Vom Projekt bis Fertigstellung – Teil 1: Projektbeginn, Programmablaufplan (PAP)

Neben der eigentlichen Arbeit eines Entwicklers am Programmcode, sind vor allem für größere Projekte Diagramme, Notizen, und Pläne von sehr hoher Bedeutung und werden nur zu gerne unterschätzt. Vernachlässigt man dieses, verliert den Durchblick und am Ende die Motivation überhaupt weiter zu machen. Was ist also wichtig, wenn man ein Projekt plant?

1. Ziele setzen

Wichtig ist, dass man zunächst nicht alle gewünschten Features auf einmal versucht einzubauen, sondern dies bewusst trennt. Auf der einen Seite habe ich die Grundfunktionen bzw. das eigentliche Programm und auf der anderen Seite der Wichtigkeit nach sortierte Features, die ich danach implementieren werde. Beides schreibt man am besten auf ein Blatt Papier, trennt sie und legt sich den Basisfunktionszettel gut Sichtbar auf dem Schreibtisch.

2. Welche Technologien? Welche Themen?

Welche Programmiersprache und Datenbank zum Einsatz kommt ist eher die leichtere Frage, die geklärt werden muss. Wichtiger ist, dass man sich dann aber mit den Themen, die auf einen zukommen genau informiert. Möchte man einen Server programmieren, so schaut man sich die Protokolle genau an und lernt erstmal den theoretischen Teil.

3. Programmablaufplan (PAP)

Wieder mit Zettel und Stift bewaffnet kann man nun einen

ungefähren Ablauf des Programms erstellen. Wichtig ist, dass man sich hier eben nur auf die Basisfunktionen beschränkt. Der Rest kommt, wenn alles fertig ist. Hier gibt es einen einfachen Einstieg in PAP -> http://www.mrknowing.com/2014/03/13/wie-erstelle-ich-ein-pap-programmablaufplan/. Vielleicht ist es nicht wichtig die einzelnen Formen genau zu kennen, doch ist es sehr Hilfreich, sich im groben aufzuschreiben, was genau passieren soll, wenn man dies oder jenes drückt / einstellt. Eben die if / else Verzweigungen sollten gründlich überdacht werden.

Methode mit unendlich vielen Parameter mit params

Hat man den Fall, dass man an eine Methode unendlich viele Parameter übergeben möchte oder die Anzahl vielleicht unbekannt ist, so bietet C# die Möglichkeit über params und ein Array unendlich viele Parameter zu hinterlegen. Ein Beispiel:
[crayon-6809b583aa304962473675/]
Zu beachten ist, dass der Typ unbedingt ein array [] sein muss.

PHP Variablen

Die Variablen in PHP sind zu mindestens bei der Initialisierung Typen undefiniert. Es ist so, als würde man unter C# ausschließlich mit dem var arbeiten.

Weiter kann die Variable entweder local, global oder static sein. Ist eine Variable außerhalb von Scopes {} definiert, so ist diese global. Kann aber nicht in den Scopes verwendet werden.

Wird eine Variable in den Scopes definiert, so gilt sie als local und kann nur in den Scopes verwendet werden.

Befindet sich eine Variable in den Scopes und ist diese als static deklariert, so behält sie ihren Wert, auch wenn der Programmcode die Scopes bereits abgeschlossen hat.

[crayon-6809b583aa7af927221631/]
Das Resultat wäre so eben 1 2 3. Hätte man hier nicht static genutzt, wäre das Ergebnis immer 1, da die Variable, wie bereits angesprochen gelöscht wird, sobald sie die Scopes verlässt.

Interessant sind auch die anonymen Methoden unter PHP. Man hat die Möglichkeit die Funktion quasi in eine Variable zu stecken. Nimmt man das obere Beispiel, so kann man einfach
[crayon-6809b583aa7b4272090737/]
dazu verwenden die Funktion zu anonymisieren.

