

Konfiguration

Table of contents

1 Allgemein.....	2
2 Datenbank.....	2
3 Seiteninformationen.....	3
4 Kalender / Zeiteinheiten.....	3
5 Authentifizierung.....	4
6 Sprache.....	5
7 Kategorien.....	5
8 Sontige Einstellungen.....	6

1. Allgemein

Jegliche Konfiguration des Ressourcenbuchungssystems findet in der Datei `config.inc.php` statt. Bei einer neuen Installation existiert diese Datei jedoch noch nicht, nur `config.inc.php-dist`, eine Beispielkonfiguration. Sie können diese Datei einfach nach `config.inc.php` kopieren und diese dann bearbeiten.

Achtung:

Nicht dokumentierte Variablen haben zur Zeit keine Wirkung, da z.B. beim [Fork](http://de.wikipedia.org/wiki/Abspaltung_(Softwareentwicklung)) ([http://de.wikipedia.org/wiki/Abspaltung_\(Softwareentwicklung\)](http://de.wikipedia.org/wiki/Abspaltung_(Softwareentwicklung))) von MRBS einige Module, wie z.B. die Authentifizierung über LDAP, deaktiviert wurden, oder die Variablen sind nur während der Entwicklung bedeutend und sollten deshalb nicht verändert werden. **Die deaktivierten Module sind zwar zur Zeit nicht verfügbar, werden aber in nächster Zeit wieder bereitgestellt, da sie manuell auf Sicherheit überprüft werden, sollten Sie eines dieser Module benötigen, so wenden Sie sich bitte an die [Mailingliste](http://groups.google.com/group/schoorbs-general-discussion) (<http://groups.google.com/group/schoorbs-general-discussion>), damit diese Modul, wenn möglich, bevorzugt überprüft wird.**

2. Datenbank

Der erste Abschnitt der Konfiguration von Schoorbs in der Datei `config.inc.php` betrifft die Einstellungen zur Datenbank. Wie schon im Kapitel zur Installation erwähnt, hat man bei Schoorbs nur die Wahl zwischen MySQL und PostgreSQL. Jedoch gibt es für die Nutzung von MySQL zwei unterschiedliche Module, ein Mal das Module `mysqli`, welches an sich nur für Systeme mit PHP 5+ und MySQL 5+ eignet, bei diesen aber sehr empfehlenswert ist. Für alle anderen Systeme mit MySQL sollte man das Modul `mysql` nutzen. Für den Zugriff auf PostgreSQL steht nur das Modul `pgsql` zur Verfügung. Das jeweils passende Modul schreiben Sie als Wert zur Variable `$dbsys`. In die Variable `$db_host` tragen sie den Namen des Computers ein, auf welchem der Datenbankserver läuft, bei PostgreSQL können sie diesen Wert auch leer lassen, wenn sie möchten, dass die Verbindung zur Datenbank anstatt über TCP/IP per Unix Socket geschieht. Den Namen der Datenbank tragen Sie bei der Variable `$db_database` ein. Nutzernamen und Passwort gehören in die Variablen `$db_login` und `$db_password`. In `$db_tbl_prefix` tragen Sie den während der Installation gewählten Tabellenpräfix ein, meistens ist dies `'schoorbs_'` oder `'mrbs_'`. Als letzte Konfigurationsmöglichkeit können Sie wählen, ob Sie persistente Datenbankverbindungen haben wollen. Sie sollten diese jedoch nur nutzen, wenn Sie genau wissen, was Sie tun, ansonsten empfiehlt sich das Deaktivieren von persistenten Verbindungen, welches durch das Setzen von `$db_nopersist = 1;` bewirkt wird.

Beispielkonfiguration (für eine Verbindung per MySQLi):

```
$dbsys = "mysqli";
$db_host = "localhost";
$db_database = "schoorbs";
```

```
$db_login = 'schoorbs';  
$db_password = 'geheimes_passwort';  
$db_tbl_prefix = "schoorbs_";  
$db_nopersist = 1;
```

3. Seiteninformationen

Als sogenannte Seiteninformationen kann man hier nur den Administrations-Kontakt und den Titel der Seite festlegen. Den Namen des Administrators weist man der Variable `$mrbs_admin` zu, seine E-Mail trägt man in `$mrbs_admin_email` ein. Den Titel der Seite legt man in der Variable `$mrbs_company` fest, dieser Titel kann auch (X)HTML-Tags wie z.B. `<a>` enthalten.

