ITandTEL PHP coding standards

Regeln und Empfehlungen

Hintergrund Coding Standard

Die ITandTEL Coding Standards sind Regeln und Empfehlungen um eine sichere und effiziente Softwareentwicklung zu gewährleisten. Sinn und Zweck dieses Dokuments ist es nicht, einen einzigen gültigen Coding Standard zu entwerfen. Es wird Projekte geben, die eine Abweichung bzw. Erweiterung dieses Standards erfordern.

Die Coding Standards wurden entworfen um im Allgemeinen folgende Punkte in der Software-Entwicklung abzuwickeln.

- Kommunikation
- Urlaubsvertretung
- Funktionierender Code
- Übersichtlicher Code
- Wiederverwendbarkeit des Codes
- Code, der leicht zu erweitern ist

Jeder Mitarbeiter an einem entsprechenden Projekt muss sich mit diesen Regeln und Empfehlungen vertraut machen und ist verpflichtet, diese einzuhalten. Sollte eine Abweichung notwendig sein, ist wie in „

Abweichungen vom coding standard“ vorzugehen.

Jeder im Team kann auf Abweichungen der coding guidelines aufmerksam machen und diese dem Service & Betrieb Prozess- Verantwortlichen zu melden.

Dieser Coding Standard gilt für die Entwicklung von PHP Applikationen. Er gilt nicht für die Entwicklung von TYPO3 Extensions.

ITandTEL
Business IT-Lösungen

Inhalt

Abweichungen vom coding standard

Sollte aus verschiedenen Gründen eine Abweichung vom coding standard notwendig sein, muss dies dokumentiert und begründet werden.

Kommentare

Jede PHP- Datei muss das Copyright und einen Kommentar enthalten, der die Funktionalität beschreibt.

Tabelle 1

```
/**
 *  Copyright notice
 *
 * (c) 2010 Michael Raberger <Michael.Raberger@ITandTEL.at>
 * All rights reserved
 *
 * Plugin 'Banner Slideshow' for the 'mr_banner' extension.
 *
 * @author       Michael Raberger <Michael.Raberger@ITandTEL.at>
 * @package      TYPO3
 * @subpackage   tx_mrbanner
 *
 * This copyright notice MUST APPEAR in all copies of the script!
 */
```

Änderungskommentare

Sollte bei einem Projekt kein Subversion System zur Verfügung stehen, sind Änderungen im Source Code mit Kommentaren zu versehen.

Diese Kommentare bestehen aus dem Kürzel des Mitarbeiters, der die Änderung durchführt inkl. dem Änderungsdatum nach dem ISO 8601 Standard (http://de.wikipedia.org/ISO-8861).

Tabelle 2

```
// RAI 2010-02-17: bugfix sql statement
```

Wenn ein SVN zur Verfügung steht, kann auf diese Art der Dokumentation verzichtet werden.

Systemkritische Variablen und Konstanten

Systemkritische Variablen und Konstanten werden mit folgender Notation kommentiert

Tabelle 3

```
$bool = ($i > $j) // IF $i > $j: $bool becomes FALSE or FALSE
```

Längere Kommentare werden wie folgt notiert

Tabelle 4

```
/*
 *
 *
 */
```

Jedes Kommentar muss Transparenz und Verständnis in den Source- Code bringen.

Debugging

Für die Entwicklung dürfen Debug- Meldungen angezeigt und geloggt werden. Es ist nicht erlaubt, Variablennamen und mögliche / erwartete Rückgabewerte in Kommentaren zu verwenden.

Tabelle 5

```
error_reporting(E_ALL);
print_r();
var_dump();
```

Im Echtsystem sind keine Informationen auszugeben, die Informationen zur verwendeten Server Infrastruktur und zum internen Code Aufbau preis geben.

Es sind alle Debug- Informationen am Anfang des Quellcodes auszuschalten.

Tabelle 6

```
error_reporting(0);
```

Dokumentation

Um eine Dokumentation bzw. eine API Referenz zu erstellen, sind Programme wie Javadoc, Doxygen oder phpDocumentor zu verwenden.

Variablenbezeichnungen

Bezeichner von Variablen, Funktionen, Klassen und Konstanten sind in Englisch zu definieren. Alle Kommentare und Dokumentationen sind ebenfalls in Englisch zu verfassen.

Unbedingt zu vermeiden ist eine Mischung der englischen und der deutschen Sprache.

Variablenbezeichnungen wie

- $deutsche_variable

sind nicht zulässig.

Variablennamen sind so zu wählen, dass sie etwas über den Zweck der Variablen aussagen. Ebenso sollten richtig platzierte Unterstriche die Lesbarkeit der Variablen sorgen.