Variablentypen

PHP hat weit weniger Typen als z.B. noch C#.

Während der boolean, integer, string und null gleich sind, haben folgende Typen so ihre Besonderheiten.

Eine numerische Zahl mit Nachkommastellen ist zwar ein **float**,

fragt man aber mit
[crayon-6809b583aa7b5598649045/]
den Variablentypen ab, sagt PHP, dass es ein **double** ist.

Der **array** in PHP darf alle Möglichen Typen enthalten und erfüllt sogar die Funktion einer aus C# bekannten List oder Diktionary mit Key and Value Prinzip.

Betrachtet man das manuel von PHP.net, so findet man auch eigenartige Funktionen um sogar die im array enthaltenen Werte zu summieren, zu mischen usw.

Man kann in PHP Funktionen und sogar Klassen in eine Variable oder als Variable? speichern. Der Typ ist dann ein object.
[crayon-6809b583aa7b7610026297/]
Wie auch in C# kann man jeden anderen Typ in ein object konvertieren, nennt sich hier aber nicht casten… Dies geschieht aber im Prinzip genau so:
[crayon-6809b583aa7b8507508388/]
Als letzter Type ist die Resource, welche quasi einen Zeiger auf externe Ressourcen zeigt. Diese beinhaltet z.B. das handeln mit Datenbanken, Pdf Dateien usw. Die folgende Liste zeigt einen Auszug davon:

Ressource-List

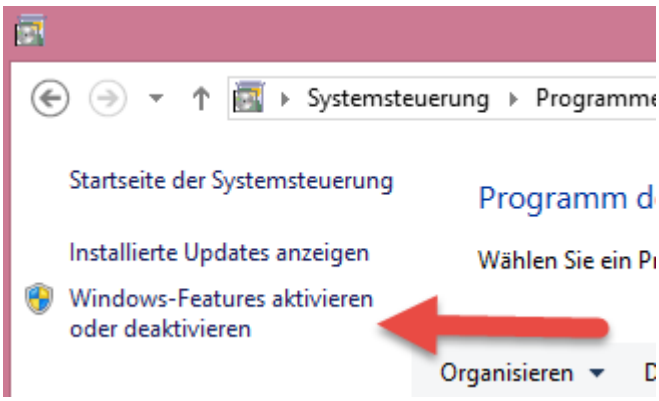
PHP 5.6.8 unter IIS und Windows 8

Mein nächstes Ziel ist die Webkomponente der Programmierung kennen zu lernen. Dazu gehört neben HTML, CSS, JavaScript auch die serverseitige Sprache PHP.

