Buttons sollen für die Userin des Systems ansprechend und verständlich sein. Hier ein paar Beispiele welche mit Hilfe von ChatGPT (immer wieder beeindruckend!) erstellt wurden.

Feel free to copy and use GPLv3 ❤

> Teilweise ist der Code noch überladen, er könnte weiter optimiert werden.

Tabelle mit blauen Buttons

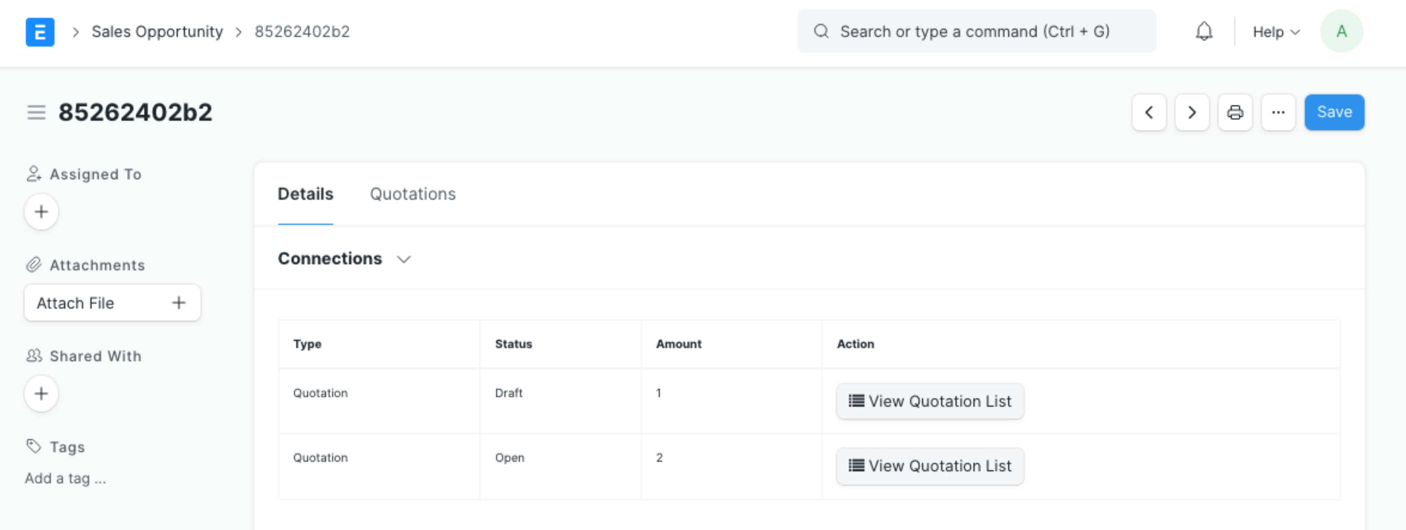


Type	Status	Amount	Action
Quotation	Draft	1	Open in List View
Quotation	Open	2	Open in List View