Namen müssen eindeutig vergeben werden

- Zwei Variablen nicht nur durch Groß- und Kleinschreibung unterscheiden, also nicht Funktion A „get_data()“ und Funktion B „Get_data()“
- Variablen nicht durchnummerieren, also nicht $var1, $var2, …
- Zahlen in Namen vermeiden
- Abkürzungen sind zu vermeiden (siehe Abkürzungen)

Benennung von Variablen und Funktionen erfolgen mit Unterstrich und in Kleinbuchstaben

Konstanten werden in Großbuchstaben definiert

Ausnahme:
Zählervariablen in Schleifen können mit einem Buchstaben („$i“, „$j“, „$k“, …) definiert werden.

Abkürzungen

Abkürzungen sind zu vermeiden um die Lesbarkeit und Wartbarkeit des Quellcodes zu verbessern.

- Verwenden Sie so wenige Abkürzungen wie möglich
- Nur Abkürzungen verwenden, die jeder im Team kennt
- Auf keinen Fall Abkürzungen verwenden, die zweideutig sein könnten

Dieser Coding Standard soll vermeiden, dass bei einem späteren Betrachten des Quellcodes die Einarbeitungszeit auf ein Minimum zu reduzieren.

Tabelle 7

```
$group_id = 1;          // instead of $grpid
$name_length = 20;      // instead of $namLn
$reset_printer = TRUE; // instead of $rstprt
```

Include Dateien

Alle Dateien die inkludiert werden, müssen im Verzeichnis includes abgelegt werden. Sollten Dateien nicht in diesem Verzeichnis, sondern im Hauptverzeichnis abgelegt werden, müssen diese die Dateiendung „.inc.php“ haben. Eine reine Bezeichnung .inc ist nicht zulässig, da diese Dateiendung standardmäßig nicht geparst wird und dadurch alle Informationen (auch Passwörter) im Klartext vorliegen.

Knorrstraße 10 I A-4600 Wels
T +43 7242 9396-7100
F +43 7242 9396-7182
info@itandtel.at I www.itandtel.at
Eine Marke der E-Werk Wels AG.
Elektrizitätswerk Wels AG
Stelzhamerstraße 27 I A-4600 Wels
DVR Nr. 0006815
UID Nr. ATU 23478001
FN Nr. 102455W I LG Wels

Empfohlene Ordnerstruktur

Tabelle 8

Ordner	Beschreibung
includes	Dateien, die in das Haupt- Script eingebunden werden
stylesheets	Kaskadierende Stylesheet Dateien
javascript	JavaScript Dateien
images	Bilddateien
media	Multimedia- Dokumente

Sollte es das aktuelle Projekt verlangen, kann diese Ordnerstruktur in Abstimmung mit dem Team abgeändert werden. Diese Änderung muss wiederum dokumentiert und begründet werden.

Funktionen, Parameter und Rückgabewerte dokumentieren

Kommentare sind ein wesentlicher Bestandteil der Dokumentation.

Tabelle 9

```
/*
 * This function creates a database connection
 *
 * @param   string   $host:       RDBMS / MySQL Host: e.g. "localhost"
 * @param   string   $user:       MySQL User: e.g. mysql_user
 * @param   string   $pass:       MySQL Passwort: e.g. mysql_password
 * @param   string   $database:   MySQL Database: e.g. mysql_database
 * @return  boolean  $connection: TRUE or FALSE
 */
function connect_dbs($host, $user, $pass, $database) {
   $connect = …
   Return $connection; // returns TRUE or FALSE
}
```

Pro Zeile wird die Erklärung für einen Parameter angegeben. Diese Zeile sollte nach Möglichkeit alles über den Parameter aussagen. Wichtig sind die Datentypen („string“, „boolean“), die Namen aller Parameter ($host, $user, $pass, $database), eine Beschreibung und ein Beispiel. Die Angabe des Wertebereichs ist optional.

Knorrstraße 10 | A-4600 Wels
T +43 7242 9396-7100
F +43 7242 9396-7182
info@itandtel.at | www.itandtel.at
Eine Marke der E-Werk Wels AG.
Elektrizitätswerk Wels AG
Stelzhamerstraße 27 | A-4600 Wels
DVR Nr. 0006815
UID Nr. ATU 23478001
FN Nr. 102455W | LG Wels

Wertebereiche sollten bei Integer Parametern verwendet werden, um Besonderheiten (z.B.: nur positive Werte) zu kennzeichnen.