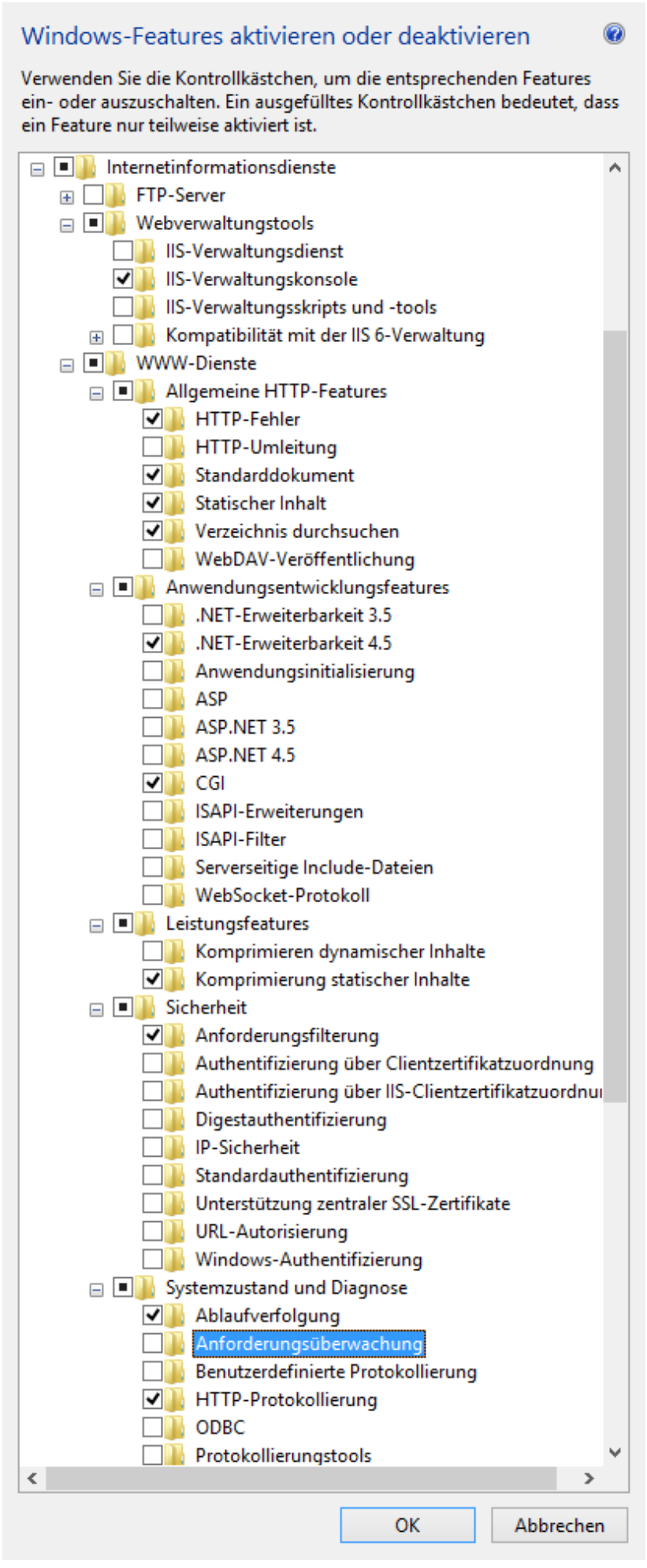
Erstes Ziel ist das Einrichten von PHP auf dem IIS (Man könnte natürlich den einfacheren Weg über XAMP gehen oder den alternativen Weg über den Webserver Apache). Ich wollte das einfach unter IIS machen, weil IIS eigentlich aus einfach von Haus aus auf dem Windows Rechner schlummert und ich so nicht wirklich was installieren brauche. Ok, los gehts.

1. Installieren von IIS

Systemsteurung – Programme – Programme und Features auf



klicken und den Internetinformationsdienste nach folgenden Muster aktivieren.


Windows-Features aktivieren oder deaktivieren
Verwenden Sie die Kontrollkästchen, um die entsprechenden Features ein- oder auszuschalten. Ein ausgefülltes Kontrollkästchen bedeutet, dass ein Feature nur teilweise aktiviert ist.
Internetinformationsdienste
FTP-Server
Webverwaltungstools
IIS-Verwaltungsdienst
IIS-Verwaltungskonsole
IIS-Verwaltungsskripts und -tools
Kompatibilität mit der IIS 6-Verwaltung
WWW-Dienste
Allgemeine HTTP-Features
HTTP-Fehler
HTTP-Umleitung
Standarddokument
Statischer Inhalt
Verzeichnis durchsuchen
WebDAV-Veröffentlichung
Anwendungsentwicklungsfeatures
.NET-Erweiterbarkeit 3.5
.NET-Erweiterbarkeit 4.5
Anwendungsinitialisierung
ASP
ASP.NET 3.5
ASP.NET 4.5
CGI
ISAPI-Erweiterungen
ISAPI-Filter
Serverseitige Include-Dateien
WebSocket-Protokoll
Leistungsfeatures
Komprimieren dynamischer Inhalte
Komprimierung statischer Inhalte
Sicherheit
Anforderungsfilterung
Authentifizierung über Clientzertifikatzuordnung
Authentifizierung über IIS-Clientzertifikatzuordnu
Digestauthentifizierung
IP-Sicherheit
Standardauthentifizierung
Unterstützung zentraler SSL-Zertifikate
URL-Autorisierung
Windows-Authentifizierung
Systemzustand und Diagnose
Ablaufverfolgung
Anforderungsüberwachung
Benutzerdefinierte Protokollierung
HTTP-Protokollierung
ODBC
Protokollierungstools
OK
Abbrechen