The Script

```
frappe.ui.form.on('Sales Opportunity', {
    refresh: function(frm) {
        // Get the linked Quotations
        frappe.call({
            method: 'frappe.client.get_list',
            args: {
                doctype: 'Quotation',
                filters: {
                    sales_opportunity: frm.doc.name
                },
                fields: ['status']
            },
            callback: function(response) {
                var data = response && response.message;

                // Count the Quotations by status
                var counts = {};
                if (data && data.length > 0) {
                    data.forEach(function(row) {
                        if (row.status) {
                            if (counts[row.status]) {
                                counts[row.status]++;
                            } else {
                                counts[row.status] = 1;
                            }
                        }
                    });
                }

                // Create or update the visual section with the table
                var section = frm.dashboard.add_section(__('Quotations'));
                var html = `<style>
                                .sales-opportunity-table {
                                    font-size: 70%;
                                }
                            </style>
                            <table class="table table-bordered sales-opportunity-table">
                                <thead>
                                    <tr>
                                        <th>Type</th>
                                        <th>Status</th>
                                        <th>Amount</th>
                                        <th>Action</th>
                                    </tr>
                                </thead>
                                <tbody>`;

                if (Object.keys(counts).length > 0) {
                    Object.keys(counts).forEach(function(status) {
                        html += `<tr>
                                    <td>Quotation</td>
                                    <td>${status}</td>
                                    <td>${counts[status]}</td>
                                    <td><button class="btn btn-primary btn-sm view-opportunity-btn" data-
status="${status}">Open in List View</button></td>
                                </tr>`;
                    });
                } else {
                    html += `<tr>
                                <td>Opportunity</td>
                                <td colspan="3" align="center">No Quotations</td>
                            </tr>`;
                }

                html += `</tbody>
                        </table>`;

                section.html(html);

                // Add click event to the button
                section.find('.view-opportunity-btn').on('click', function() {
                    var status = $(this).data('status');
                    viewQuotationList(status, frm.doc.name);
                });
            }
        });
    }
});

function viewQuotationList(status, salesOpportunity) {
    // Redirect to the Quotation List view with the status filter applied
    frappe.set_route('List', 'Quotation', { 'status': status, 'sales_opportunity': salesOpportunity });
}
```

Tabelle mit Button "List View" aus der Listensicht



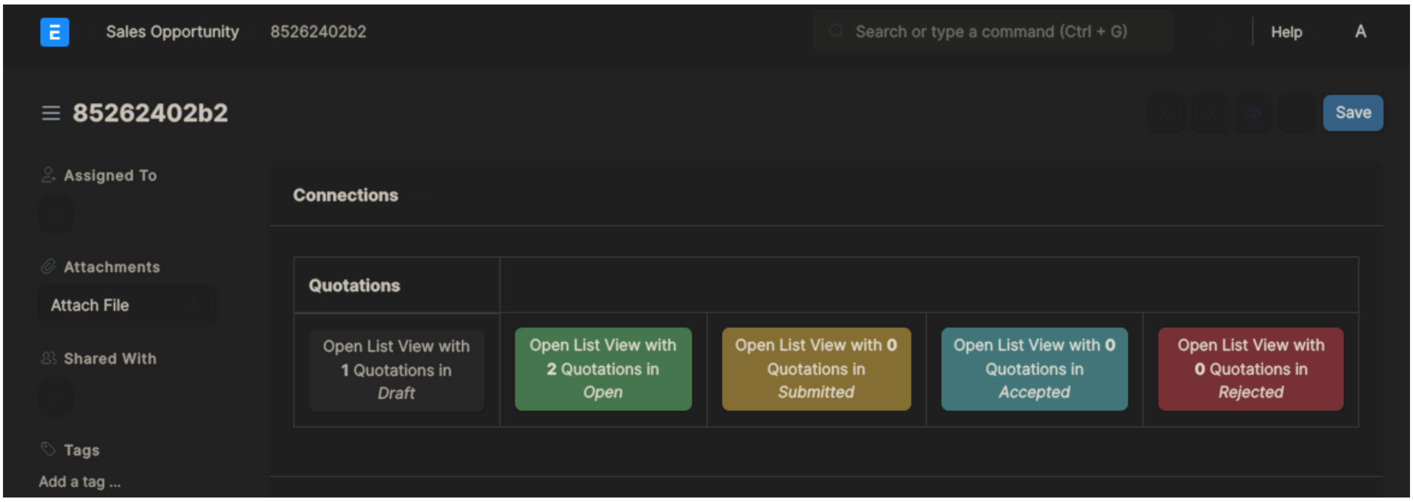
85262402b2

Type	Status	Amount	Action
Quotation	Draft	1	View Quotation List
Quotation	Open	2	View Quotation List

