

Vertiefungs-Arbeit

M133 / M151

Web-Applikationen entwickeln und eine
DB einbinden

Suna Raman KSB-1HWI_2020

3HWI Kantonsschule am Brühl

Inhaltsverzeichnis

Kernidee der Masterpage	2
Vorteile Masterpage	2
Bestandteile und Aufbau Masterpage	2
Funktionsweise in der Webanwendung	4

Masterpage Theorie

Kernidee der Masterpage

Ein Vorlagentyp, der auf mehreren Seiten verwendet wird. Diese erlauben auch eine Zuordnung zu mehreren Seiten. Bei PowerPoint kann man auch in der Masterfolienansicht ein Masterpage erstellen und wird meistens für Logos, einheitliches Design, Footer usw. gebraucht. So kann man es sich in etwa vorstellen!

Masterpages vereinfacht den Entwicklungsprozess erheblich, da jeder an separaten Webformularen arbeitet.

Vorteile Masterpage

- **Vorlagen:**

Stellt eine oder mehrere Webformularvorlagen bereit.

- **Teamwork:**

Masterpages erleichtern Teams die Zusammenarbeit. Jeder kann mit Webformularen arbeiten und niemand muss sich darum kümmern.

- **DRY → Don't Repeat yourself**

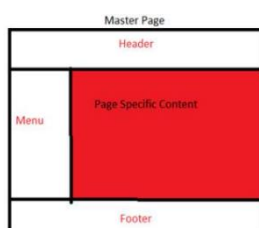
Header und Footer müssen in der Regel nur einmal implementiert werden, da man eine Redundanz vermeiden möchte.

- **KISS → Keep it simple stupid**

Ein Designprinzip, dass ein Design und/oder System so einfach wie möglich sein sollte.

Bestandteile und Aufbau Masterpage

Ein Masterpage besitzt einen bestimmten Aufbau. Dies funktioniert so, dass es eine es eine normale HTML-datei mit Head und Body ist. In der Masterpage (ASPX) kann man den Header, Navigation, Footer einbeziehen, die soeben in jedem Web-Formular existieren. Das DRY-Prinzip (Don't repeat yourself) ist sozusagen eine gute Beschreibung für Masterpages. Content-Placeholder ist auch ein Bestandteil für Masterpages. Man kann neue Webformulare erstellen und diese mit Inhaltsplatzhaltern verknüpfen. Wenn man also nicht mehrere Musterseiten verknüpfen muss, kann man das Hauptelement nur einmal hinzufügen und nicht danach.



Der Header, das Menu und der Footer werden in der Masterpage-Datei erstellt und das Page Specific Content sind für all die Pages zum Benutzen.

Dieser Ausschnitt wird bei der Erstellung von Masterpages erstellt:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Masterlayout.master.cs"
... %>
```

Benennung von unserem Projekt sieht wie folgt aus:

```
Inherits="VertiefungsarbeitV133.Masterlayout" %>
```

Beginn von HTML wird ein Head erstellt, was nicht sichtbar ist auf der Seite:

```
<!DOCTYPE html>

<html>

<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>Master Page Vertiefungsarbeit V133</title>
    <link href="StyleSheet1.css" rel="stylesheet" />
    <asp:ContentPlaceHolder ID="head" runat="server">
        </asp:ContentPlaceHolder>
.....
</head>
```

Beim <body> Teil wird dann alles angezeigt, was man auf der Website dann sehen möchte:

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

Header mit dem St. Gallen Wappen (img). Sonst wurden beim Header ein paar Dinge mit CSS konfiguriert:

```
<form id="form1" runat="server">
    <div class="auto-style1">
        &nbsp;&nbsp;&nbsp;<br />
        <br />
        <br />
        <span class="auto-style2"><a href="https://www.gbssg.ch" class="auto-style5"/> <span class="auto-style4"><strong>GBS SG
&lt;Northwind&gt;</strong></span></a></span>
```

So sieht die Navigation aus und wurde noch mit CSS verschönert:

```
<div class="topnav">
  <a class="active" href="login.aspx">Login</a>
  <a href="regionverwaltung.aspx">Regionverwaltung</a>
  <a href="Impressum.aspx">Impressum</a>
</div>
```

Der ContentPlaceholder ist für die Webforms-Masterpage Verknüpfung da:

```
<div class="content">
    <div>
        <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
            <p>
                &nbsp;</p>
            <p>
                &nbsp;</p>
        </asp:ContentPlaceHolder>
    </div>
</div>
```

Der Footer, um das Design etwas zu verschönern und natürlich, dass ich der Eigentümer dieses Programmes bin.