Wenn IIS richtig installiert wurde, sind wir für uns unter http://localhost/ erreichbar.

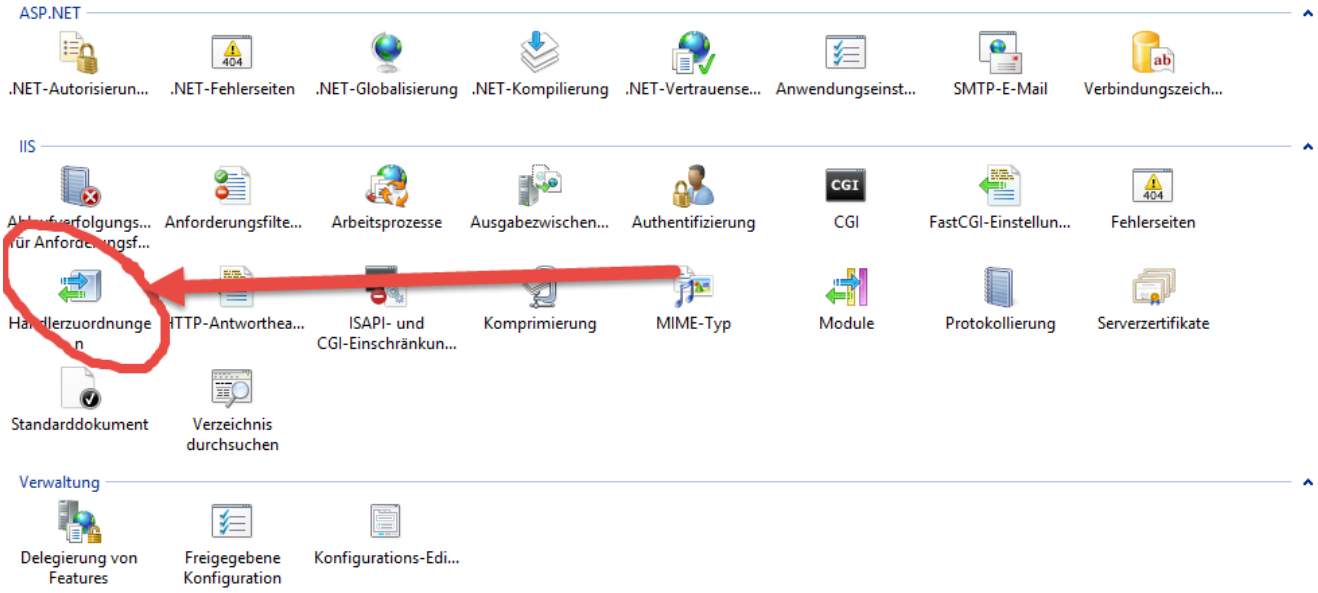
2. PHP installieren

Die Version **VC11 x86 Non Thread Safe** PHP für Windows auf http://windows.php.net/download/ herunterladen, und in C:\Program Files (x86)\PHP\php 5.6.8 entpacken