The Script

```
frappe.ui.form.on('Sales Opportunity', {
    refresh: function(frm) {
        // Get the linked Quotations
        frappe.call({
            method: 'frappe.client.get_list',
            args: {
                doctype: 'Quotation',
                filters: {
                    sales_opportunity: frm.doc.name
                },
                fields: ['status']
            },
            callback: function(response) {
                var data = response && response.message;

                // Count the Quotations by status
                var counts = {};
                if (data && data.length > 0) {
                    data.forEach(function(row) {
                        if (row.status) {
                            if (counts[row.status]) {
                                counts[row.status]++;
                            } else {
                                counts[row.status] = 1;
                            }
                        }
                    });
                }

                // Create or update the visual section with the table
                var section = frm.dashboard.add_section(__('Quotations'));
                var html = `<style>
                                .sales-opportunity-table {
                                    font-size: 70%;
                                }
                            </style>
                            <table class="table table-bordered sales-opportunity-table">
                                <thead>
                                    <tr>
                                        <th>Type</th>
                                        <th>Status</th>
                                        <th>Amount</th>
                                        <th>Action</th>
                                    </tr>
                                </thead>
                                <tbody>`;

                if (Object.keys(counts).length > 0) {
                    Object.keys(counts).forEach(function(status) {
                        html += `<tr>
                                    <td>Quotation</td>
                                    <td>${status}</td>
                                    <td>${counts[status]}</td>
                                    <td><button class="btn btn-secondary btn-sm view-opportunity-btn" data-status="${status}" data-doctype="Quotation"><i class="fa fa-list"></i> View Quotation List</button></td>
                                </tr>`;
                    });
                } else {
                    html += `<tr>
                                <td>Opportunity</td>
                                <td colspan="3" align="center">No Quotations</td>
                            </tr>`;
                }

                html += `</tbody>
                        </table>`;

                section.html(html);

                // Add click event to the button
                section.find('.view-opportunity-btn').on('click', function() {
                    var status = $(this).data('status');
                    var doctype = $(this).data('doctype');
                    viewQuotationList(status, doctype, frm.doc.name);
                });
            }
        });
    }
});

function viewQuotationList(status, doctype, salesOpportunity) {
    // Redirect to the Quotation List view with the status and sales opportunity filter applied
    frappe.set_route('List', doctype, { 'status': status, 'sales_opportunity': salesOpportunity });
}
```

Tabelle mit farbigen Buttons



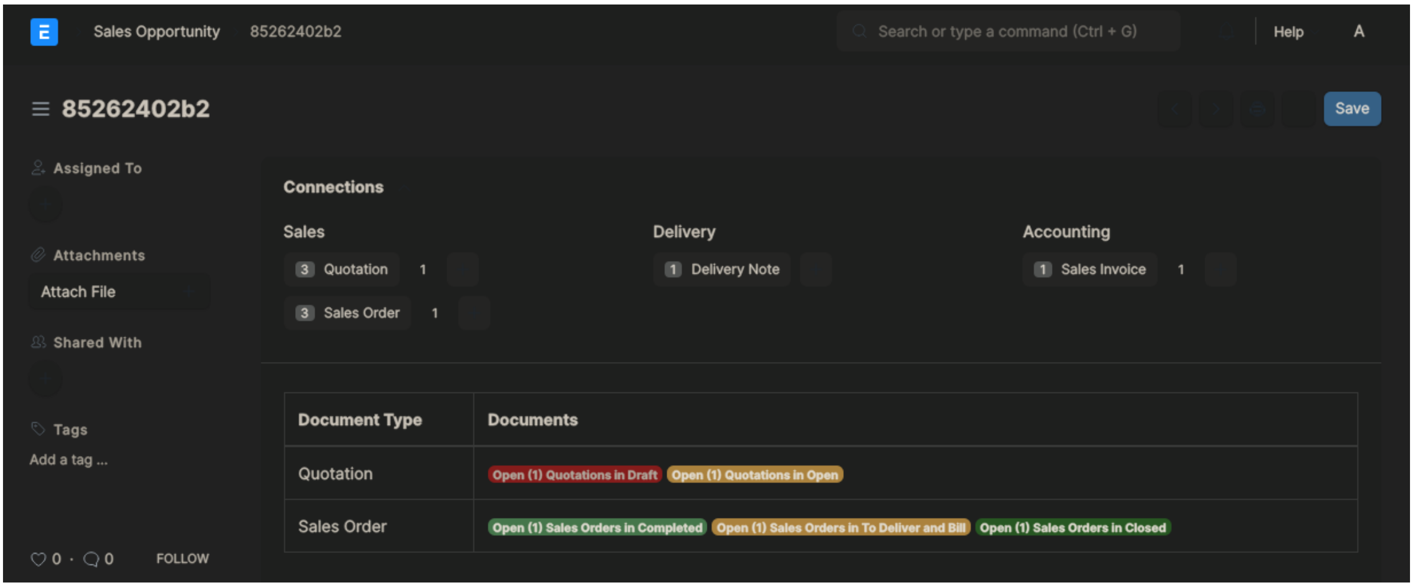
E
Sales Opportunity
85262402b2
Search or type a command (Ctrl + G)
Help
A
85262402b2
Save
Assigned To
Attachments
Attach File
Shared With
Tags
Add a tag ...
Connections
Quotations
Open List View with 1 Quotations in Draft
Open List View with 2 Quotations in Open
Open List View with 0 Quotations in Submitted
Open List View with 0 Quotations in Accepted
Open List View with 0 Quotations in Rejected