Beispielkonfiguration:

```
$mrbs_admin = "Herr Mustermann";  
$mrbs_admin_email = "mail@example.org";  
$mrbs_company = "Die Example Firma";
```

4. Kalender / Zeiteinheiten

Bei Schoorbs gibt es 2 Möglichkeiten die Zeiteinheiten, welche gebucht werden können festzulegen. Auf der einen Seite gibt es die Möglichkeit eine feste Dauer (`$resolution`) in Sekunden anzugeben, die für jede Einheit gilt. Solche Einheiten kann man zwischen einem angegebenen Tagesstart (`$morningstarts` (Stunde) und `$morningstarts_minutes` (Minuten)) und einem angegebenen Tagesende (`$eveningends` (Stunde) und `$eveningends_minutes` (Minuten)) buchen. Hat man z.B. als Tagesstart 7:30 angeben, und als Auflösung 1800 (-> 1/2 h) gewählt, so kann man jeweils um 7:30, 8:00, 8:30, ... eine Einheit buchen. Möchte man nun aber Zeiteinheiten mit unterschiedlicher Länge bzw. möchte man die Einheit überhaupt nicht über die Zeit, sondern über einen spezifischen Namen bestimmen, so kann man einen Array (`$periods`) angeben, welche Einheiten verfügbar sein sollen. Um diese Funktion zu nutzen, muss man die Variable `$enable_periods` auf `true` setzen.

Allgemein gesehen kann man noch auswählen, ob der Wochenbeginn an einem Sonntag (`$weekstarts = 0;`) oder an einem Montag (`$weekstarts = 1;`) ist.

Beispielkonfiguration mit festen Zeiteinheiten:

```
$enable_periods = false;  
$resolution = 2700;  
$morningstarts = 8;  
$eveningends = 17;  
$morningstarts_minutes = 0;  
$eveningends_minutes = 0;  
$periods[] = array();  
$weekstarts = 1;
```

Beispielkonfiguration mit benannten Zeiteinheiten

```
$enable_periods = true;  
$resolution = 2700;  
$morningstarts = 8;
```

```

$eveningends      = 17;
$morningstarts_minutes = 0;
$eveningends_minutes = 0;
$periods[] = "08:00-08:45";
$periods[] = "08:45-09:30";
$periods[] = "09:30-09:45";
$periods[] = "09:45-10:30";
$periods[] = "10:30-11:15";
$periods[] = "11:15-11:30";
$periods[] = "11:30-12:15";
$periods[] = "12:15-13:00";
$periods[] = "13:00-13:15";
$periods[] = "13:15-14:00";
$periods[] = "14:00-14:45";
$periods[] = "14:45-15:30";
$periods[] = "15:30-16:15";
$weekstarts      = 1;

```

5. Authentifizierung

Bei der Authentifizierung bei Schoorbs gibt es 2 modulare Bereiche, 'authentication' (Variable: `$auth["type"]`) und 'session' (Variable: `$auth["session"]`).

Der Bereich 'session' ist für die Schnittstelle verantwortlich, die vom Benutzer Nutzernamen und Passwort verlangt und dieses auch über mehrere HTTP-Requests hinweg speichert. Hier stehen die Module 'cookie' (Abfrage des Passworts per HTML-GUI, speichern des Namens/Passworts in einem Cookie), 'host' (Der DNS-Hostname des Nutzers ist der Nutzernamen, Passwörter gibt es nicht), 'http' (Name/Passwort werden über HTTP-Authentifikation nachgefragt), 'ip' (Die IP des Nutzers ist der Nutzernamen, Passwörter gibt es nicht), 'php' (Name/Passwort werden per HTML abgefragt, Name/Passwort werden mit PHPs internem Session-Handling abgespeichert) und 'remote_user' (wie 'http', nur hier ist der Webserver für die HTTP-Authentifikation zuständig).

Zusatz zu 'http': Den Titel der HTTP-Authentifikation kann man mit der Variable `$auth['realm']` setzen.

Zusatz zu 'remote_user': Man kann einen Link spezifizieren, auf dem sich Nutzer ausloggen können: `$auth['remote_user']['logout_link']`

Die Überprüfung der Kombination von Nutzernamen und Passwort wird im Bereich 'authentication' erledigt. Hier existieren 4 Module: 'config' (Die zulässigen Name/Passwort-Kombinationen werden in der `config.inc.php` fest abgespeichert), 'db' (Name und Passwort werden in der Datenbank abgespeichert), 'ext' (Es wird ein Programm ausgeführt, welches die Kombination überprüft) und 'none' (Jegliche Kombination von Name/Passwort wird als korrekt angesehen).