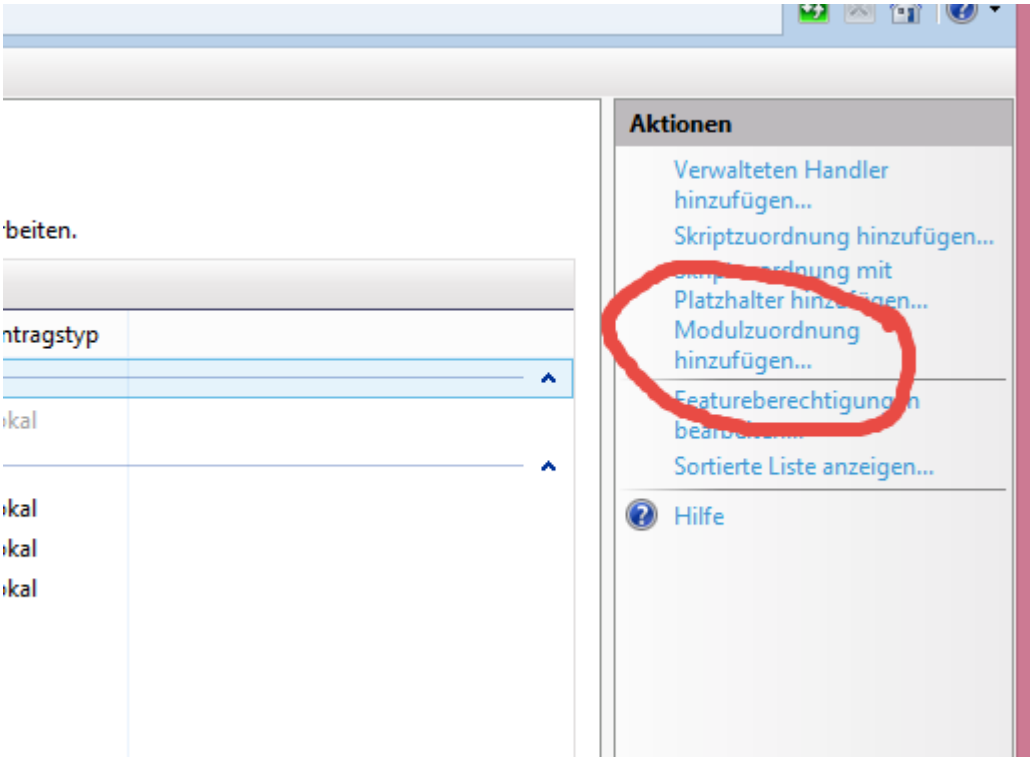
(Wegen den User Rechten muss man die Ordnerstruktur manuell anlegen)

3. IIS für PHP einrichten

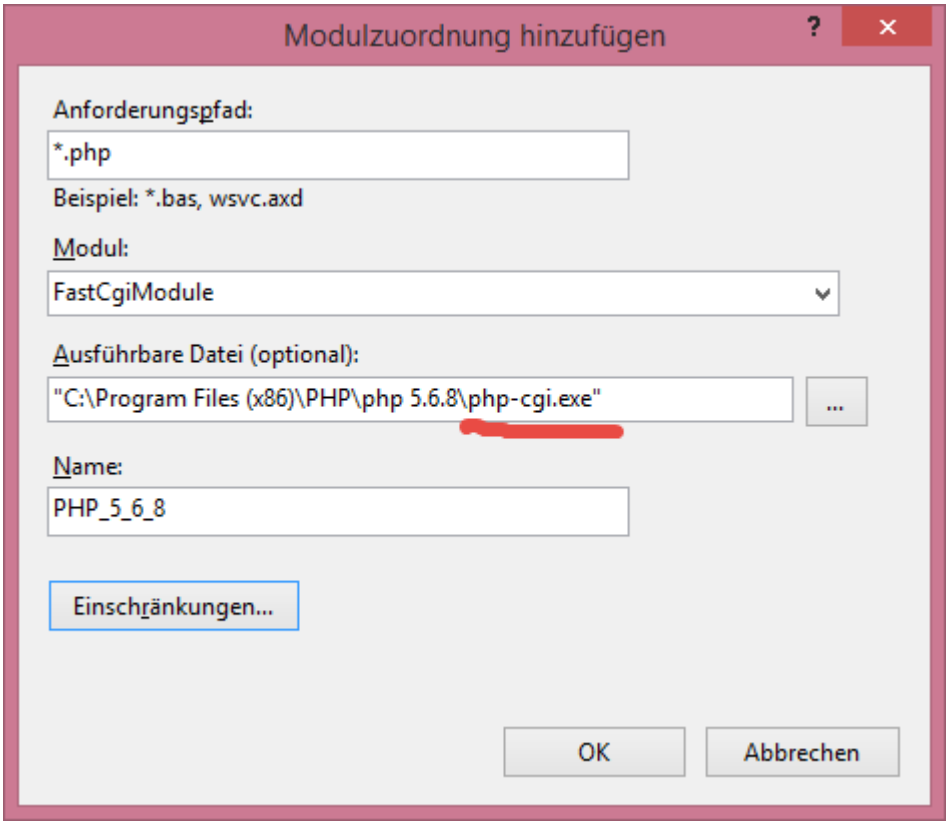
unter Start „IIS“ eintippen und den IIS-Manager wählen.