The Script

```
frappe.ui.form.on('Sales Opportunity', {
    refresh: function(frm) {
        // Get the linked Quotations
        frappe.call({
            method: 'frappe.client.get_list',
            args: {
                doctype: 'Quotation',
                filters: {
                    sales_opportunity: frm.doc.name
                },
    fields: ['status']
},
callback: function(response) {
    var data = response && response.message;

    // Count the Quotations by status
    var counts = {};
    if (data && data.length > 0) {
        data.forEach(function(row) {
            if (row.status) {
                if (counts[row.status]) {
                    counts[row.status]++;
                } else {
                    counts[row.status] = 1;
                }
            }
        });
    }

    // Create or update the visual section with the table
    var section = frm.dashboard.add_section(__('Quotations'));
    var html = `<style>
                    .sales-opportunity-table {
                        font-size: 100%;
                    }
                    .status-draft {
                        background-color: #f2f2f2;
                        color: #495057;
                    }
                    .status-open {
                        background-color: #28a745;
                        color: #fff;
                    }
                    .status-submitted {
                        background-color: #ffc107;
                        color: #fff;
                    }
                    .status-accepted {
                        background-color: #17a2b8;
                        color: #fff;
                    }
                    .status-rejected {
                        background-color: #dc3545;
                        color: #fff;
                    }
                </style>
                <table class="table table-bordered sales-opportunity-table">
                    <thead>
                        <tr>
                            <th>Quotations</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>`;

    // Add buttons for each status
    var statusOrder = ['Draft', 'Open', 'Submitted', 'Accepted', 'Rejected'];
    statusOrder.forEach(function(status) {
        var statusClass = getStatusClass(status);
        var amount = counts[status] || 0;
        var buttonText = `Open List View with <b>${amount}</b> Quotations in <i>${status}</i>`;

        html += `<td><button class="btn btn-sm view-opportunity-btn ${statusClass}" data-status="${status}" data-doctype="Quotation">${buttonText}</button></td>`;
    });

    html += `</tr>
            </tbody>
        </table>`;

    section.html(html);

    // Add click event to the buttons
    section.find('.view-opportunity-btn').on('click', function() {
        var status = $(this).data('status');
        var doctype = $(this).data('doctype');
        viewQuotationList(status, doctype, frm.doc.name);
    });
```

```
            }
        });
    }
});

function viewQuotationList(status, doctype, salesOpportunity) {
    // Redirect to the Quotation List view with the status and sales opportunity filter applied
    frappe.set_route('List', doctype, { 'status': status, 'sales_opportunity': salesOpportunity });
}

function getStatusClass(status) {
    // Define the class for each status
    var statusClass = {
        'Draft': 'status-draft',
        'Open': 'status-open',
        'Submitted': 'status-submitted',
        'Accepted': 'status-accepted',
        'Rejected': 'status-rejected'
    };

    return statusClass[status] || '';
}
```

Using Badges instead of buttons to save space!