Tabelle 10

```
@param   integer   $rows: (range 1 - 65536): e.g. 257
```

Funktionen

Eine Verschachtelung von Funktionen ist nicht zulässig. Sollten mehrere Funktionen auf eine Variable angewendet werden müssen, sind diese nacheinander durchzuführen.

Tabelle 11

```
$variable = trim($variable);
$variable = substr($variable, 0, 1);

FALSCH
$variable = trim(substr($variable, 0, 1));
```

Klammern für eine Funktion () stehen direkt am Funktionsnamen. Die Parameter der Funktion werden ohne Leerzeichen notiert, mit Ausnahme der Kommata.

Tabelle 12

```
function devision($devisor, $dividend) {
  ..
}

devision(5, 2);

FALSCH
function devision ( $devisor , $dividend ) {
  ..
}

devision(5,2);
```

Ausnahme:

Die Notation kann abweichen, wenn Funktionen mit vielen Parametern zum Einsatz kommen.

Funktionen mit langen oder vielen Parametern müssen übersichtlich notiert werden.

Es wird oft notwendig sein, Funktionen mit vielen Parametern zu benutzen und / oder lange Variablennamen zu definieren.

Tabelle 13

```
mktime(
    0,
    0,
    0,
    $month,
    $day,
    $year
);
```

Validierung

HTML Templates müssen validiert werden. Templates die erfolgreich validiert wurden, können in Zukunft leichter gewartet werden, da Designfehler schneller gefunden werden.

Zum validieren des HTML Quellcodes ist der offizielle Validator des W3C Konsortiums zu verwenden (http://validator.w3.org).

Stylesheets sind ebenfalls über den von W3C zu Verfügung gestellten Validator zu validieren.

Template Engines

Der Code muss von Design getrennt werden. Vorteile die der Einsatz von Template Engines bringen sind

- Arbeiten im Team
- Module bzw. modulares Programmieren
- Wiederverwendbarkeit
- Design von Designern erledigen lassen, Programmierarbeiten an Programmierer übergeben

Für den Einsatz in unseren Projekten wurde die Template Engine *Smarty* evaluiert und es darf, wenn eine Template Engine zum Einsatz kommt, keine andere Bibliothek verwendet werden.

Trotz der Verwendung von Template Engines hat die Lesbarkeit des Quellcodes höchste Priorität. Sollte ein Projekt komplexe Strukturen verlangen, ist im Anlassfall zu besprechen, ob von der Verwendung der Template Engine abzusehen ist.

SQL Befehle

SQL Funktionen wie SELECT, UPDATE, DELETE, MAX(), SUM() usw. werden großgeschrieben.

Datenbank Abstraktions- Layer

Zur Verbindung zu Datenbanken und zur Verarbeitung von SQL Statements wird der PHP interne Datenbank Abstraktions- Layer PHP Data Objects (PDO) verwendet. SQL Statements sind als prepared Statements auszuführen und die inkludierte Funktion mysql_real_escape_string bzw. $dbh->quote(); zu verwenden. Sollten Statements über $dbh->exec() bzw. $dbh->query() ausgeführt werden, muss die interne Funktion PDO::Statements->quote(); verwendet werden.

Ausnahmen

Es kann von der Benutzung der PHP Data Objects abgesehen werden, wenn es ausdrücklich vom Kunden gewünscht wird, die bestehende Infrastruktur die Verwendung nicht erlaubt oder es die Anforderungen des Projektes einen anderen Layer oder eine andere Bibliothek erfordern.

Debugging

Für die Entwicklung dürfen Debugging- Informationen angezeigt und geloggt werden. Im Falle von PDO sind die Attribute EXCEPTION bzw. NOTICE zu verwenden.

Tabelle 14

```
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_NOTICE);
```

Sämtliche Debug- Informationen die Hinweise zur Datenbankstruktur oder internen Quellcode geben, müssen entfernt werden. Dies kann über ein Flag in einem config File passieren oder durch entfernen der entsprechenden Anweisungen.

Insert Anweisungen

Insert Anweisungen müssen die einzelnen Spalten für die VALUES-Klausel enthalten. Um zu verhindern, dass jeder Insert fehlschlägt, werden die Spalten explizit ausgewiesen und über VALUES dann mit Inhalten belegt.

Tabelle 15

```
$dbh->prepare("INSERT INTO mysql.tyble (value)
               VALUES('string_value')");
```

Unbenutzter Code

Code, der nicht benutzt wird, muss gelöscht werden. Es reicht nicht, den Code über Kommentare auszublenden und im Produktiv- Qellcode zu belassen. Sollten Codeteile zu einem späteren Zeitpunkt erneut benötigt werden, muss eine Sicherheitskopie der Datei erstellt und danach die überflüssigen

Zeilen gelöscht werden. Eine andere Möglichkeit ist, eine neue Revision im Subversion Repositorie zu erstellen. Der Code kann zu einem späteren Zeitpunkt erneut über einen checkout wiederhergestellt werden.