```
<div class="footer">
  <h1 style="border-style: none; border-color: inherit; ... >&nbsp;</h1>
    <span class="auto-style3">© | Raman Suna, 3HWI</span></h1>

</div>

</form>

</body>

</html>
```

Funktionsweise in der Webanwendung

Zusammenfassend lässt sich sagen, dass Masterpages hervorragend für Gruppenarbeiten geeignet sind, d. h. sie haben eine sehr einfache Vorlage, um sich einen Überblick über Ihr Programm zu verschaffen. Es ist insofern sehr effektiv, als es keine Redundanz im Programm gibt. Sie können Header, Navigation, Footer und Content-Placeholder erstellen. Der Placeholder wird dann mit dem Webformular verknüpft. Auf diese Weise können Sie unabhängig voneinander an verschiedenen Webformularen arbeiten, ohne jemanden zu stören, und Sie müssen die Kopf-/Navigations- und Fußzeile nicht neu erstellen.

Header -Nav - Content-placeholder (Webformular) - Footer

Stored Procedures

Kernidee von Stored Procedures

Eine gespeicherte Prozedur ist eine Anweisung innerhalb eines Datenbankverwaltungssystems, die verwendet werden kann, um einen ganzen Satz von Anweisungen von einem Datenbankclient aufzurufen. Es handelt sich also um einen eigenständigen Befehl, der eine Reihe gespeicherter Befehle ausführt.

Vorteile SP:

- Höhere Sicherheit:
 - Mehrere Benutzer und Client-Programme können Operationen an zugrunde liegenden Datenbankobjekten durch Prozeduren ausführen, ohne direkte Berechtigungen für die zugrunde liegenden Objekte zu haben.
- Verbesserte Leistung:
 - Standardmäßig werden Prozeduren bei der ersten Ausführung kompiliert, um einen Ausführungsplan zu erstellen, der bei nachfolgenden Ausführungen wiederverwendet wird.
- Einfachere Wartung:
 - Wenn eine Clientanwendung Prozeduren aufruft und Datenbankoperationen in der Datenschicht beibehält, müssen nur die Prozeduren für zugrunde liegende Datenbankänderungen aktualisiert werden.

Nachteile SP:

- Debugging:
 - Abhängig von der Datenbanktechnologie kann das Debuggen gespeicherter Prozeduren sehr schwierig oder unmöglich sein. Einige relationale Datenbanken wie SQL-Server verfügen über einige Debugging-Funktionen.
- Tragbarkeit:
 - Komplexe gespeicherte Prozeduren werden nicht immer auf aktualisierte Versionen derselben Datenbank portiert.
- Testen:
 - In Stored Procedures eingekapselte Logik ist sehr schwer zu testen. Datenfehler bei der Verarbeitung gespeicherter Prozeduren werden nur zur Laufzeit generiert.

Erstellen

Zum Erstellen einer Prozedur benutzt man den Befehl CREATE und dann noch den Namen + den SQL-Statement. Mit GO wird es ausgeführt.

```
CREATE PROC (.....);
```

```
AS
```

```
Statement
```

```
GO
```

Löschen

Mit DROP löscht man eine Tabelle.

```
DROP PROC (.....)
```

```
GO
```

Ausführen

Das EXEC(execute) also es wird ausgeführt.

```
EXEC (.....);
```

Verwenden von Parametern

Hier ist ein kurzes Beispiel zur Verwendung der Parameter.

Erstellt wird eine gespeicherte Prozedur, die zum Löschen von Regionen gedacht ist.

```
CREATE PROC spDeleteRegion  
(@RegionDescription VARCHAR(50))  
AS  
DECLARE @DelRegionID AS INT;  
SELECT @DelRegionID = MIN(RegionID) FROM Region WHERE RegionDescription =  
@RegionDescription;
```

Quellen

[Erstellen einer gespeicherten Prozedur - SQL Server | Microsoft Learn](#)

[Vor- und Nachteile der Verwendung von Stored Procedures – SQL – Acervo Lima](#)

[Gespeicherte Prozeduren \(Datenbankmodul\) - SQL Server | Microsoft Learn](#)

<https://youtu.be/V9r-Gp3uNCE>