Document Type	Documents
Quotation	Open (1) Quotations in Draft Open (1) Quotations in Open
Sales Order	Open (1) Sales Orders in Completed Open (1) Sales Orders in To Deliver and Bill Open (1) Sales Orders in Closed

The Script

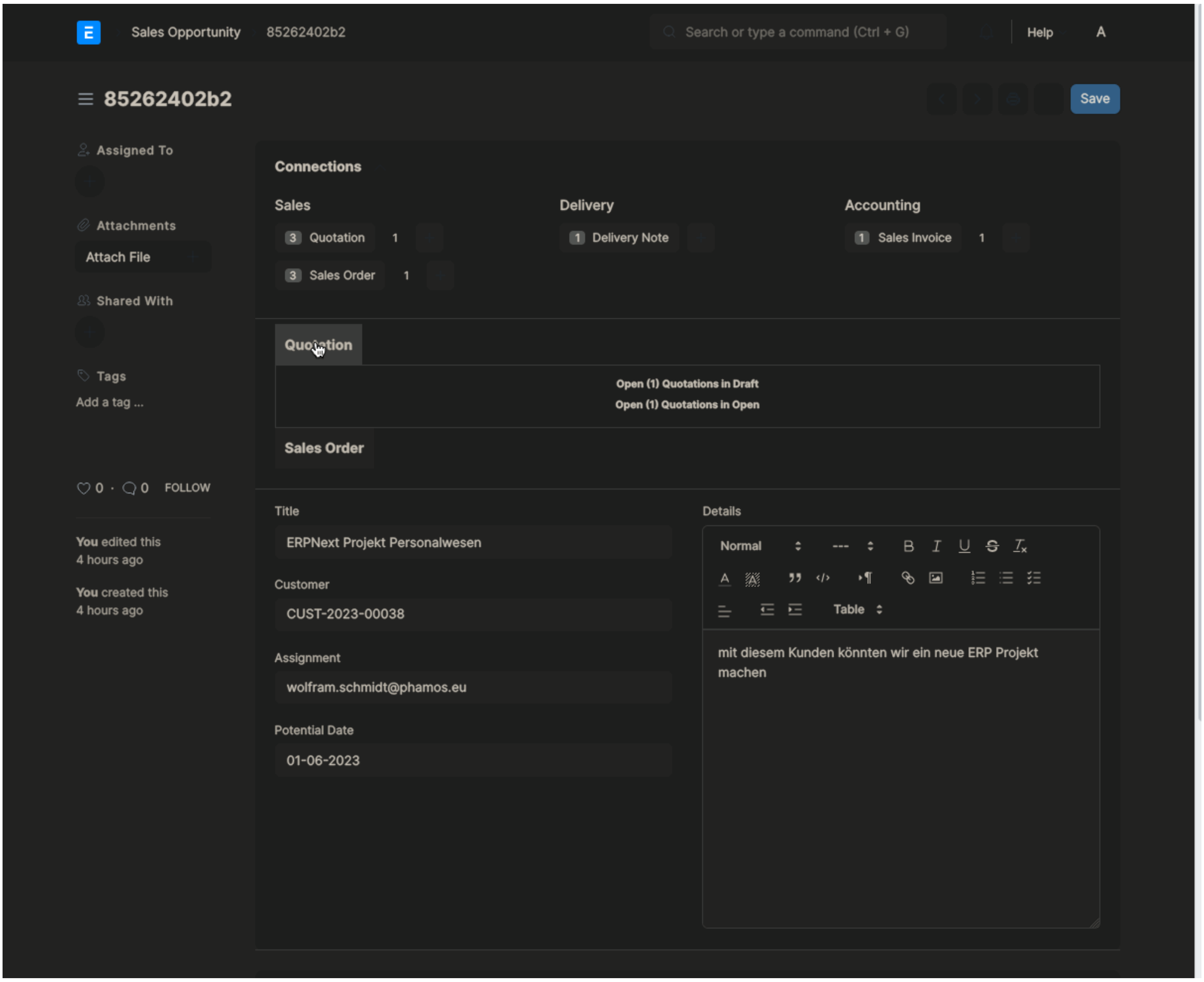
```
frappe.ui.form.on('Sales Opportunity', {
    refresh: function(frm) {
        // Get the linked Quotations
        frappe.call({
            method: 'frappe.client.get_list',
            args: {
                doctype: 'Quotation',
                filters: {
                    sales_opportunity: frm.doc.name
                },
                fields: ['status']
            },
            callback: function(response) {
                var data = response && response.message;

                // Count the Quotations by status
                var quotationCounts = {};
                if (data && data.length > 0) {
                    data.forEach(function(row) {
                        if (row.status) {
                            if (quotationCounts[row.status]) {
                                quotationCounts[row.status]++;
                            } else {
                                quotationCounts[row.status] = 1;
            }
        }
    });
}

// Get the linked Sales Orders
frappe.call({
    method: 'frappe.client.get_list',
    args: {
        doctype: 'Sales Order',
        filters: {
            sales_opportunity: frm.doc.name
        },
        fields: ['status']
    },
    callback: function(response) {
        var data = response && response.message;

        // Count the Sales Orders by status
        var orderCounts = {};
        if (data && data.length > 0) {
            data.forEach(function(row) {
                if (row.status) {
                    if (orderCounts[row.status]) {
                        orderCounts[row.status]++;
                    } else {
                        orderCounts[row.status] = 1;
                    }
                }
            });
        }

        // Create or update the visual section with the table
        var section = frm.dashboard.add_section(__('Sales Documents'));
        var html = `<style>
                        .sales-documents-table {
                            font-size: 100%;
                        }
                        .badge-closed {
```

```
        background-color: green;
        color: #fff;
    }
    .badge-draft {
        background-color: red;
        color: #495057;
    }
    .badge-overdue {
        background-color: red;
        color: #495057;
    }
    .badge-open {
        background-color: orange;
        color: #fff;
    }
    .status-submitted {
        background-color: #ffc107;
        color: #fff;
    }
    .status-accepted {
        background-color: #17a2b8;
        color: #fff;
    }
    .status-rejected {
        background-color: #dc3545;
        color: #fff;
    }
    .badge-to_deliver_and_bill {