Codestrukturierung

Allman- Stil

Geschweifte Klammern werden dem Allman- Stil entsprechend untereinander in die gleiche Spalte gesetzt. Die Klammer erhält eine eigene Zeile.

Der Allman- Stil unterstützt die Übersichtlichkeit und ermöglicht ein leichteres kopieren ganzer Code-Blöcke und auch das Bearbeiten in einem Code Editor wie *vi*.

Tabelle 16

```
if ($a > $b)
{
   echo "a ist größer als b";
}
else
{
   echo „a ist kleiner als $b";
}

Ungültig
if ($a > $b) {
  …
}
…
```

Kontrollstrukturen / Klammern

Jede Kontrollstruktur hat einen Block, der mit geschweiften Klammern umschlossen wird. Die Ausnahme bilden SWITCH- CASE Strukturen, wo ein „case:“ mehrere Anweisungen enthalten kann und dann mit einem „break;“ abgeschlossen wird.

Tabelle 17

```
while ($row = PDOStatement->fetchAll(PDO::FETCH_ASSOC))
{
  print_r($row);
}

Ungültig
while ($row = PDOStatement->fetchAll(PDO::FETCH_ASSOC))
print_r($row);
```

Einrücken von Quellcode

Zum Einrücken von Quellcode werden Tabulatoren, keine Leerzeichen verwendet. Tabulatoren werden verwendet, damit jeder Entwickler die gewünschten Abstände in seinem favorisierten Texteditor einstellen kann.

Zweitens erschweren Leerzeichen das Refactoring.

String Deklarationen

Bei der Deklaration von Strings dürfen keine Anführungsstriche sondern Hochkommata verwendet werden. Die Verwendung von Anführungszeichen birgt Probleme beim nachträglichen Einsetzen von Konstanten in den String.

Tabelle 18

```
echo 'This value ' . $value . , should be a string.';

Ungültig
Echo „This value $value should be a string.";
```

Es dürfen keine Anführungsstriche bei String-Deklarationen verwendet werden, sondern Hochkommata in Verbindung mit dem Verkettungsoperator.

Applikation- Security

An die Absicherung der Applikation PHP seitig muss von Anfang an gedacht werden und nicht nach Fertigstellung des Skriptes. Als Sicherheitsleitfaden gilt der vom PHP Security Consortium veröffentlichte Leitfaden (www.phpsec.org) oder der ITandTEL Wiki Artikel PHP Sicherheit http://wiki.itandtel.at/index.php/C:_PHP_Security_Guide. Ebenfalls müssen die Empfehlungen der ÖNORM A 7700 angewendet werden http://wiki.itandtel.at/index.php/Datei:A_7700.pdf.

Grundlegende Anweisungen, die zu befolgen sind:

- REQUEST-Variablen sind bei register_globals=on auch "normale" Variablen
- REQUEST-Variablen dürfen in keiner include() Anweisung stehen
- REQUEST-Variablen dürfen nicht ungefiltert ausgegeben werden
    - Wenn eine Ausgabe von REQUEST-Variablen nötig ist, muss htmlspecialchars() verwendet werden
- Zum Transport von Variablen müssen Sessions verwendet werden und keine REQUEST Variablen
- Wenn Variablen nur einmal definiert werden sollen (etwa Konfigurationsdaten, besonders Pfade für include()), sind Konstanten zu verwenden

- Daten sind spezifisch zu nutzen - die Unterschiede zwischen POST und GET sollten genutzt werden
- REQUEST-Daten, die in einen Datenbankquery übernommen werden, müssen validiert werden
- REQUEST-Daten dürfen nicht in Dateioperationen (fopen() etc.) verwendet werden
- Variablen in globalem Kontext sind immer zu initialisieren und zu prüfen

Golde Regel

Traue keinen Benutzereingaben

Maßnahmen aus Penetration Test

Dateiupload

- Es sind nur als sicher geltende Dateitypen (PDF, TXT) für den Dateiupload erlaubt
- Neben der Dateiendung muss auch der MIME-Typ der Datei überprüft werden
- Die Größe je Dateiupload muss beschränkt werden

Übertragung von Cookies

Wenn es technisch Möglich ist, muss das HttpOnly-Flag bei Cookies aktiviert werden. Damit wird verhindert, das clientseitige Skripte den Cookie lesen können.