Dort auf Händlerzuordnungen und



Modelzuordnung hinzufügen…



Fenster folgend ausfüllen. Wichtig, bei der Ausführbaren Datei die **php-chi.exe** auswählen!

Dann auf OK und die Meldung mit **JA** auswählen.

Nun kann man eine info.php Datei mit dem Inhalt

<?php phpinfo() ?> in C:\inetpub\wwwroot speichern und schauen, was passiert wenn wir diese nun unter http://localhost/info.php versuchen zu öffnen.

Werden uns die Werte richtig dargestellt, ist alles richtig installiert. Kann die Seite aber nicht geöffnet werden, fehlt dem höchstwahrscheinlich das C++ Redistributable, welches man hier wiederum herunterladen kann http://www.microsoft.com/en-us/download/details.aspx?id=30679

HttpListenerContext decode/encode Umlaute

Liest man die Url aus dem HttpListenerContext, die Umlaute wie äöü enthält, so sieht das ungefähr so aus:

aus süß wird s%FC%df.

Abhilfe schaft da die Klasse HttpUtility:
[crayon-6809b583aa9ff572671964/]
möchte man zurück encodieren macht man einfach :
[crayon-6809b583aaa03349851165/]

Button Color aus Hexwert ändern

[crayon-6809b583aab8a358598614/]
um Button zurück zu setzen:
[crayon-6809b583aab8e941677352/]

guter einfacher ColorPicker:

http://www.colorpicker.com/

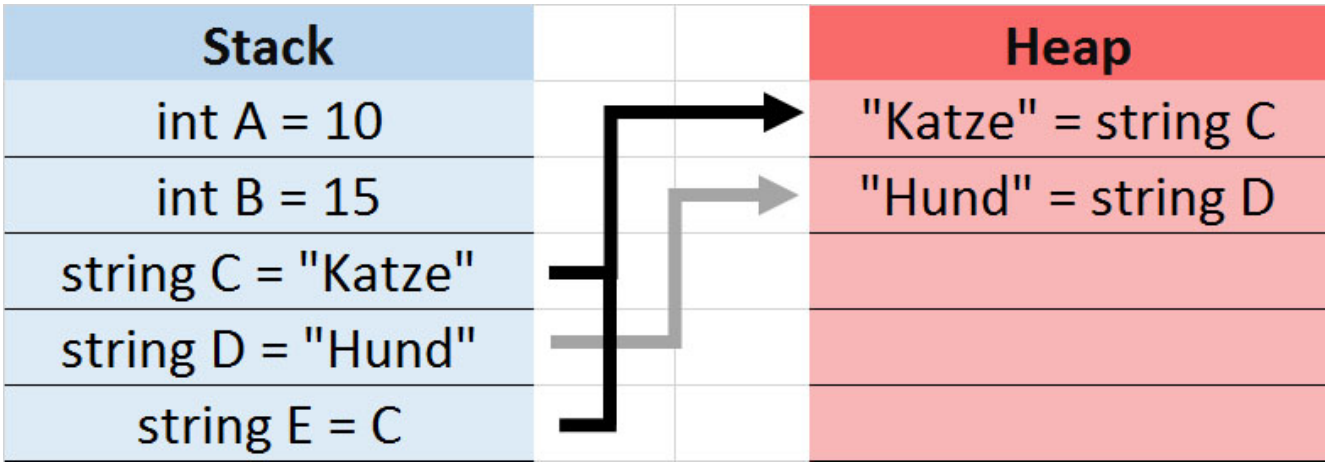
Referenztypen, Wertetypen

Stack und Heap sind Teil des Arbeitsspeichers, die zu mindestens unter C# vom Kompiler automatisch verwaltet wird.

Beim **Stack** (dt. Stapel) werden die Daten quasi aufeinander gelegt. Wenn Speicher freigegeben werden kann, so wird dies auch von oben heraus getan (LIFO Prinzip). Durch dieses Prinzip ist der Stack sehr schnell in seiner Arbeitsweise. Verlässt der Programmcode die geschwungene Klammer { }, werden sämtliche innerhalb angelegte Variablen aus dem Stack entfernt. Obwohl der Speicher sehr gering ist, wächst und schrumpft er während des Programmablaufs.

Der **Heap** ist nicht so strukturiert wie der Stack, dort liegen die Daten quasi durcheinander und werden vom Stack aus per Zeiger gefunden. Deswegen ist der Heap auch deutlich langsamer in seiner Arbeitsweise.

Was in den Stack und was in den Heap gelangt, bestimmen unter anderem die Datentypen. Weiter beinhaltet der Stack auch die Zeiger der Variablen auf den Heap.



Stack – Wertetypen:

• Alle numerischen Datentypen
• Boolean, Char und Date
• Alle Strukturen, auch wenn ihre Member Verweistypen sind
• Enumerationen, da der zugrunde liegende Typ immer SByte, Short, Integer, Long, Byte, UShort, UInteger oder ULong ist

Heap – Referenztypen, Verweistypen:

• String
• Alle Arrays, auch wenn ihre Elemente Wertetypen sind
• Klassentypen, z.B. Form
• Delegaten
• reine Objekte

Das Verschieben der Daten vom Stack (Wertetyp) zu Heap (Referenztyp) bezeichnet man als **boxing**
[crayon-6809b583aad10723739810/]
und vom Heap zu Stack als **unboxing**
[crayon-6809b583aad14297548839/]

Auch beim **casten** können die Daten vom Heap zum Stack geschoben werden:

[crayon-6809b583aad15544411070/]

Quelle:

Understanding Boxing and Unboxing

Aus einer anderen Klasse, aus einem anderen Thread in MainWindow schreiben

Möchte man aus einer anderen Klasse, aus einem anderen Thread etwas in die Mainklasse Steuerelemente schreiben, stößt man auf 2 Probleme:

1. Man kann aus Thread 2 nicht in Thread 1 schreiben

2. Man kann nicht, ohne ein Objekt angelegt zu haben nicht in die Steuerelemente schreiben.

Abhilfe schafft ein kleiner Trick.

MainWindow.cs:
[crayon-6809b583aaea7226632598/]
[crayon-6809b583aaeaa614592831/]

meineAndereKlasse.cs:
[crayon-6809b583aaeab360997480/]
Kleine Erweiterung, selbes Prinzip um ein Imagecontrol zu

ändern:

MainWindow.cs:
[crayon-6809b583aaead963203849/]
meineAndereKlasse.cs:
[crayon-6809b583aaeae270717160/]

***Quelle**: Stackoverflow*

Schnell mal ein Bild einfügen

Dazu einfach ein Bild in den Designer ziehen.