Zusatz zu 'config': Die Name/Passwort-Kombinationen werden im Array `$auth["user"]` gespeichert, z.b.: `$auth["user"]["nutzernamen1"] = "password1";`

Zusatz zu 'ext': Das auszuführende Programm wird in der Variable `$auth["prog"]` festgelegt, die Parameter für das Programm in `$auth["params"]`, wobei bei den Parametern '#PASSWORD#' und '#USERNAME#' jeweils durch die zu überprüfende

Kombination ersetzt wird.

Die beiden Bereiche regeln zwar, wie und wer sich einloggen darf, jedoch legen sie noch nicht fest, welcher Benutzer Administratorrechte hat. Einem Benutzer kann man Administratorrechte geben in dem man ihm dem Array `$auth["admin"]` hinzufügt, z.B.: `$auth["admin"][] = "nutzernamen1";`. Administratoren können neue Bereiche und Ressourcen erstellen und Einträge anderer Nutzer bearbeiten.

Beispielkonfiguration (mit 'session'='php' und 'type'='config'):

```
$auth["session"] = "php";
$auth["type"] = "config";
# The list of administrators (can modify other peoples settings)
$auth["admin"][] = "nutzernamen1";
# 'auth_config' user database
$auth["user"]["nutzernamen1"] = "password1";
$auth["user"]["nutzernamen2"] = "password2";
```

6. Sprache

Zur Sprache sind bei Schoorbs nicht viele Einstellungen notwendig. Normalerweise wird alles per UTF-8 ausgegeben, an einigen Stellen besteht jedoch die Möglichkeit dies zu unterlassen, möchte man dies, so kommentiert man die Zeile `$unicode_encoding = 1;` aus. Schoorbs versucht die Sprache des Nutzers automatisch zu ermitteln, sollte dies jedoch mal nicht gelingen, so wird auf die Variable `$default_language_tokens` zurückgegriffen, welche einen zweistelligen Sprachcode (kleine Buchstaben, z.B.: 'de') enthält. Standardmäßig werden die häufig gestellten Fragen (FAQ) in Englisch angezeigt, es existieren jedoch auch Versionen in Spanisch ('_es'), Deutsch ('_de') und Französisch ('_fr'), möchte man lieber eine dieser Sprachen haben, so muss man die jeweilige Zeichenkette in die Variable `$faqfilelang` eintragen.

7. Kategorien

Schoorbs bietet die Möglichkeit Buchungen in Kategorien einzuordnen und sie passend farblich zu hinterlegen. Dazu existieren die Arrayspalten 'A' - 'J' des Arrays `$typel`. Jeder Buchstabe färbt den jeweiligen Eintrag anders, die Farbe wird per CSS (`schoorbs-misc/style/style.css`) festgelegt. Die Kategorien 'external' und 'internal' müssen hierbei jedoch immer erhalten bleiben.

Beispielkategorien (bei schulischer Nutzung):

```
$typel["A"] = "Referat";
$typel["B"] = "normaler Unterricht";
$typel["C"] = "Kursarbeit";
# $typel["D"] = "D";
$typel["E"] = get_vocab("external");
# $typel["F"] = "F";
# $typel["G"] = "G";
# $typel["H"] = "H";
```

```
$typel["I"] = get_vocab("internal");  
# $typel["J"] = "J";
```

8. Sontige Einstellungen

Sonstige Konfigurationen sind z.B. die maximale Anzahl an Wiederholungen, die ein regelmäßiger Termin haben kann (`$max_rep_entries`). Weiterhin kann man noch festlegen, ob auch auf der rechten Seite der Tabelle die Uhrzeiten angezeigt werden sollen, was man durch das Setzen von `$times_right_side` auf `true` bzw. `false` aktivieren bzw. deaktivieren kann. Mit `$javascript_cursor` kann man das farbliche Hervorheben der Tabellenspalte aktivieren, über die man gerade mit dem Mauszeiger geht, diese Funktion wäre zwar auch mit CSS implementierbar, jedoch würde sie dann nicht korrekt von allen Browser dargestellt werden, weshalb hier die Javascript-Variante, die einzig verfügbare ist. Mit `$default_view` legt man fest, welche Ansicht normalerweise benutzt wird, wenn der Nutzer im Browser die Index-Seite ansteuert, mögliche Ansichten sind: Tag('day'), Woche('week') oder Monat('month').

Konfiguration

(<http://creativecommons.org/licenses/by-sa/3.0/>)