        background-color: orange;
        color: #fff;
    }
</style>
<table class="table table-bordered sales-documents-table">
    <thead>
        <tr>
            <th>Document Type</th>
            <th>Documents</th>
        </tr>
    </thead>
```

```
                <tbody>
                    <tr>
                        <td>Quotation</td>
                        <td>`;

// Add badges for Quotations
var quotationStatusOrder = ['Draft', 'Open', 'Submitted', 'Accepted', 'Rejected'];
quotationStatusOrder.forEach(function(status) {
    var amount = quotationCounts[status] || 0;
    if (amount > 0) {
        var badgeClass = getStatusBadgeClass(status);
        var badgeText = `Open (${amount}) Quotations in ${status}`;
        var linkUrl = getQuotationListUrl(status, frm.doc.name);
        html += `<a class="badge ${badgeClass}" href="${linkUrl}">${badgeText}</a> `;
    }
});

html += `</td>
        </tr>
        <tr>
            <td>Sales Order</td>
            <td>`;

// Add badges for Sales Orders
var orderStatusOrder = ['Draft', 'Submitted', 'Completed', 'To Deliver and Bill', 'Overdue', 'Closed'];
orderStatusOrder.forEach(function(status) {
    var amount = orderCounts[status] || 0;
    if (amount > 0) {
        var badgeClass = getStatusBadgeClass(status);
        var badgeText = `Open (${amount}) Sales Orders in ${status}`;
        var linkUrl = getSalesOrderListUrl(status, frm.doc.name);
        html += `<a class="badge ${badgeClass}" href="${linkUrl}">${badgeText}</a> `;
    }
});

html += `</td>
        </tr>
    </tbody>
```

```
                        </table>`;

                        section.html(html);
                    }
                });
            }
        });
    }
});

function getStatusBadgeClass(status) {
    // Define the class for each status badge
    var statusBadgeClass = {
        'Closed': 'badge-closed',
        'Draft': 'badge-draft',
        'Open': 'badge-open',
        'Overdue': 'badge-overdue',
        'Submitted': 'badge-warning',
        'Accepted': 'badge-info',
        'Rejected': 'badge-danger',
        'Completed': 'badge-success',
        'To Deliver and Bill': 'badge-to_deliver_and_bill'

    };

    return statusBadgeClass[status] || 'badge-secondary';
}

function getQuotationListUrl(status, salesOpportunity) {
    // Generate the Quotation List URL with the status and sales opportunity filter
    var url = frappe.urllib.get_base_url();
    url += `/app/list/Quotation?status=${encodeURIComponent(status)}&sales_opportunity=${encodeURIComponent(salesOpportunity)}`;
    return url;
}

function getSalesOrderListUrl(status, salesOpportunity) {
    // Generate the Sales Order List URL with the status and sales opportunity filter
    var url = frappe.urllib.get_base_url();
    url +=
`/app/list/Sales%20Order?status=${encodeURIComponent(status)}&sales_opportunity=${encodeURIComponent(salesOpportunity)}`;
    return url;
}
```