Tabelle 19

```
bool setcookie ( string $name [, string $value [, int $expire = 0 [,
string $path [, string $domain [, bool $secure = false [, bool $httponly =
false ]]]]] )
```

Session Expiration / Timeout

Session-IDs müssen auch bei unsachgemäßem Beenden der Webanwendung deaktiviert warden. Session-IDs müssen durch die Webanwendung vorgegeben werden. Clientseitig erstellte Session-IDs dürfen durch die Webanwendung nicht akzeptiert werden.

Tabelle 20

Beispielskript:

```
#session starten
session_start();

#maximale Leerlaufzeit in Sekunden definieren
define("MAX_IDLE", "60");

#session variable registrieren
```


Knorrstraße 10 I A-4600 Wels
T +43 7242 9396-7100
F +43 7242 9396-7182
info@itandtel.at I www.itandtel.at
Eine Marke der E-Werk Wels AG.
Elektrizitätswerk Wels AG
Stelzhamerstraße 27 I A-4600 Wels
DVR Nr. 0006815
UID Nr. ATU 23478001
FN Nr. 102455W I LG Wels

```
$_SESSION['test'] = "test";

#prüfen, ob nicht Session Variable registriert ist
if (!isset($_SESSION['timeout_idle']))
{
  echo "create";

  # Session Variable registrieren
  #aktuellen Unix timestamp + maximale Leerlaufzeit
  $_SESSION['timeout_idle'] = time() + MAX_IDLE;
}
# Wenn Session Variable Registriert ist
else
{
  # Prüfen ob Leerlaufzeit kleiner als aktuelle Zeit
  # Timestamp vergleichen
   if ($_SESSION['timeout_idle'] < time())
  {
  echo "destroy";

  # Alle Session Variablen löschen
  session_unset();

  # Session Variablen leeren
  $_SESSION = array();

  # aktuelle Session beeenden
  session_destroy();
  }
  else
  {
    echo "restore";

    # Wenn Leerlaufzeit nicht überschritten wurde,
    # Session Variable mit aktuellem
    # Timestamp + Leerlaufzeit belegen
    $_SESSION['timeout_idle'] = time() + MAX_IDLE;
  }
}

# Session Inhalt zu Debug Zwecken ausgeben
# Session sollte leer sein!
print_r($_SESSION);
```

Directory Listing

Directory Listing muss grundsätzlich für alle Verzeichnisse des Webservers per Default deaktiviert werden.

Penetration Test

Informationen, Protokolle und Maßnahmen zu den Penetration- Tests sind unter T:\MAIN\Allgemein\Organisation\Risiko-Management\2010 Pen Test KPMG abgelegt.

Versionskontrolle / Change- und Feature Requests

Subversion

Zur Versionskontrolle sind die ITandTEL SVN Repositories zu verwenden, die über svn://svn.itandtel.at bzw. http://svn.itandtel.at zu erreichen sind.

SVN bietet die Möglichkeit, Dateien eines Projektes an einer zentralen Stelle abzulegen, eine Historie der Veränderungen abzubilden und einzelne Versionen eines Projektes zu vergleichen.

Zur Verwendung von Subversion ist das separate Abweisungsdokument „SVN“ zu beachten.

http://wiki.itandtel.at/index.php/C:_Subversion_SVN

ITSM

Change- und Feature Requests sind in der ITSM Software abzubilden. ITSM bietet die Möglichkeit, Anforderungen mit einer Priorität zu versehen und einem Bearbeiter zuzuweisen.

Zur Verwendung von ITSM ist das separate Anweisungsdokument „ITSM“ zu beachten (http://wiki.itandtel.at/index.php/ITSM).

Sicherstellung / Kontrolle

Um sicherzustellen, dass die Regeln und Empfehlungen eingehalten und von jedem Team Mitglied eingehalten werden, gilt das Vier- Augen- Prinzip. Jeder Projekt Mitarbeiter muss zu jedem Zeitpunkt nachweisen, dass die Regeln eingehalten wurden nach diesem Anweisungsdokumentiert entwickelt wurde. Jeder Mitarbeiter hat zum Projekt Ordner bzw. SVN Repositorie Zugriff und kann Stichproben machen.

Bei Fertigstellung des Projektes gibt es ein Abnahmeprotokoll, dessen Bestandteil der Punkt „Einhaltung Coding Guidelines“ ist. Diese abschließende Kontrolle und Freigabe für das Abnahmeprotokoll wird vom Service & Betrieb Prozess- Verantwortlichen durchgeführt.