Möchte man im Code-Behind, dann das Bild ersetzen, gibt man der Image-Control einen Namen und führt folgendes aus:
[crayon-6809b583ab0ae862922376/]

TCP / IP – Teil 3: HTTP – Protokoll

Das HTTP (Hypertext Transfer Protocol) baut auf das TCP auf, bildet in seiner Übertragung jedoch weitreichende eigene Konzepte, als das die Übertragung über TCP/IP.

Ebenfalls ist das HTTP ist heute das Standardprotokoll um Webseiten im Webbrowser darzustellen, welches sich laut W3C im 1.1 Standard befindet. HTTP/2 befindet sich aber bereits in

Entwicklung.

Eine Ausführung sieht im kurzem immer so aus:

1. Server wartet über einen Port i.d.R. Port 80 auf eine Anfrage

2. Client sendet über POST oder GET eine Anfrage (Request) an den Server

3. Falls dieser Erreichbar ist sendet der Server eine Antwort (Response) zurück

URL

Eine URL kann z.B. so aussehen:
[crayon-6809b583ab25d951821945/]
http – ist das verwendete Protokoll

www.devandy.de – hostname oder Domain

/meinOrdner/2015-04-01/ – Verzeichnis wo sich die Datei auf dem Server befindet

datei01.php – die Datei an die ein Request gesendet wird

?vorname=peter&nachname=lustig – Querystring, der immer mit einem ? beginnt. vorname= und nachname= bilden dabei die variablen und peter und lustig sind die darin enthaltenen werte.

Übertragungsmethode GET

Die Übertragung findet über die URI, über den Querystring statt. Die Anweisungen aus der method sind in der URL Sichtbar und können für spätere selbe Zwecke gespeichert werden

Übertragungsmethode POST

Die Übertragung über POST hingegen kann man nicht bookmarken, da die Übertragung im Content geschieht. Dabei werden die

Informationen im Header weiter gegeben. Ein Dateiupload ist z.B. nur mit Post möglich, ein Textfeld mit einem Roman macht auch nur über Post Sinn, weil dieser nicht begrenzt ist.

1. Schritt: Server wartet auf Anfrage

In der Regel sind die Serverprogramme wie der IIS Server oder Apache standardisiert eingerichtet, so dass immer über Port 80 gelauscht wird. Man kann dies aber Serverseitig auch auf einen anderen Port umlegen.

2. Schritt: Client Request

Der Client wie der Webbrowser sendet an den Host über das http Protokoll eine Anfrage in Form einer URL.

Dabei enthält diese Anfrage 2 Teile. Zum einen die Anforderungszeile (URL) und zum anderen einen Header, der so aussehen kann:
[crayon-6809b583ab262756643728/]
Die erste Zeile beinhaltet mit GET die Methode über den Request, index.html die zu aufrufende Datei und dann die Protokollart und Versionsnummer.

Darunter folgen weitere Informationen wie Ursprungsland, Bilder erlauben, Browsertyp und Versionsnr, Referer (welche Seite man vorher besucht hat), usw.

3. Schritt Server Response

Ist der Server erreichbar, sendet er eine 3 teilige Antwort wieder. Diese besteht aus

1. Statuszeile
[crayon-6809b583ab265115654105/]
bestehend aus dem Protokoll, Versionsnummer, Statusnummer und Status.

2. Header

Der Header beinhaltet das Date, Datum und Uhrzeit wann die Antwort geschickt wurde, Server Name und Versionsnummer des Webservers, Content-Length, Länge des Nachrichten-Body in Byte usw.

3. Body

Darin befindet sich der eigentliche HTML Code, der dann im Browser angezeigt wird.

Beispiel Projekt in C#, ein Http Server:

HTTP Server

Quellen:

SelfPHP Get/Post

Andreas Olesch