Accordion



Open (1) Quotations in Draft
Open (1) Quotations in Open

Title: ERPNext Projekt Personalwesen

Customer: CUST-2023-00038

Assignment: wolfram.schmidt@phamos.eu

Potential Date: 01-06-2023

Details: mit diesem Kunden könnten wir ein neue ERP Projekt machen

The Script

```
frappe.ui.form.on('Sales Opportunity', {
    refresh: function(frm) {
        // Get the linked Quotations
        frappe.call({
            method: 'frappe.client.get_list',
            args: {
                doctype: 'Quotation',
                filters: {
                    sales_opportunity: frm.doc.name
                },
                fields: ['status']
            },
            callback: function(response) {
                var data = response && response.message;

                // Count the Quotations by status
                var quotationCounts = {};
                if (data && data.length > 0) {
                    data.forEach(function(row) {
                        if (row.status) {
                            if (quotationCounts[row.status]) {
                                quotationCounts[row.status]++;
                            } else {
                                quotationCounts[row.status] = 1;
                            }
                        }
                    });
                }

                // Get the linked Sales Orders
                frappe.call({
                    method: 'frappe.client.get_list',
                    args: {
                        doctype: 'Sales Order',
                        filters: {
                            sales_opportunity: frm.doc.name
                        },
                        fields: ['status']
                    },
callback: function(response) {
    var data = response && response.message;

    // Count the Sales Orders by status
    var orderCounts = {};
    if (data && data.length > 0) {
        data.forEach(function(row) {
            if (row.status) {
                if (orderCounts[row.status]) {
                    orderCounts[row.status]++;
                } else {
                    orderCounts[row.status] = 1;
                }
            }
        });
    }

    // Create or update the visual section with the accordion
    var section = frm.dashboard.add_section(__('Sales Documents'));
    var accordionHtml = `<style>
                            .sales-accordion {
                                font-size: 100%;
                            }
                            .sales-accordion .accordion-title {
                                background-color: #f5f5f5;
                                color: #333;
                                cursor: pointer;
                                padding: 10px;
                                border: none;
                                text-align: left;
                                outline: none;
                                font-weight: bold;
                                transition: background-color 0.3s;
                            }
                            .sales-accordion .accordion-content {
                                padding: 10px;
                                display: none;
                                overflow: hidden;
                                background-color: #fff;
```

```
                                                    border: 1px solid #e7e7e7;
                                                }
                                                .sales-accordion .accordion-content a {
                                                    display: block;
                                                    margin-bottom: 5px;
                                                }
                                                .sales-accordion .accordion-content a:hover {
                                                    text-decoration: underline;
                                                }
                                                .sales-accordion .accordion-title.active {
                                                    background-color: #ccc;
                                                }
                                                .sales-accordion .accordion-content.active {
                                                    display: block;
                                                }
                                            </style>
                                            <div class="sales-accordion">`;

                        // Add accordion for Quotations
                        var quotationStatusOrder = ['Draft', 'Open', 'Submitted', 'Accepted', 'Rejected'];
                        accordionHtml += `<button class="accordion-title">Quotation</button>
                                         <div class="accordion-content">`;

                        quotationStatusOrder.forEach(function(status) {
                            var amount = quotationCounts[status] || 0;
                            if (amount > 0) {
                                var badgeClass = getStatusBadgeClass(status);
                                var badgeText = `Open (${amount}) Quotations in ${status}`;
                                var linkUrl = getQuotationListUrl(status, frm.doc.name);
                                accordionHtml += `<a class="badge ${badgeClass}"
href="${linkUrl}">${badgeText}</a>`;
                            }
                        });

                        accordionHtml += `</div>`;

                        // Add accordion for Sales Orders
                        var orderStatusOrder = ['Draft', 'Submitted', 'Completed', 'To Deliver and Bill', 'Overdue',
'Closed'];
```

```
                        accordionHtml += `<button class="accordion-title">Sales Order</button>
                                          <div class="accordion-content">`;

                        orderStatusOrder.forEach(function(status) {
                            var amount = orderCounts[status] || 0;
                            if (amount > 0) {
                                var badgeClass = getStatusBadgeClass(status);
                                var badgeText = `Open (${amount}) Sales Orders in ${status}`;
                                var linkUrl = getSalesOrderListUrl(status, frm.doc.name);
                                accordionHtml += `<a class="badge ${badgeClass}"
href="${linkUrl}">${badgeText}</a>`;
                            }
                        });

                        accordionHtml += `</div></div>`;

                        section.html(accordionHtml);

                        // Add event listener to toggle accordion content
                        var accordionTitles = section.find('.accordion-title');
                        accordionTitles.on('click', function() {
                            var accordionContent = $(this).next('.accordion-content');
                            accordionContent.slideToggle();
                            $(this).toggleClass('active');
                            accordionContent.toggleClass('active');
                        });
                    }
                });
            }
        });
    }
});

function getStatusBadgeClass(status) {
    // Define the class for each status badge
    var statusBadgeClass = {
        'Closed': 'badge-closed',
        'Draft': 'badge-draft',
        'Open': 'badge-open',
        'Overdue': 'badge-overdue',
        'Submitted': 'badge-warning',
        'Accepted': 'badge-info',
        'Rejected': 'badge-danger',
        'Completed': 'badge-success',
        'To Deliver and Bill': 'badge-to_deliver_and_bill'
    };

    return statusBadgeClass[status] || 'badge-secondary';
}

function getQuotationListUrl(status, salesOpportunity) {
    // Generate the Quotation List URL with the status and sales opportunity filter
    var url = frappe.urllib.get_base_url();
    url += `/app/list/Quotation?status=${encodeURIComponent(status)}&sales_opportunity=${encodeURIComponent(salesOpportunity)}`;
    return url;
}

function getSalesOrderListUrl(status, salesOpportunity) {
    // Generate the Sales Order List URL with the status and sales opportunity filter
    var url = frappe.urllib.get_base_url();
    url += `/app/list/Sales%20Order?status=${encodeURIComponent(status)}&sales_opportunity=${encodeURIComponent(salesOpportunity)}`;
    return url;
}
```

Version #4

Erstellt: 27 Mai 2023 20:35:55 von Wolfram Schmidt

Zuletzt aktualisiert: 27 Mai 2023 22:39:06 von Wolfram Schmidt